Contents

Greetings and foreword ........................................................................................................ vi
Technical Program Chair’s Message ..................................................................................... vii
Organizing Committee .......................................................................................................... viii
Technical Program Committee .............................................................................................. ix
Advisory Committee .............................................................................................................. x
NOLTA Steering Committee ................................................................................................ x
Special Session Organizers ................................................................................................ xi
Symposium Information .................................................................................................... xii
Symposium Venue .............................................................................................................. xiii
Social Events ...................................................................................................................... xiii
Session at a Glance .............................................................................................................. xiv

Technical Program xvii

P1L-A [Plenary Talk] ........................................................................................................ xvi
A1L-A [Special Session] Evolving Complex Networks and their Applications - 1 ............. xvi
A1L-B Cellular Neural Networks .......................................................................................... xviii
A1L-C Nonlinear Circuits and Systems ................................................................................ xviii
A1L-D System Analysis and Modeling ............................................................................... xix
A2L-B [Special Session] Hybrid Manycore Architectures - Capabilities and Limitations - 1 xx
A2L-C [Special Session] Pattern Formation, Emergence, and Imaging Featuring Nonlinear Dynamics - 1 . xxi
A2L-D Neural Networks 1 ................................................................................................... xxi
A3L-A [Special Session] Verified Numerical Computations for Linear and Nonlinear Problems ........... xxii
A3L-B [Special Session] Hybrid Manycore Architectures - Capabilities and Limitations - 2 .......... xxii
A3L-C [Special Session] Pattern Formation, Emergence, and Imaging Featuring Nonlinear Dynamics - 2 . xxiii
A3L-D Neural Networks 2 ................................................................................................ xxiii
B1L-B [Special Session] Nonlinear Analysis and Processing of Facial Images ............. xxiv
B1L-C Complex Systems ..................................................................................................... xxv
B1L-D Bifurcation and Chaos ............................................................................................ xxv
B1L-E Circuit Implementation ............................................................................................. xxvi
P2L-A [Plenary Talk] ........................................................................................................ xxvi
B2L-B [Special Session] Complex Networks and their Dynamics - 1 ......................... xxvii
B2L-C Communication Systems .......................................................................................... xxviii
B2L-D Chaotic Circuits ...................................................................................................... xxviii
B2L-E Circuit Analysis ....................................................................................................... xxix
B3L-B [Special Session] Complex Networks and their Dynamics - 2 ......................... xxx
B3L-C Time Sequences ...................................................................................................... xxx
B3L-D Control and Robotics ............................................................................................. xxxi
C1L-A [Special Session] Nonlinear Time Series Analysis - 1 ............................................ xxxii
C1L-B Image and Signal Processing ................................................................................... xxxii
C1L-C [Special Session] Nonlinear Maps and Applications - 1 ....................................... xxxiii
C1L-D Bio-Inspired Algorithms ......................................................................................... xxxiii
P3L-A [Plenary Talk] ........................................................................................................ xxxiv
2010 International Symposium on Nonlinear Theory and its Applications

September 5–8, 2010
MCK (International Cultural Center), Krakow, Poland

Organizer:
Research Society of Nonlinear Theory and its Applications, IEICE

In Cooperation with:
Technical Group on Nonlinear Problems, IEICE
Technical Group on Circuits and Systems, IEICE
IEEE Circuits and Systems Society
IEEE Poland Section
Jagiellonian University
Greetings and Foreword

On behalf of the Organizing Committee, we sincerely welcome you to Krakow and to the 2010 International Symposium on Nonlinear Theory and its Applications (NOLTA2010). This is the 20th NOLTA Symposium organized by the Research Society of Nonlinear Theory and its Applications, IEICE, and more recently, sponsored by the Circuits and Systems Society, IEEE. Since its foundation, NOLTA has been held in Japan, USA, Switzerland, Germany, China, Belgium, Italy, Canada, and Hungary. We consider a great honor and privilege to organize the NOLTA2010 in Krakow, Poland, and offer you a taste of Polish hospitality.

Krakow was the capital of Poland from 1038 to 1596, sit of Kings of Poland. It is a sister City with Kyoto, Japan! The city rose to prominence in 1364, when Casimir the Great founded the University of Krakow, the second oldest university in central Europe after the Charles University in Prague. Eversince, being administrative center of Poland, Krakow has been also the center of sciences and arts. Among alumni of the Jagiellonian University were, astronomer Nicolaus Copernicus and famous writers of Polish renaissance Jan Kochanowski, Andrzej Frycz-Modrzewski and Mikolaj Rej, later Olszewski and Wroblewski who liquefied oxygen and nitrogen, Cybulski who explained the functioning of adrenaline, Marchlewski, the author of pioneering works on chlorophyll and Browicz who identified a typhoid microbe. In 1978, UNESCO placed Kraków on the list of World Heritage Sites. In the same year, Karol Wojtyła was elevated to the papacy as John Paul II, the first non-Italian pope in 455 years. For many people, thanks to its rich history, Krakow represents a synthesis of all Polish culture, connecting tradition with modernity. In the special atmosphere of the beautiful and mysterious streets of the Old Town and Kazimierz, after the sessions, you will find everything you need to enjoy your free time. Galleries full of exhibitions, concerts, performances, cafes, pubs and restaurants.

The main feature of the NOLTA 2010 Symposium is its technical program offering invited lectures, regular papers and special sessions. We would like to thank the Technical Program Co-Chairs, Prof. Zbigniew Galias (AGH-UST University) and Prof. Yoshifumi Nishio (Tokushima University) for putting together such an excellent technical program, the Special Sessions Co-Chairs, Prof. Sergio Callegari (University of Bologna) and Toshimichi Saito (Hosei University) for selecting and organizing such valuable special sessions. Special thanks go out to the Technical Program Committee Members and reviewers for their dedicated service. We would also like to thank the keynote speakers and session co-chairs for their fine contributions to the technical program. Last but not least we would like to thank all the authors for their excellent contributions, without their efforts NOLTA2010 might never have been a success. The NOLTA2010 Symposium would not be possible without the endless help and dedication of the members of the Symposium Organization Committee, including the Publication Chair, Finance Co-Chairs, Publicity Chair, Technical Program Secretary, Local Arrangement and, above all, the General Co-Secretary, Hiroo Sekiya (Chiba University).

Maciej Ogorzalek
Jagiellonian University
General Co-Chairs, NOLTA2010

Mamoru Tanaka
Sophia University

vi
On behalf of the technical program committee, we are delighted to welcome all of you to the International Symposium on Nonlinear Theory and Its Applications (NOLTA 2010). NOLTA 2010’s technical program covers topics in a variety of areas of nonlinear circuits and theory, bifurcation and chaos, design and analysis of nonlinear oscillators, neural networks, circuit and systems, complex systems, control and robotics, time series analysis, and so on. We are confident that the attendees will enjoy the technical program. In total, 162 papers including 78 special session papers are scheduled for presentations in 38 sessions. These papers were submitted from 21 countries. NOLTA 2010 also features three exciting plenary talks. Topics of the plenary talks are: “Nonlinear Dynamics of Biological Rhythms” by Prof. Albert Goldbeter (Universite Libre de Bruxelles, Belgium), “Advanced PI/SI/EMI Simulation Technology for High-Speed Electronic Design” by Prof. Hideki Asai (Shizuoka University, Japan) and “Asymptotic Behaviour of Blinking (Stochastically Switched) Dynamical Systems” by Prof. Martin Hasler (EPFL, Switzerland).

We are pleased to have these speakers presenting their latest research topics. We would like to offer special thanks to the reviewers, the technical program committee, consisting 25 committee members, the special session co-chairs, the organizers of special sessions, for their dedicated services. We also would like to thank the session chairs for their fine contributions to the technical program. Finally, our sincere thanks also go to the all authors for their excellent contributions and participation in NOLTA 2010. Thank you very much for all of the participants. We really hope that all of you enjoy the fruitful technical discussions in Krakow!

Zbigniew Galias  
AGH University  
Technical Program Co-Chairs, NOLTA 2010
Organizing Committee

GENERAL CO-CHAIRS
Maciej Ogorzalek (Jagiellonian University)
Mamoru Tanaka (Sophia University)

TECHNICAL PROGRAM CO-CHAIRS
Zbigniew Galias (AGH University)
Yoshifumi Nishio (Tokushima University)

TECHNICAL PROGRAM SECRETARY
Yoko Uwate (Tokushima University)

SPECIAL SESSION CO-CHAIRS
Sergio Callegari (University of Bologna)
Toshimichi Saito (Hosei University)

FINANCE CHAIR
Masahiro Wada (Konan University)
Kenya Jin’no (Nippon Institute of Technology)

PUBLICATION CHAIR
Tetsushi Ueta (Tokushima University)

PUBLICITY CHAIR
Yuichi Tanji (Kagawa University)

GENERAL CO-SECRETARIES
Leszek Bogacz (Jagiellonian University)
Hiroo Sekiya (Chiba University)
Technical Program Committee

**Technical Program Co-Chairs**
Zbigniew Galias (AGH Univ.)
Yoshifumi Nishio (Tokushima Univ.)

**Technical Program Secretary**
Yoko Uwate (Tokushima University)

**Members**
Tetsuya Asai (Hokkaido Univ.)
Shinji Doi (Kyoto Univ.)
Orla Feely (Univ. College Dublin)
Daniele Fournier-Prunaret (INSA, Toulouse)
Martin Hasler (EPFL)
Yoshihiko Horio (Tokyo Denki Univ.)
Tohru Ikeguchi (Saitama Univ.)
Mieczyslaw Jessa (Poznan Univ. of Tech.)
Geza Kolumban (Pazmany Peter Catholic Univ.)
Keiji Konishi (Osaka Prefecture Univ.)
Francis Lau (Hong Kong Polytechnic Univ.)
Michael Small (Hong Kong Polytechnic Univ.)
Marco Storace (Univ. of Genoa)
Yoshihiko Susuki (Kyoto Univ.)
Peter Szolgay (Hungarian Academy of Sciences)
Norikazu Takahashi (Kyushu Univ.)
Isao Tokuda (JAIST)
Torikai Hiroyuki (Osaka Univ.)
Ljiljana Trajkovic (Simon Fraser Univ.)
Chi K. Tse (Hong Kong Polytechnic Univ.)
Mustak E. Yalcin (Istanbul Tech. Univ.)
Kiyotaka Yamamura (Chuo Univ.)
Advisory Committee

K. Aihara (Univ. of Tokyo) S. Amari (FRP, RIKEN)
G. Chen (City Univ. Hong Kong) L. O. Chua (U. C. Berkeley)
R. Eberhart (IUPUI) T. Endo (Meiji Univ.)
A. Fettweis (Ruhr Univ. Bochum) L. Fortuna (Univ. of Catania)
W. J. Freeman (U. C. Berkeley) M. Hasler (EPFL)
H. Hayashi (Kyushu Inst. Tech.) T. Hikihara (Kyoto Univ.)
K. Horiuchi (Waseda Univ.) M. Iri
K. Judd (Univ. of Western Australia) H. Kawakami (Tokushima Univ.)
M. P. Kennedy (Univ. College Cork) T. Kohda (Kyushu Univ.)
E. S. Kuh (U. C. Berkeley) R. W. Liu (Univ. of Notre Dame)
T. Matsumoto (Fukui Univ. of Tech.) I. Mezic (UCSB)
S. Mori (Keio Univ.) T. Nagashima (Muroran Inst. of Tech.)
K. Nakajima (Tohoku Univ.) T. Nishi (Waseda Univ.)
J. A. Nossek (Tech. Univ. of Munich) S. Oishi (Waseda Univ.)
M. J. Ogorzalek (Jagiellonian Univ.) K. Okumura (Simon Fraser Univ.)
M. Plum (Karlsruhe Univ.) T. Roska (Hungarian Academy of Science)
S. M. Rump (Technical Univ. of Hamburg) T. Saito (Hosei Univ.)
I. W. Sandberg (Univ. of Texas at Austin) Y. Sawada (Tohoku Inst. of Tech.)
W. Schwarz (Tech. Univ. of Dresden) G. Setti (Univ. of Ferrara)
R. Stoop (ETH / Univ. of Zurich) M. Tanaka (Sophia Univ.)
L. Trajkovic (Simon Fraser Univ.) C. K. Tse (Hong Kong Polytech. Univ.)
Y. Ueda (Waseda Univ.) A. Ushida (Tokushima Bunri Univ.)
T. Ushio (Osaka Univ.) J. Vandewalle (KU Leuven)
P. Werbos (National Science Foundation) A. N. Willson, Jr. (UCLA)
A. H. Zemanian (State Univ. NY at Stony Brook)

IEICE NOLTA Subsociety Steering Committee (2010)

Chair
Takashi Hikihara (Kyoto Univ.)

Secretary
Takashi Hisakado (Kyoto Univ.)

Treasurer
Hiroyuki Torikai (Osaka Univ.)

Members
Masaharu Adachi (Tokyo Denki Univ.) Shinji Doi (Kyoto Univ.)
Tetsuro Endo (Meiji Univ.) Yoshihiko Horio (Tokyo Denki Univ.)
Tohru Ikeguchi (Saitama Univ.) Kenya Jin’no (Nippon Inst. Tech.)
Takuji Kousaka (Oita Univ.) Kunihiko Mitsubori (Takushoku Univ.)
Tomomasa Nagashima (Muroran Inst. Tech.) Yoshifumi Nishio (Tokushima Univ.)
Yuzo Ohta (Kobe Univ.) Norikazu Takahashi (Kyushu Univ.)
Mamoru Tanaka (Sophia Univ.) Isao Tokuda (JAIST)
Tetsushi Ueta (Tokushima Univ.)
Special Session Organizers

A1L-A Evolving Complex Networks and their Applications - 1
A2L-A Evolving Complex Networks and their Applications - 2

Organizers Norbert Marwan (Potsdam Institute for Climate Impact Research) and Jurgen Kurths (University of Potsdam)

A3L-A Verified Numerical Computations for Linear and Nonlinear Problems

Organizers Shin’ichi Oishi (Waseda University) Siegfried M. Rump (Hamburg University of Technology) Takeshi Ogita (Tokyo Woman’s Christian University) and Katsuhisa Ozaki (Waseda University)

A2L-B Hybrid Manycore Architectures - Capabilities and Limitations - 1
A3L-B Hybrid Manycore Architectures - Capabilities and Limitations - 2

Organizer Peter Szolgay (Hungarian Academy of Sciences)

A2L-C Pattern Formation, Emergence, and Imaging Featuring Nonlinear Dynamics - 1
A3L-C Pattern Formation, Emergence, and Imaging Featuring Nonlinear Dynamics - 2

Organizers Ken’ichi Fujimoto (Tokushima University) and Tetsushi Ueta (Tokushima University)

B1L-A Aspects of Optimization with Nonlinear Dynamics - 1
B2L-A Aspects of Optimization with Nonlinear Dynamics - 2
B3L-A Aspects of Optimization with Nonlinear Dynamics - 3

Organizers Kenya Jin’no (Nippon Institute of Technology) and Mikio Hasegawa (Tokyo University of Science)

B1L-B Nonlinear Analysis and Processing of Facial Images

Organizer Yasue Mitsukura (Tokyo University of Agriculture and Technology)

B2L-B Complex Networks and their Dynamics - 1
B3L-B Complex Networks and their Dynamics - 2

Organizers Kohshi Okumura (Simon Fraser University) and Ljiljana Trajkovic (Simon Fraser University)

C1L-A Nonlinear Time Series Analysis - 1
C2L-A Nonlinear Time Series Analysis - 2
C3L-A Nonlinear Time Series Analysis - 3

Organizers Max Little (University of Oxford) and Michael Small (Hong Kong Polytechnic University)
C2L-B  A Nonlinear Dynamics Perspective of Cellular Automata - 1

C3L-B  A Nonlinear Dynamics Perspective of Cellular Automata - 2

Organizers  Giovanni Pazienza (MTA-SZTAKI & Pazmany University) and Tamas Roska (MTA-SZTAKI & Pazmany University)

C1L-C  Nonlinear Maps and Applications - 1

C2L-C  Nonlinear Maps and Applications - 2

Organizers  Daniele Fournier-Prunaret (INSA Toulouse) Tetsushi Ueta (Tokushima University) Laura Gardini (University of Urbino) and Yoshifumi Nishio (Tokushima University)
Symposium Information

Symposium Venue
MCK (International Cultural Center)
Rynek Główny (Main Market Square) 25, 31-008 Kraków, Poland

Session Room Information
• Registration Desk
  On the ground floor (Ravens).

• Room A
  On the ground floor (Ravens).

• Room B
  On the fourth floor.

• Room C
  In the cellar for 45 people.

• Room D
  On the fifth floor.

• Room E
  In the cellar for 35 people.

Social Events
• September 5, 18:00–20:00, Welcome reception: On the first floor (Ravens), MCK.

• September 6, Countryside Dinner at the Zalesie Manor
  Bus departure: at 18:30 from the Radisson Blu Hotel.
  Location: ul. Straszewskiego 17, Stare Miasto, 31-100.
  Zalesie Manor is located 25 km from Krakow.

• September 7, 19:30–, Gala Dinner at the “Wierzynek” restaurant
  Location: Rynek Główny (Main Market Square) 15, 31-008
## Session at a Glance

### September 6, 2010 (Monday)

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Session</th>
</tr>
</thead>
<tbody>
<tr>
<td>September 6</td>
<td>9:00–9:20</td>
<td>Opening Ceremony, Room A</td>
</tr>
<tr>
<td>September 6</td>
<td>9:20–10:20</td>
<td>Plenary Talk 1, Prof. Albert Goldbeter, Chair: Maciej Ogorzalek, Room A</td>
</tr>
<tr>
<td>September 6</td>
<td>10:20–10:50</td>
<td>Coffee break</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Room A</th>
<th>Room B</th>
<th>Room C</th>
<th>Room D</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1L-A [Special Session] Evolving Complex Networks and their Applications - 1 Chair: Norbert Marwan Page xvii</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A1L-B Cellular Neural Networks Chair: Hideki Asai Page xviii</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A1L-C Nonlinear Circuits and Systems Chair: Jinhu Lu Page xviii</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A1L-D System Analysis and Modeling Chair: Takashi Hikihara Page xix</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Session</th>
</tr>
</thead>
<tbody>
<tr>
<td>September 6</td>
<td>10:50–12:30</td>
<td>Coffee break</td>
</tr>
<tr>
<td>September 6</td>
<td>12:30–14:00</td>
<td>Lunch break</td>
</tr>
<tr>
<td>September 6</td>
<td>14:00–15:20</td>
<td>A2L-A [Special Session] Evolving Complex Networks and their Applications - 2 Chair: Norbert Marwan Page xix</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A2L-B [Special Session] Hybrid Manycore Architectures - Capabilities and Limitations - 1 Chair: Peter Szolgay Page xx</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A2L-C [Special Session] Pattern Formation, Emergence, and Imaging Featuring Nonlinear Dynamics - 1 Chair: Ken’ichi Fujimoto Page xxi</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A2L-D Neural Networks 1 Chair: Yoshihiko Horio Page xxi</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Session</th>
</tr>
</thead>
<tbody>
<tr>
<td>September 6</td>
<td>15:20–15:40</td>
<td>Coffee break</td>
</tr>
<tr>
<td>September 6</td>
<td>15:40–17:00</td>
<td>A3L-A [Special Session] Verified Numerical Computations for Linear and Nonlinear Problems Chair: Katsuhisa Ozaki Page xxii</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A3L-B [Special Session] Hybrid Manycore Architectures - Capabilities and Limitations - 2 Chair: Peter Szolgay Page xxii</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A3L-C [Special Session] Pattern Formation, Emergence, and Imaging Featuring Nonlinear Dynamics - 2 Chair: Tetsushi Ueta Page xxiii</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A3L-D Neural Networks 2 Chair: Tohru Ikeguchi Page xxiii</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Session</th>
</tr>
</thead>
<tbody>
<tr>
<td>September 6</td>
<td>18:30–</td>
<td>Countryside Dinner at Zalesie</td>
</tr>
</tbody>
</table>

xiv
### September 7, 2010 (Tuesday)

<table>
<thead>
<tr>
<th>Time</th>
<th>Room A</th>
<th>Room B</th>
<th>Room C</th>
<th>Room D</th>
<th>Room E</th>
</tr>
</thead>
<tbody>
<tr>
<td>9:00–10:40</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10:40–11:10</td>
<td>Coffee break</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>September 7,</td>
<td></td>
<td>Plenary Talk 2, Prof. Hideki Asai Chair: Zbigniew Galias, Room A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:10–12:10</td>
<td>Lunch break</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>September 7,</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12:10–13:40</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>September 7,</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13:40–15:20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>September 7,</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15:20–15:50</td>
<td>Coffee break</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>September 7,</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15:50–17:30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>September 7,</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19:30–</td>
<td>Gala Dinner at Wierzynek</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### September 8, 2010 (Wednesday)

<table>
<thead>
<tr>
<th>Time</th>
<th>Room A</th>
<th>Room B</th>
<th>Room C</th>
<th>Room D</th>
</tr>
</thead>
<tbody>
<tr>
<td>10:40–11:10</td>
<td>Coffee break</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>September 8, 11:10–12:10</td>
<td>Plenary Talk 3, Prof. Martin Hasler Chair: Yoshifumi Nishio, Room A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>September 8, 12:10–13:40</td>
<td>Lunch break</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15:00–15:20</td>
<td>Coffee break</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17:00–17:30</td>
<td>Farewell and Award Ceremony, on the first floor (Ravens)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Technical Program

P1L-A [Plenary Talk]
DATE: September 6, 10:20–11:20
ROOM: Room A
Chair: Maciej Ogorzalek (Jagiellonian University)

P1L-A1 Nonlinear Dynamics of Biological Rhythms
Albert Goldbeter (Université Libre de Bruxelles)

A1L-A [Special Session] Evolving Complex Networks and their Applications - 1
DATE: September 6, 10:50–12:30
ROOM: Room A
Chair: Norbert Marwan (Potsdam Institute for Climate Impact Research)

A1L-A1 Dynamics on Complex Networks with Time Varying Topology
Jürgen Kurths (Potsdam Institute for Climate Impact Research), Jonathan F. Donges (Potsdam Institute for Climate Impact Research), Norbert Marwan (Potsdam Institute for Climate Impact Research), Yong Zou (Potsdam Institute for Climate Impact Research)

A1L-A2 Evolving Climate Networks
Norbert Marwan (Potsdam Institute for Climate Impact Research), Jonathan F. Donges (Potsdam Institute for Climate Impact Research), Alexander Radebach (Potsdam Institute for Climate Impact Research), Jakob Runge (Potsdam Institute for Climate Impact Research), Jürgen Kurths (Potsdam Institute for Climate Impact Research)

A1L-A3 Synchronization Regulation of a Complex Network by Link Rewiring or Node Pinning
Irene Sendiña-Nadal (Universidad Rey Juan Carlos), Juan Antonio Almendral (Universidad Rey Juan Carlos), Inmaculada Leyva (Universidad Rey Juan Carlos), Javier Buldú (Universidad Rey Juan Carlos), Dongchuan Yu (University of Electronic Science and Technology), Stefano Boccaletti (CNR-Istituto dei Sistemi Complessi)

A1L-A4 Why Scale-Free Networks Are a Good Thing for Controlling Disease Transmission
Michael Small (Hong Kong Polytechnic University)
A1L-B Cellular Neural Networks

DATE: September 6, 10:50–12:30
ROOM: Room B
Chair: Hideki Asai (Shizuoka University)

**A1L-B1** Propagation Mechanism of Phase-Inversion Wave in in-and-Anti-Phase Synchronization on 2D Lattice Oscillator
Seiko Kunihiro (Hiroshima Institute of Technology), Hitoshi Aburatani (Hiroshima Institute of Technology), Masayuki Yamauchi (Hiroshima Institute of Technology), Yoshifumi Nishio (Tokushima University)

**A1L-B2** Cellular Neural Networks with Hopfield Neural Networks Considering the Confidence Degree
Yasuhiro Ueda (Tokushima University), Masakazu Kawahara (Tokushima University), Yoko Uwate (Tokushima University), Yoshifumi Nishio (Tokushima University)

**A1L-B3** Characteristics of Cellular Neural Networks with Dynamic Template for Motion Pictures
Masakazu Kawahara (Tokushima University), Yoko Uwate (Tokushima University), Yoshifumi Nishio (Tokushima University)

**A1L-B4** DT-CNN Annealing with Additive Noise Generated by Class 3 CA: a Comparison with Chaos Annealing
Tomohiro Fujita (Ritsumeikan University), Takeshi Ogura (Ritsumeikan University)

**A1L-B5** Can You Achieve Any Function with a 2-Neuron CNN?
Mireia Vinyoles-Serra (Universitat Ramon Llull), Xavier Vilasís-Cardona (Universitat Ramon Llull)

A1L-C Nonlinear Circuits and Systems

DATE: September 6, 10:50–12:30
ROOM: Room C
Chair: Jinhu Lu (Chinese Academy of Sciences)

**A1L-C1** Generation Method of Extremely Ill-Conditioned Integer Matrices
Tetsuo Nishi (Waseda University), Siegfried Rump (Technische Universität Hamburg), Shin’ichi Oishi (Waseda University)

**A1L-C2** Performance of Adiabatic Quantum Computation Using Neuron-Like Interconnections
Shigeo Sato (Tohoku University), Aiko Ono (Tohoku University), Mitsunaga Kinjo (University of the Ryukyus), Koji Nakajima (Tohoku University)

**A1L-C3** Relaxation Oscillation in Single-Electron Transistor with Resistively-Shunted Gate
Yoshinao Mizugaki (University of Electro-Communications)
A1L-C4  **Rotation Angle Measurement System Using Printed Spiral Inductor and Attractor of Chua’s Circuit**
Takahiro Kurokou (Hiroshima Institute of Technology), Kazuhisa Yoshimatsu (Hiroshima Institute of Technology), Masayuki Yamauchi (Hiroshima Institute of Technology), Mamoru Tanaka (Sophia University)

A1L-C5  **Inter-Connection of Parallel Connected Class D Amplifiers Operated at Different Switching Frequencies**
Yusuke Ishikawa (Fukuoka University), Ryuta Yamamoto (Fukuoka University), Hiroyuki Uchiyama (Fukuoka University), Xiuqin Wei (Chiba University), Hiroo Sekiya (Chiba University), Tadashi Suetsugu (Fukuoka University)

**A1L-D System Analysis and Modeling**

DATE: September 6, 10:50–12:30  
ROOM: Room D  
Chair: Takashi Hikihara (Kyoto University)

A1L-D1  **Stochastic Resonance in a Simple Electric Circuit Having a Double-Well Potential**
Akira Utagawa (Hokkaido University), Tetsuya Asai (Hokkaido University), Yoshihito Amemiya (Hokkaido University)

A1L-D2  **New Idea of the Pseudo-Inverse Maps in Optimal Pre-Correction of Nonlinear Systems As the Result of Modeling and Optimal Past-Correction**
Grzegorz Ciesielski (Technical University of Lodz), Paulina Sobanska (INWAT Ltd., Research and Design Company, Lodz)

A1L-D3  **Differences Between Theoretical and Measured Spectrum in Systems Employing a Spread-Spectrum Clock for EMI Reduction Purposes**
Fabio Pareschi (Universita degli studi di Ferrara), Gianluca Setti (Universita degli studi di Ferrara), Riccardo Rovatti (Universitdi Bologna), Giovanni Frattini (National Semiconductor)

A1L-D4  **Rigorous Parameter Estimation for Noisy Mixed-Effects Models**
Alexander Danis (Uppsala University), Andrew Hooker (Uppsala University), Warwick Tucker (Uppsala University)

A1L-D5  **Evolution of Density of States for Delay Blood Cell Production Model**
Pawel Jozef Mitkowski (AGH University of Science and Technology), Maciej Ogorzalek (Jagiellonian University)

**A2L-A [Special Session] Evolving Complex Networks and their Applications - 2**

DATE: September 6, 14:00–15:20  
ROOM: Room A
A2L-A1 Generalized Modeling of Heterogeneous Nonlinear Networks 75
Thilo Gross (Max-Planck Institute for the Physics of Complex Systems), Dirk Stiefs (Max-Planck Institute for the Physics of Complex Systems), Lars Rudolf (Max-Planck Institute for the Physics of Complex Systems), Martin Zumsande (Max-Planck Institute for the Physics of Complex Systems)

A2L-A2 Hierarchical and Modular Organization of Corticocortical Networks Supports Functional Integration and Segregation in the Mammalian Brain 79
Gorka Zamora-López (University of Potsdam), Changsong Zhou (Hong Kong Baptist University), Jürgen Kurths (Potsdam Institute for Climate Impact Research)

A2L-A3 Recurrence Network Approach to a Phase Space of a Time-Delay System 83
Dharmapuri Vijayan Senthilkumar (Potsdam Institute for Climate Impact Research), Norbert Marwan (Potsdam Institute for Climate Impact Research), Jürgen Kurths (Potsdam Institute for Climate Impact Research)

A2L-A4 Recurrence-Based Evolving Networks for Time Series Analysis of Complex Systems 87
Reik V. Donner (Potsdam Institute for Climate Impact Research), Jonathan F. Donges (Potsdam Institute for Climate Impact Research), Yong Zou (Potsdam Institute for Climate Impact Research), Norbert Marwan (Potsdam Institute for Climate Impact Research), Jürgen Kurths (Potsdam Institute for Climate Impact Research)

A2L-B [Special Session] Hybrid Manycore Architectures - Capabilities and Limitations - 1

DATE: September 6, 14:00–15:20
ROOM: Room B
Chair: Peter Szolgay (Hungarian Academy of Sciences)

A2L-B1 Clustering CNN Devices for Smart Networks 91
Lambert Spaanenburg (Lund University), Suleyman Malki (Lund University)

A2L-B2 Hierarchical Feature Extraction for Dynamic Feature and Signature Tracking 95
Vilmos Szabo (Pazmany Peter Catholic University), Csaba Rekeczky (Eutecus Inc.)

A2L-B3 Mapping of High Performance Data-Flow Graphs Into Programmable Logic Devices 99
Csaba Nemes (Péter Pázmány Catholic University), Zoltán Nagy (Hungarian Academy of Sciences), Miklós Ruszinkó (Hungarian Academy of Sciences), András Kiss (Péter Pázmány Catholic University)

A2L-B4 An Improved Emulated Digital CNN Architecture for High Performance FP-GAs 103
László Füredi (Péter Pázmány Catholic University), Zoltán Nagy (Hungarian Academy of Sciences), András Kiss (Péter Pázmány Catholic University), Péter Szolgay (Péter Pázmány Catholic University)
A2L-C [Special Session] Pattern Formation, Emergence, and Imaging Featuring Nonlinear Dynamics - 1

DATE: September 6, 14:00–15:20
ROOM: Room C
Chair: Ken’ichi Fujimoto (Tokushima University)

**A2L-C1** Fractal Basins and Boundaries in 2D Maps Inspired in Discrete Population Models
   Daniele Fournier-Prunaret (LATTIS-INSA, LAAS-CNRS), Ricardo Lopez-Ruiz (University of Zaragoza)

**A2L-C2** Accurate Formulas Locating Unstable Periodic Points in Chaos
   Tetsushi Ueta (Tokushima University), Kei Nagao (Tokushima University)

**A2L-C3** Stability of a Switched System for Continuous-Time Tomographic Image Reconstruction
   Omar Abou Al-ola (University of Tokushima), Ken’ichi Fujimoto (University of Tokushima), Tetsuya Yoshinaga (University of Tokushima)

A2L-D Neural Networks 1

DATE: September 6, 14:00–15:20
ROOM: Room D
Chair: Yoshihiko Horio (Tokyo Denki University)

**A2L-D1** Neuronal Avalanches Induced by Spike-Timing-Dependent Plasticity
   Shuhei Ohno (Saitama University), Hideyuki Kato (Saitama University), Tohru Ikeguchi (Saitama University)

**A2L-D2** Estimation of Neural Network Structure by Transforming Spike Sequences to Continuous Time Series
   Kaori Kuroda (Saitama University), Tohru Ikeguchi (Saitama University)

**A2L-D3** Improvement of Accuracy and Processing Speed of a Maximum Neural Network Algorithm for the Channel Assignment Problem
   Kazunori Nemoto (University of Aizu), Junji Kitamichi (University of Aizu)

**A2L-D4** Effect of Piecewise Linear Function on Maximum-Flow Neural Network
   Masatoshi Sato (Sophia University), Hisashi Aomori (Tokyo University of Science), Mamoru Tanaka (Sophia University)
A3L-A [Special Session] Verified Numerical Computations for Linear and Nonlinear Problems

DATE: September 6, 15:40–17:00  
ROOM: Room A  
Chair: Katsuhisa Ozaki (Shibaura Institute of Technology)

Akitoshi Takayasu (Waseda University), Shin’ichi Oishi (Waseda University), Takayuki Kubo (University of Tsukuba)

A3L-A2 Verified Bounds for Singular Values, in Particular for the Spectral Norm of a Matrix and its Inverse  
Siegfried Rump (Technische Universität Hamburg)

A3L-A3 Accurate Matrix Singular Values  
Takeshi Ogita (Tokyo Woman’s Christian University)

A3L-A4 Condition Numbers of Two-Dimensional Orientation Problem  
Katsuhisa Ozaki (Shibaura Institute of Technology), Takeshi Ogita (Tokyo Woman’s Christian University), Shin’ichi Oishi (Waseda University)

A3L-B [Special Session] Hybrid Manycore Architectures - Capabilities and Limitations - 2

DATE: September 6, 15:40–17:00  
ROOM: Room B  
Chair: Peter Szolgay (Hungarian Academy of Sciences)

A3L-B1 GPGPU Accelerated Scene Segmentation Using Nonparametric Clustering  
Balazs Varga (Péter Pázmány Catholic University), Kristof Karacs (Péter Pázmány Catholic University)

A3L-B2 Architecture of the Next Generation Real Time CNN Processor: RTCNNP-v2  
Evren Cesur (Yildiz Technical University), Nerhun Yildiz (Yildiz Technical University), Vedat Tavsanoglu (Yildiz Technical University)

A3L-B3 Hardware-Software Co-Design of Nonlinear Active Wave Generator with Microblaze Soft Core Processor  
Selman Ergunay (Istanbul Technical University), Ramazan Yeniceri (Istanbul Technical University), Mustak Erhan Yalcin (Istanbul Technical University)

A3L-B4 Standard C++ Compiling to GPU with Lambda Functions  
Adam Rak (Péter Pázmány Catholic University), Gergely Feldhoffer (Péter Pázmány Catholic University), Gergely Balázs Soós (StreamNovation Ltd.), György Cserey (Péter Pázmány Catholic Univer-
A3L-C [Special Session] Pattern Formation, Emergence, and Imaging Featuring Nonlinear Dynamics - 2

DATE: September 6, 15:40–17:00
ROOM: Room C
Chair: Tetsushi Ueta (Tokushima University)

**A3L-C1 Suppression of Nonlinear Traffic Pattern Generation Using Dynamic Bandwidth Management Method**
Shigeaki Tanimoto (Chiba Institute of Technology), Yohsuke Kinouchi (University of Tokushima)

**A3L-C2 Extraction of Image Regions Using Oscillatory Responses in Chaotic Neuronal Network**
Mio Musashi (University of Tokushima), Ken’ichi Fujimoto (University of Tokushima), Tetsuya Yoshinaga (University of Tokushima)

**A3L-C3 Dynamical Micro-Bead Pattern Forming Using Laser Manipulation Techniques**
Yoshio Tanaka (National Institute of Advanced Industrial Science and Technology), Shogo Tsutsui (Kagawa University), Mitsuru Ishikawa (National Institute of Advanced Industrial Science and Technology), Hiroyuki Kitajima (Kagawa University)

A3L-D Neural Networks 2

DATE: September 6, 15:40–17:00
ROOM: Room D
Chair: Tohru Ikeguchi (Saitama University)

**A3L-D1 Support Vector Machines with Online Unsupervised Learning Method and its Application to Surface-Electromyogram Recognition Problems**
Hiroki Tamura (University of Miyazaki), Takeshi Yoshimatu (University of Miyazaki), Koichi Tanno (University of Miyazaki)

**A3L-D2 Kohonen Feature Map Probabilistic Associative Memory Based on Weights Distribution and Area Neuron Increase and Decrease**
Takahiro Hada (Tokyo University of Technology), Yuko Osana (Tokyo University of Technology)

**A3L-D3 Reinforcement Learning Using Improved Kohonen Feature Map Probabilistic Associative Memory Based on Weights Distribution**
Shingo Noguchi (Tokyo University of Technology), Yuko Osana (Tokyo University of Technology)

**A3L-D4 Multi-Layer Perceptron Having Neuro-Glia Network**
Chihiro Ikuta (Tokushima University), Yoko Uwate (Tokushima University), Yoshifumi Nishio (Tokushima University)
B1L-A [Special Session] Aspects of Optimization with Nonlinear Dynamics - 1

DATE: September 7, 09:00–10:40
ROOM: Room A
Chair: Mikio Hasegawa (Tokyo University of Science)

**B1L-A1** Canonical Particle Swarm Optimization System  
Kenya Jin’no (Nippon Institute of Technology), Takuya Shindo (Nippon Institute of Technology)

**B1L-A2** Particle Swarm Optimization with Novel Concept of Complex Network  
Haruna Matsushita (Hosei University), Yoshifumi Nishio (Tokushima University), Toshimichi Saito (Hosei University)

**B1L-A3** Application of Particle Swarm Optimizers to Finding Desired Parameters of Switched Dynamical Systems  
Haruna Matsushita (Hosei University), Toshimichi Saito (Hosei University)

**B1L-A4** Dynamical Noise Injection to Chaotic Dynamics for Solving Combinatorial Optimization Problems  
Takayuki Suzuki (Saitama University), Takaufumi Matsuura (Tokyo University of Science), Tohru Ikeguchi (Saitama University)

B1L-B [Special Session] Nonlinear Analysis and Processing of Facial Images

DATE: September 7, 09:00–10:40
ROOM: Room B
Chair: Hironobu Fukai (Ritsumeikan University)

**B1L-B1** Facial Expression Recognition Using a Simplified Head Model and RBF Networks  
Koichi Takahashi (Tokyo University of Agriculture and Technology), Hironobu Fukai (Ritsumeikan University), Yasue Mitsukura (Tokyo University of Agriculture and Technology)

**B1L-B2** Facial Feature Animation and its Artistic Representation  
Yang Yang (Xi’an Jiaotong University/Tokushima University), Yuanqi Su (Xi’an Jiaotong University), Yuehu Liu (Xi’an Jiaotong University), Yoshifumi Nishio (Tokushima University)

**B1L-B3** An Embedding and Detection Method of Invisible Calibration Pattern for Print-Type Data Hiding  
Hironori Takimoto (Okayama Prefectural University), Seiki Yoshimori (Nippon Bunri University), Yasue Mitsukura (Tokyo University of Agriculture and Technology), Minoru Fukumi (University of Tokushima)

**B1L-B4** Age Estimation Using Kernel Regression Analysis  
Hironobu Fukai (Ritsumeikan University), Hironori Takimoto (Okayama Prefectural University),
B1L-C Complex Systems

DATE: September 7, 09:00–10:40
ROOM: Room C
Chair: Guanrong Chen (City University of Hong Kong)

**B1L-C1 A Basic Fuzzy-Estimation Theory for Available Operation of Complicated Large-Scale Network Systems**

Kazuo Horiuchi (Waseda University)

**B1L-C2 Potential Games Based Coverage Control with Voronoi Partition**

Saori Teraoka (Osaka University), Toshimitsu Ushio (Osaka University), Takafumi Kanazawa (Osaka University), Naoki Hayashi (Osaka University)

**B1L-C3 Convergence Analysis of Discrete-Time Multi-Agent Systems Based on Sequential Connectivity**

Yao Chen (Academy of Mathematics and Systems Science, Chinese Academy of Sciences), Jinhu Lü (Academy of Mathematics and Systems Science, RMIT University), Daniel W. c. Ho (City University of Hong Kong), Xinghuo Yu (RMIT University)

**B1L-C4 Equatorial Climate Data Analysis and Forecasting by Singular Spectrum Analysis**

Naoki Itoh (University of Potsdam), Jügen Kurths (Potsdam Institute for Climate Impact Research)

**B1L-C5 Directional Spike Propagation by Anisotropic Inhibitory Connections Modulated Through STDP in a Recurrent Network**

Toshikazu Samura (Kyushu Institute of Technology), Hatsuo Hayashi (Kyushu Institute of Technology)

B1L-D Bifurcation and Chaos

DATE: September 7, 09:00–10:40
ROOM: Room D
Chair: Keiji Konishi (Osaka Prefecture University)

**B1L-D1 Delay and Release System by Utilized Lorenz Attractor**

Yoshiyuki Kobori (Sophia University), Masatoshi Sato (Sophia University), Mamoru Tanaka (Sophia University)

**B1L-D2 Numerical Calculation Method of Characteristic Multiplier for the Fixed Point in a Rigid Overhead Wire-Pantograph System**

Shota Hirashima (Oita University), Shu Karube (Oita National College of Technology), Takuji Kousaka (Oita University)
B1L-D3 Nonlinear Dynamics in Buck-Boost Converter with Spike Noise
Hiroyuki Asahara (Oita University), Takuji Kousaka (Oita University)

B1L-D4 Decomposition of Symmetric Almost Periodic Oscillation in Three-Phase Circuit
Takashi Hisakado (Kyoto University), Shota Ukai (Kyoto University)

B1L-E Circuit Implementation

DATE: September 7, 09:00–10:40
ROOM: Room E
Chair: Tetsuya Asai (Hokkaido University)

B1L-E1 A Mathematical-Structure-Based aVLSI Silicon Neuron Model
Takashi Kohno (University of Tokyo), Kazuyuki Aihara (University of Tokyo)

B1L-E2 Circuit Implementation of an A/D Converter Based on the Negative $\beta$-Map with a Discrete-Time Integrator
Yoshihiko Horio (Tokyo Denki University), Kenya Jin’no (Nippon Institute of Technology), Tohru Kohda (Kyushu University), Kazuyuki Aihara (University of Tokyo)

B1L-E3 Digital-Signal-Waveform Improvement for High-Speed VLSI Packaging
Moritoshi Yasunaga (University of Tsukuba), Hiroshi Nakayama (University of Tsukuba), Yuki Shimauchi (University of Tsukuba), Ikuo Yoshihara (Miyazaki University)

B1L-E4 Realization of Three-Dimensional DT-CNN on FPGA
Nguyen Tien Dat (Hanoi University of Science and Technology), Nguyen Tien Dzung (Hanoi University of Science and Technology), Thang Manh Hoang (Hanoi University of Science and Technology)

B1L-E5 Implementation of CNN-Based FFT/IFFT Algorithms on FPGA
Hiep Hoang Le (Hanoi University of Science and Technology), Nguyen Tien Dzung (Hanoi University of Science and Technology), Thang Manh Hoang (Hanoi University of Science and Technology)

P2L-A [Plenary Talk]

DATE: September 7, 11:10–12:10
ROOM: Room A
Chair: Zbigniew Galias (AGH University)

P2L-A1 Advanced PI/SI/EMI Simulation Technology for High-Speed Electronic Design
Hideki Asai (Shizuoka University)

DATE: September 7, 13:40–15:20
ROOM: Room A
Chair: Kenya Jin’no (Nippon Institute of Technology)

B2L-A1 Effects of Chaotic Fluctuation on the Hopfield Model for the Traveling Salesman Problem 285
Ayano Hirahara (Tokyo University of Science), Osamu Araki (Tokyo University of Science)

B2L-A2 Realizing Ideal Chaotic Dynamics for Combinatorial Optimization Using a Spatiotemporal Filter 289
Mikio Hasegawa (Tokyo University of Science)

B2L-A3 A Method for Solving Very Large Scale TSPs by Chaotic Dynamics 293
Tohru Ikeguchi (Saitama University), Shun Motohashi (Saitama University), Takafumi Matsuura (Tokyo University of Science)

Toshihiro Tachibana (Tokyo Denki University), Masaharu Adachi (Tokyo Denki University)

B2L-A5 Double-Assignment Method Driven by Chaotic Neurodynamics for Quadratic Assignment Problems 301
Kazuaki Shibata (Tokyo Denki University), Yoshihiko Horio (Tokyo Denki University)

B2L-B [Special Session] Complex Networks and their Dynamics - 1

DATE: September 7, 13:40–15:20
ROOM: Room B
Chairs: Kohshi Okumura (Simon Fraser University) and Ljiljana Trajkovic (Simon Fraser University)

B2L-B1 Spectral Analysis and Dynamical Behavior of Complex Networks 305
Kohshi Okumura (Simon Fraser University), Ljiljana Trajkovic (Simon Fraser University)

B2L-B2 Complex Congestion Behavior in Deflection Routing 309
Wilson Wang kit Thong (City University of Hong Kong), Guanrong Chen (City University of Hong Kong)

B2L-B3 Scale-Free Property of Affordable Neural Network 310
Yoshifumi Nishio (Tokushima University), Yoko Uwate (Tokushima University)

B2L-B4 Dynamical AD Converter by Saito’s Rotation Map in Functional Cellular Neural Sensor Network 314
Mamoru Tanaka (Sophia University), Toshimichi Saito (Hosei University)
### B2L-C Communication Systems

**DATE:** September 7, 13:40–15:20  
**ROOM:** Room C  
**Chair:** Gianluca Setti (Università degli studi di Ferrara)

<table>
<thead>
<tr>
<th>Session</th>
<th>Title</th>
<th>Authors</th>
</tr>
</thead>
<tbody>
<tr>
<td>B2L-C1</td>
<td>FCC-Compliant Operation of Low-Rate UWB Impulse Radio Applying Multiple Pulses Per Bit</td>
<td>Tamás Krébesz (Péter Pázmány Catholic University), Géza Kolumbán (Péter Pázmány Catholic University), Chi K. Tse (Hong Kong Polytechnic University), Francis C. M. Lau (Hong Kong Polytechnic University)</td>
</tr>
<tr>
<td>B2L-C2</td>
<td>Radio Coverage Extension of the FCC-Compliant Low-Rate UWB Networking Devices</td>
<td>Géza Kolumbán (Péter Pázmány Catholic University), Tamás Krébesz (Péter Pázmány Catholic University), Francis C. M. Lau (Hong Kong Polytechnic University), Chi K. Tse (Hong Kong Polytechnic University)</td>
</tr>
<tr>
<td>B2L-C3</td>
<td>Application of Open-Plus-Closed-Loop Control to Secure Communications Using Chaos Masking</td>
<td>Yusuke Yoshida (Ritsumeikan University), Takaya Miyano (Ritsumeikan University)</td>
</tr>
<tr>
<td>B2L-C4</td>
<td>A Constructive Approach for the Design of Finite Time Self-Synchronizing Coupled Systems with Unknown Inputs</td>
<td>Gilles Millerioux (Nancy University)</td>
</tr>
<tr>
<td>B2L-C5</td>
<td>CA-Based Stream Cryptography with Variable-Length Key</td>
<td>Masaki Igarashi (Hokkaido University), Masayuki Ikebe (Hokkaido University), Junichi Motohisa (Hokkaido University)</td>
</tr>
</tbody>
</table>

### B2L-D Chaotic Circuits

**DATE:** September 7, 13:40–15:20  
**ROOM:** Room D  
**Chairs:** Arunas Tamasevicius (Center for Physical Sciences and Technology)

<table>
<thead>
<tr>
<th>Session</th>
<th>Title</th>
<th>Authors</th>
</tr>
</thead>
<tbody>
<tr>
<td>B2L-D1</td>
<td>Horseshoe Chaos in the Nano-Electro-Mechanical Casimir Nonlinear Oscillator</td>
<td>Michele Bonnin (Politecnico di Torino), Pier Paolo Civalleri (Politecnico di Torino), Marco Gilli (Politecnico di Torino)</td>
</tr>
<tr>
<td>B2L-D2</td>
<td>Forced Chaos Generator with CMOS Variable Active Inductor Circuit</td>
<td>Yusuke Tsubaki (Tokyo Denki University), Munehisa Sekikawa (Tokyo University), Yoshihiko Horio (Tokyo Denki University)</td>
</tr>
<tr>
<td>B2L-D3</td>
<td>Chaos in Chua’s Oscillator with Chua’s Diode and Memristor</td>
<td>Tuyen Tran Xuan (Hanoi University of Science and Technology), Nguyen Tien Dzung (Hanoi University of Science and Technology), Thang Manh Hoang (Hanoi University of Science and Technology)</td>
</tr>
</tbody>
</table>
Chaos Synchronization in Coupled Delayed Two-Stage Colpitts Circuits for UWB Communications
Hien Quang Vu (Hanoi University of Science and Technology), Nguyen Tien Dzung (Hanoi University of Science and Technology), Vu Van Yem (Hanoi University of Science and Technology), Thang Manh Hoang (Hanoi University of Science and Technology)

An Energy Based Investigation of Rössler Type Chaos on Chua’s Circuit
Mustafa Kösem (Istanbul Technical University), Neslihan Serap Sengör (Istanbul Technical University)

B2L-E Circuit Analysis

DATE: September 7, 13:40–15:20
ROOM: Room E
Chair: Roni Khazaka (McGill University)

Global Asymptotic Stability Analysis of Nonlinear Circuits for Solving the Maximum Flow Problem
Norikazu Takahashi (Kyushu University)

Analysis and Design of Class-DE Amplifier with Nonlinear Shunt Capacitances at Any Duty Ratio
Hiroo Sekiya (Chiba University), Marian Kazimierczuk (Wright State University)

Soft Fault Diagnosis of Nonlinear Analog Circuits Using the Continuation Approach
Michal Tadeusiewicz (Technical University of Lodz), Stanislaw Halgas (Technical University of Lodz)

Single-Tone Moments Based Adjoint Sensitivity Analysis of Nonlinear Intermodulation Distortion in RF Circuits
Dani Tannir (McGill University), Roni Khazaka (McGill University)

SPICE-Oriented Algorithm for Assessment of Stability for Periodic Solutions
Hiroshige Kataoka (Tokushima University), Yoshihiro Yamagami (Tokushima University), Yoshi-fumi Nishio (Tokushima University), Akio Ushida (Tokushima University)

B3L-A [Special Session] Aspects of Optimization with Nonlinear Dynamics - 3

DATE: September 7, 15:50–17:30
ROOM: Room A
Chair: Masaharu Adachi (Tokyo Denki University)

A Combinatorial Optimization Method Which Combines Ant Colony Optimization and Chaotic Dynamical
B3L-A2  Slide-and-Insert Assignment Method with Chaotic Dynamics for Quadratic Assignment Problems
Yusuke Sakamoto (Tokyo Denki University), Yoshihiko Horio (Tokyo Denki University)

B3L-A3  Heuristics Methods for Asymmetric Traveling Salesman Problem and Their Applications to DNA Fragment Assembly
Tomohiro Kato (Tokyo University of Science), Mikio Hasegawa (Tokyo University of Science)

B3L-A4  A Relay Sensor Node Selection Scheme in Wireless Sensor Networks Using a Chaotic Neural Network
Kyohei Fujii (Tokyo City University), Tomoyuki Sasaki (Tokyo City University), Hidehiro Nakano (Tokyo City University), Akihide Utani (Tokyo City University), Arata Miyauchi (Tokyo City University), Hisao Yamamoto (Tokyo City University)

B3L-A5  Solving a Sink Node Allocation Problem in Wireless Sensor Networks Using a Competitive Particle Swarm Optimization
Yuta Kanamori (Tokyo City University), Yu Taguchi (Tokyo City University), Hidehiro Nakano (Tokyo City University), Akihide Utani (Tokyo City University), Arata Miyauchi (Tokyo City University), Hisao Yamamoto (Tokyo City University)

B3L-B [Special Session] Complex Networks and their Dynamics - 2
DATE: September 7, 15:50–17:30
ROOM: Room B
Chairs: Kohshi Okumura (Simon Fraser University) and Ljiljana Trajkovic (Simon Fraser University)

B3L-B1  A Brief Overview of Some Recent Advances in Pinning Control of Complex Networks
Jinhu Lu (RMIT University), Xinghuo Yu (RMIT University), David Hill (Australian National University)

B3L-B2  A Complex Network Perspective to Volatility in Stock Markets
Xiao Liu (Hong Kong Polytechnic University), Chi K. Tse (Hong Kong Polytechnic University)

B3L-B3  Property of the Chaotic Propagating Pulse Wave in a Ring of Coupled Bistable Oscillators
Kyohei Kamiyama (Meiji University), Tetsuro Endo (Meiji University), Kuniyasu Shimizu (Chiba Institute of Technology), Hiroyuki Kamata (Meiji University)

B3L-B4  Self-Organized Behaviors in an Adaptive Network of Movable Oscillators
Takaaki Aoki (Kyoto University), Toshio Aoyagi (Kyoto University)

B3L-C Time Sequences
DATE: September 7, 15:50–17:30
ROOM: Room C
Chair: Masayuki Ikebe (Hokkaido University)

B3L-C1  On Auto-Correlation Values of De Bruijn Sequences  414
Hiroshi Fujisaki (Kanazawa University), Yuhki Nabeshima (Kanazawa University)

B3L-C2  A Perturbation-Based Algorithm with Extremely Long Periods of Generated Cycles  418
Mieczyslaw Jessa (Poznan University of Technology)

B3L-C3  Spreading Sequences with Negative Auto-Correlation Based on Chaos Theory and Gold Sequences -Increase of Family Sizes and Performance Evaluation-  422
Akio Tsuneda (Kumamoto University), Taizo Sagara (Kumamoto University)

B3L-C4  Is Markov Code Superior to i.i.d. in Communication Systems?  426
Tohru Kohda (Kyushu University), Yutaka Jitsumatsu (Kyushu University), Kazuyuki Aihara (University of Tokyo)

B3L-C5  Applications of Recurrence Plots in Road Traffic Analysis  430
Tomasz Grabowski (AGH University of Science and Technology), Maciej Ogorzalek (Jagiellonian University)

B3L-D Control and Robotics

DATE: September 7, 15:50–17:30
ROOM: Room D
Chair: Yuzo Ohta (Kobe University)

B3L-D1  Discrete Gait Generation for the Compass-Type Biped Robot Modeled by Discrete Mechanics  434
Tatsuya Kai (Osaka University), Takeshi Shintani (Osaka University)

B3L-D2  Attitude Stabilization Control of 3D Space Robot Model with Initial Angular Momentum via Model Predictive Control  438
Tatsuya Kai (Osaka University), Katsunori Kondoh (Osaka University)

B3L-D3  Adaptive Stabilization of a Saddle Steady State of a Conservative Dynamical System: a Spacecraft at the Lagrange Point L2 of the Sun-Earth System  442
Elena Tamaseviciute (Swiss Federal Institute of Technology in Zurich), Gytis Mykolaitis (Center for Physical Sciences and Technology), Skaidra Bumeliene (Center for Physical Sciences and Technology), Arunas Tamasevicius (Center for Physical Sciences and Technology)

B3L-D4  Suppression of Spatio-Temporal Chaos in Excitable Media with Nonexcitable Cells  446
Masaki Irisawa (Osaka Prefecture University), Keiji Konishi (Osaka Prefecture University), Naoyuki Hara (Osaka Prefecture University), Hideki Kokame (Osaka Prefecture University)

B3L-D5  Optimal Backstepping Control for Genesio-Tesi Chaotic System Using Genetic Algorithm  450
Mohammad Reza Modabbernia (Islamic Azad University, Lahijan Branch), Ali Reza Sahab (Islamic Azad University, Lahijan Branch), Masoud Taleb Ziabari (Islamic Azad University, Qazvin Branch),
C1L-A [Special Session] Nonlinear Time Series Analysis - 1

DATE: September 8, 09:00–10:40
ROOM: Room A
Chairs: Max Little (University of Oxford) and Michael Small (Hong Kong Polytechnic University)

C1L-A1 Recovering Piecewise Constant Signals from Noisy Time Series
Max Little (University of Oxford), Nick Jones (University of Oxford)

C1L-A2 Multivariate Synchronization Analysis of EEG Recordings from Epilepsy Patients
Ralph Gregor Andrzejak (Universitat Pompeu Fabra), Georgia Emmanouil Polychronaki (National Technical University of Athens), Andreas Schulze-Bonhage (Epilepsy Center, University Hospital), Konstantina Spilios Nikita (National Technical University of Athens)

C1L-A3 New Nonlinear Markers and Insights Into Speech Signal Degradation for Effective Tracking of Parkinson’s Disease Symptom Severity
Athanasios Tsanas (University of Oxford), Max Little (University of Oxford), Patrick Mcsharry (University of Oxford), Lorraine Ramig (University of Colorado / National Center for Voice and Speech)

C1L-A4 Detecting System State Transitions in Environmental Time-Series Using Nonlinear Time Series Analysis
Theodoros Karakasidis (University of Thessaly), Athanasios Fragkou (University of Thessaly), Antonios Liakopoulos (University of Thessaly)

C1L-B Image and Signal Processing

DATE: September 8, 09:00–10:40
ROOM: Room B
Chair: Mio Kobayashi (Anan National College of Tech.)

C1L-B1 Associative Dynamics of Color Images in a Chaotic Neural Network
Makito Oku (University of Tokyo), Kazuyuki Aihara (University of Tokyo)

C1L-B2 Nonlinear Image Processing for Multiple Object Tracking on Cellular Hardware Platform
Takao Matsui (Ritsumeikan University), Tomohiro Fujita (Ritsumeikan University), Mamoru Nakanishi (Ritsumeikan University), Takeshi Ogura (Ritsumeikan University)

C1L-B3 New Method of Sequential Symbolic Analysis of Biomedical Signals
Robert Stepien (Nalecz Institute of Biocybernetics and Biomedical Engineering PAS), Wlodzimierz Klonowski (Nalecz Institute of Biocybernetics and Biomedical Engineering PAS)

C1L-B4 Speaker Identification with Voiced Speech Variability Modeling Using Phase Space Reconstruction
C1L-C [Special Session] Nonlinear Maps and Applications - 1

DATE: September 8, 09:00–10:40
ROOM: Room C
Chair: Daniele Fournier-Prunaret (INSA, Toulouse)

C1L-C1 Basic Learning Characteristics of Digital Spike Maps
Takashi Ogawa (Hosei University), Toshimichi Saito (Hosei University)

C1L-C2 Bifurcation Analysis of Coupled Nagumo-Sato Models
Kazutoshi Kinoshita (Tokushima University), Tetsushi Ueta (Tokushima University), Jun’ichi Imura (Tokyo Institute of Technology), Kazuyuki Aihara (University of Tokyo)

C1L-C3 Synchronization Phenomena of Globally Coupled Logistic Maps with Time-Varying Parameters
Hironori Kumeno (Tokushima University), Yoshifumi Nishio (Tokushima University), Daniele Fournier-Prunaret (LATTIS-INSA, LAAS-CNRS)

C1L-C4 Synchronization in Coupled Maps with Triangular Networks
Yoko Uwate (Tokushima University), Yoshifumi Nishio (Tokushima University)

C1L-C5 Analysis of Several Spatio-Temporal Phase Patterns in Coupled Chaotic Maps by Varying Coupling Strength
Masahiro Wada (Konan University), Takuya Fukuda (Konan University)

C1L-D Bio-Inspired Algorithms

DATE: September 8, 09:00–10:40
ROOM: Room D
Chair: Hidehiro Nakano (Tokyo City University)

C1L-D1 Ant Colony Optimization with Intelligent and Dull Ants
Sho Shimomura (Tokushima University), Masaki Sugimoto (Tokushima University), Taku Haraguchi (Tokushima university), Haruna Matsushita (Hosei university), Yoshifumi Nishio (Tokushima University)

C1L-D2 Comparison of the Linear Algebra Approach and the Evolutionary Computing for Magnetic Field Shaping in Linear Coils
Bartlomiej Garda (AGH-University of Science and Technology), Zbigniew Galias (AGH-University of Science and Technology)
Office Layout Support System Using Genetic Algorithm - Generation of Room Arrangement Plans for Polygonal Space -
Ryota Tachikawa (Tokyo University of Technology), Yuko Osana (Tokyo University of Technology)

Design of Class E Amplifier Using Particle Swarm Optimization
Yuichi Tanji (Kagawa University), Hiroo Sekiya (Chiba University)

Improvement of Tug-of-War Model for Two-Armed Bandit Problem: Biologically Inspired Computing Method for Nonlocally-Correlated Parallel Searches
Song-Ju Kim (Advanced Science Institute, RIKEN), Masashi Aono (Advanced Science Institute, RIKEN), Masahiko Hara (Advanced Science Institute, RIKEN)

P3L-A [Plenary Talk]

DATE: September 8, 11:10–12:10
ROOM: Room A
Chair: Yoshifumi Nishio (Tokushima University)

Asymptotic behaviour of blinking (stochastically switched) dynamical systems
Martin Hasler (EPFL)


DATE: September 8, 13:40–15:00
ROOM: Room A
Chairs: Max Little (University of Oxford) and Michael Small (Hong Kong Polytechnic University)

Entropy-Based Measures of Causality and Application to Epilepsy
Dimitris Kugiumtzis (Aristotle University of Thessaloniki)

Nonlinear and Nonparametric Models for Forecasting the US Gross National Product
Siddharth Arora (University of Oxford), Max Little (University of Oxford), Patrick Mcsharry (University of Oxford)

Using Permutation Complexity Tools to Analyze Complex Spatiotemporal Dynamics
José M. Amigó (Universidad Miguel Hernández), Samuel Zambrano (Universidad Rey Juan Carlos), Miguel A. f. Sanjuán (Universidad Rey Juan Carlos)
C2L-B [Special Session] A Nonlinear Dynamics Perspective of Cellular Automata - 1

DATE: September 8, 13:40–15:00
ROOM: Room B
Chairs: Giovanni Pazienza (MTA-SZTAKI and Pazmany University) and Tamás Roska (Hungarian Academy of Sciences)

C2L-B1 Robust and Non-Robust Omega-Limit Orbits in 1D Cellular Automata
Giovanni E. Pazienza (MTA-SZTAKI and Péter Pázmány Catholic University)

C2L-B2 Uncertainty Profiles for Predicting Complex Nonlinear Dynamics in Cellular Automata: the Case of Five Cells Neighborhood
Radu Dogaru (University Politehnica of Bucharest), Ioana Dogaru (University Politehnica of Bucharest)

C2L-C [Special Session] Nonlinear Maps and Applications - 2

DATE: September 8, 13:40–15:00
ROOM: Room C
Chair: Tetsushi Ueta (Tokushima University)

C2L-C1 Border Collision Bifurcations in a Simple Switching Circuit
Daniele Fournier-Prunaret (LATTIS-INSA, LAAS-CNRS), Laura Gardini (Universitdegli studi di Urbino Carlo Bo), Pascal Charge (LATTIS-INSA, LAAS-CNRS)

C2L-C2 Analysis of Spike-Trains from Simple Resonate-and-Fire Chaotic Circuit
Satoshi Imai (Hosei University), Toshimichi Saito (Hosei University)

C2L-C3 Switched Systems and Applications to Mutual Synchronization
Andrea Espinel (IRCCyN, Ecole Centrale de Nantes), Ina Taralova (IRCCyN, Ecole Centrale de Nantes)

C2L-C4 Synthesis of a Spiking Oscillator with a Desired Inter-Spike-Interval Density
Tadashi Tsubone (Nagaoka University of Technology)

C2L-D Coupled Oscillators 1

DATE: September 8, 13:40–15:00
ROOM: Room D
Chair: Takashi Hisakado (Kyoto University)

C2L-D1 Exponential Transient Oscillations and Standing Pulses in Rings of Coupled Symmetric Bistable Maps
Yo Horikawa (Kagawa University)
C2L-D2 Clustering Synchronization in Pulse-Coupled Oscillators with a Refractory Period and Frequency Distribution
Takuya Okuda (Osaka Prefecture University), Keiji Konishi (Osaka Prefecture University), Naoki Hara (Osaka Prefecture University), Hideki Kokame (Osaka Prefecture University)

C2L-D3 Statistical Characters of Synchronization-Optimized Oscillator Networks
Tatsuo Yanagita (Hokkaido University), Alexander S. Mikhailov (Fritz-Haber-Institut der Max-Planck-Gesellschaft)

C2L-D4 An Efficient Algorithm for the Evaluation of Master Stability Function in Networks of Coupled Oscillators
Marco Righero (Politecnico di Torino), Fernando Corinto (Politecnico di Torino), Mario Biy (Politecnico di Torino)

C3L-A [Special Session] Nonlinear Time Series Analysis - 3
DATE: September 8, 15:20–16:40
ROOM: Room A
Chairs: Max Little (University of Oxford) and Michael Small (Hong Kong Polytechnic University)

C3L-A1 Linearity and Nonlinearity Within Recurrence Plots
Yoshito Hirata (University of Tokyo), Kazuyuki Aihara (University of Tokyo)

C3L-A2 Transformation of Growing Networks to Time Series and its Nonlinear Time Series Analysis
Yutaka Shimada (Saitama University), Yuta Haraguchi (Saitama University), Tohru Ikeguchi (Saitama University)

C3L-A3 Recurrence Based Complex Network Analysis of Cardiovascular Variability Data to Predict Pre-Eclampsia
Norbert Marwan (Potsdam Institute for Climate Impact Research), Niels Wessel (Humboldt University Berlin), Holger Stepan (University of Leipzig), Jürgen Kurths (Potsdam Institute for Climate Impact Research)

C3L-A4 Standard Complex Network Measures of Recurrence-Based Phase Space Networks Constructed from Time Series
Ruoxi Xiang (Hong Kong Polytechnic University), Jie Zhang (Hong Kong Polytechnic University), Michael Small (Hong Kong Polytechnic University)

C3L-B [Special Session] A Nonlinear Dynamics Perspective of Cellular Automata - 2
DATE: September 8, 15:20–16:40
ROOM: Room B
Chairs: Giovanni Pazienza (MTA-SZTAKI and Pazmany University) and Tamás Roska (Hungarian Academy of Sciences)
Isle of Eden in 1D Binary Cellular Automaton As a Manifestation of Gödel Incompleteness and a Proposal for a Bridge Between Analytical Results and Spatial-Temporal Logic Patterns

Tamás Roska (Péter Pázmány Catholic University)

Symbolic Dynamics of Some Bernoulli-Shift Cellular Automata Rules

Guanrong Chen (City University of Hong Kong), Fangyue Chen (Hangzhou Dianzi University), Junbiao Guan (Hangzhou Dianzi University), Weifeng Jin (Zhejiang Chinese Medical University)

Gardens of Eden: Where Nonlinear Dynamics and Formal Languages Meet

Giovanni E. Pazienza (MTA-SZTAKI and Péter Pázmány Catholic University), Marion Oswald (MTA-SZTAKI and Vienna University of Technology)

C3L-C Optimization

DATE: September 8, 15:20–16:40
ROOM: Room C
Chair: Norikazu Takahashi (Kyushu University)

Hard and Fuzzy c-Means Clustering Algorithms with Geodesic Dissimilarity

Yuchi Kanzawa (Shibaura Institute of Technology), Yasunori Endo (University of Tsukuba), Sadaaki Miyamoto (University of Tsukuba)

Kernelized Fuzzy c-Means Clustering for Uncertain Data with L1-Regularization Term of Penalty Vectors Using Explicit Mapping

Yasunori Endo (University of Tsukuba), Isao Takayama (University of Tsukuba), Yukihiro Hamasuna (University of Tsukuba), Sadaaki Miyamoto (University of Tsukuba)

A Classification System Based on Collaboration of Adaptive Resonance Theory Maps and Learning Vector Quantization

Yoko Enosawa (Hosei University), Haruna Matsushita (Hosei University), Toshimichi Saito (Hosei University)

Discrete Higher Order Inverse Function Delayed Network

Takahiro Sota (Tohoku University), Yoshihiro Hayakawa (Sendai National College of Technology), Shigeo Sato (Tohoku University), Koji Nakajima (Tohoku University)

C3L-D Coupled Oscillators 2

DATE: September 8, 15:20–16:40
ROOM: Room D
Chair: Takuji Kousaka (Oita University)

Analyses of Coupled Hindmarsh-Rose Type Bursting Oscillators

Koji Kurose (Tohoku University), Takahiro Sota (Tohoku University), Yoshihiro Hayakawa (Sendai
Spectral Analysis of the Propagating Pulse Wave in 6 Coupled Bistable Oscillators

Kuniyasu Shimizu (Chiba Institute of Technology), Motomasa Komuro (Teikyo University of Science and Technology), Tetsuro Endo (Meiji University)

Bifurcation Analysis of Two Coupled Izhikevich Oscillators

Daisuke Ito (Tokushima University), Tetsushi Ueta (Tokushima University), Kazuyuki Aihara (University of Tokyo)

Amplitude Death in a Pair of Time-Delayed Chaotic Oscillators Coupled by a Static Connection

Luan Ba Le (Osaka Prefecture University), Keiji Konishi (Osaka Prefecture University), Hideki Kokame (Osaka Prefecture University), Naoyuki Hara (Osaka Prefecture University)
Author Index

A
Abou Al-ola, Omar: 115, A2L-C3(xxi)
Aburatani, Hitoshi: 15, A1L-B1(xviii)
Adachi, Masaharu: 297, B2L-A4(xxvii)
Aihara, Kazuyuki: 261, B1L-E1(xxvi), 265, B1L-E2(xxvi), 426, B3L-C4(xxxi), 465, C1L-B1(www), 488, C1L-C2(xxxiii), 577, C3L-A1(xxxvi), 627, C3L-D3(xxxviii)
Almendral, Juan Antonio: 7, A1L-A3(xxvii)
Amemiya, Yoshihito: 55, A1L-D1(xxviii)
Amigó, José M.: 533, C2L-A3(xxxiv)
Andrzejak, Ralph Gregor: 455, C1L-A2(xxxvii)
Aoki, Takaaki: 410, B3L-B4(xxx)
Aomori, Hisashi: 131, A2L-D4(xxvii)
Aono, Masashi: 520, C1L-D5(xxxiv)
Aoyagi, Toshio: 410, B3L-B4(xxx)
Araki, Osamu: 285, B2L-A1(xxvii)
Arora, Siddharth: 529, C2L-A2(xxxiv)
Asahara, Hiroaki: 253, B1L-D3(xxvii)
Asai, Hideki: 281, P2L-A1(xxvi)
Asai, Tetsuya: 55, A1L-D1(xix)

B
Ba Le, Luan: 631, C3L-D4(xxxviii)
Beji, Mario: 573, C2L-D4(xxxvi)
Boccaletti, Stefano: 7, A1L-A3(xxvii)
Bonin, Michele: 338, B2L-D1(xxviii)
Branowski, Lukasz: 476, C1L-B4(xxxii)
Buldú, Javier: 7, A1L-A3(xxvii)
Bumeliene, Skaidra: 442, B3L-D3(xxxi)

C
Cesur, Evren: 153, A3L-B2(xxvii)
Charge, Pascal: 545, C2L-C1(xxxv)
Chen, Fangyue: 595, C3L-B2(xxxvii)
Chen, Yao: 233, B1L-C3(xxv)
Ciesielski, Grzegorz: 59, A1L-D2(xix)
Civalleri, Pier Paolo: 338, B2L-D1(xxviii)

D
Danis, Alexander: 67, A1L-D4(xix)
Dogaru, Ioana: 541, C2L-B2(xxxv)
Dogaru, Radu: 541, C2L-B2(xxxv)
Donner, Reik V.: 87, A2L-A4(xx)

E
Endo, Tetsuro: 406, B3L-B3(xxx), 623, C3L-D2(xxxviii)
Endo, Yasunori: 603, C3L-C1(xxxvii), 607, C3L-C2(xxxvii)
Enosawa, Yoko: 611, C3L-C3(xxxvii)
Ergunay, Selman: 157, A3L-B3(xxii)
Espinel, Andrea: 553, C2L-C3(xxxv)

F
Füredi, László: 103, A2L-B4(xx)
Feldhöffer, Gergely: 161, A3L-B4(xxvii)
Fournier-Prunaret, Daniele: 107, A2L-C1(xxvii), 492, C1L-C3(xxxviii), 545, C2L-C1(xxxv)
Fragkou, Athanasios: 461, C1L-A4(xxxii)
Frattini, Giovanni: 63, A1L-D3(xxiv)
Fujii, Kyohei: 390, B3L-A4(xxxvii)
Fujimoto, Ken’ichi: 115, A2L-C3(xxvii), 169, A3L-C2(xxvii)
Fujisaki, Hiroshi: 414, B3L-C1(xxv)
Fujita, Tomohiro: 27, A1L-B4(xxvii), 469, C1L-B2(xxvii)
Fukai, Hironobu: 209, B1L-B1(xxvii), 221, B1L-B4(xxvii)
Fukuda, Takuya: 500, C1L-C5(xxxxiii)
Fukumoto, Minoru: 217, B1L-B3(www), 221, B1L-B4(xxvii)
G
Galias, Zbigniew: 508, C1L-D2(xxxiii)
Garda, Bartlomiej: 508, C1L-D2(xxxiii)
Gardini, Laura: 545, C2L-C1(xxxiii)
Gilli, Marco: 338, B2L-D1(xxviii)
Goldbeter, Albert: 1, P1L-A1(xvii)
Grabowski, Tomasz: 430, B3L-C5(xxi)
Gross, Thilo: 75, A2L-A1(xx)
Guan, Junbiao: 595, C3L-B2(xxxvii)

H
Hada, Takahiro: 181, A3L-D2(xxxiii)
Haidar, Ali: 480, C1L-B5(xxxxiii)
Halgas, Stanislaw: 366, B2L-E3(xxix)
Hamasuna, Yukihiro: 607, C3L-C2(xxxiii)
Hara, Masahiko: 520, C1L-D5(xxxiv)
Hara, Naoki: 565, C2L-D2(xxxvi)
Hara, Naoyuki: 446, B3L-D4(xxxx), 631, C3L-D4(xxxviii)
Haraguchi, Taku: 504, C1L-D1(xxxiii)
Haraguchi, Yuta: 581, C3L-A2(xxxxvi)
Hasegawa, Mikio: 289, B2L-A2(xxvii), 378, B3L-A1(xxx), 386, B3L-A3(xxx)
Hasler, Martin: 524, P3L-A1(xxxxiv)
Hayakawa, Yoshihiro: 615, C3L-C4(xxxviii), 619, C3L-D1(xxxiii)
Hayashi, Hatsuo: 241, B1L-C5(xxv)
Hayashi, Naoki: 229, B1L-C2(xxv)
Hill, David: 398, B3L-B1(xxx)
Hirashima, Shota: 249, B1L-D2(xxv)
Hirata, Yoshito: 577, C3L-A1(xxxxvi)
Hisakado, Takashi: 257, B1L-D4(xxvi)
Ho, Daniel W. c.: 233, B1L-C3(xxv)
Hoang Le, Hiep: 277, B1L-E5(xxvi)
Hooker, Andrew: 67, A1L-D4(xix)
Horikawa, Yo: 561, C2L-D1(xxv)
Horio, Yoshihiko: 265, B1L-E2(xxvii), 301, B2L-A5(xxvii), 342, B2L-D2(xxviii), 382, B3L-A2(xxx)
Horiuchi, Kazuo: 225, B1L-C1(xxv)

I
Igarashi, Masaki: 334, B2L-C5(xxviii)
Igeta, Hideki: 378, B3L-A1(xxxiv)
Ikebe, Masayuki: 334, B2L-C5(xxviii)
Ikeguchi, Tohru: 119, A2L-D1(xxi), 123, A2L-D2(xxi), 205, B1L-A4(xxiv), 293, A2L-D2(xxi), 205, B1L-A4(xxiv), 293, B2L-A3(xxvii), 581, C3L-A2(xxxvi)
Ikuta, Chihiro: 189, A3L-D4(xxxiii)
Imai, Satoshi: 549, C2L-C2(xxxv)
Imura, Jun’ichi: 488, C1L-C2(xxxiii)
Irisawa, Masaki: 446, B3L-D4(xxxi)
Ishikawa, Mitsuru: 173, A3L-C3(xxxiii)
Ishikawa, Yusuke: 51, A1L-C5(xix)
Ito, Daisuke: 627, C3L-D3(xxxxviii)
Itoh, Naoki: 237, B1L-C4(xxv)

J
Jessa, Mieczyslaw: 418, B3L-C2(xxxxvi)
Jin’no, Kenya: 193, B1L-A1(xxiv), 265, B1L-E2(xxvii)
Jin, Weifeng: 595, C3L-B2(xxxxvii)
Jitsumatsu, Yutaka: 426, B3L-C4(xxvi)
Jones, Nick: 454, C1L-A1(xxiii)

K
Kösem, Mustafa: 354, B2L-D5(xxix)
Kai, Tatsuya: 434, B3L-D1(xxv), 438, B3L-D2(xxv)
Kamata, Hiroyuki: 406, B3L-B3(xxx)
Kamiyama, Kyoei: 406, B3L-B3(xxx)
Kanamori, Yuta: 394, B3L-A5(xxx)
Kanazawa, Takaumi: 229, B1L-C2(xxv)
Kanzawa, Yuchi: 603, C3L-C1(xxvii)
Karaes, Kristof: 149, A3L-B1(xxii)
Karacas, Theodoros: 461, C1L-A4(xxxii)
Karube, Shu: 249, B1L-D2(xxv)
Kataoka, Hiroshige: 374, B2L-E5(xxix)
Kato, Hideyuki: 119, A2L-D1(xxii)
Kato, Tomohiro: 386, B3L-A3(xxvii)
Kawahara, Masakazu: 19, A1L-B2(xxviii), 23, A1L-B3(xxvii)
Kazimierczuk, Marian: 362, B2L-E2(xxix)
Kazunaka, Rumi: 370, B2L-E4(xxix)
Kim, Song-Ju: 520, C1L-D5(xxiv)
Kinjo, Mitsunaga: 39, A1L-C2(xxvii)
Kinosita, Kazutoshi: 488, C1L-C2(xxxxii)
Kinosita, Yohsuke: 165, A3L-C1(xxvii)
Kiss, Andras: 99, A2L-B3(xx), 103, A2L-B4(xx)
Kitajima, Hiroyuki: 173, A3L-C3(xxxxii)
Kitamichi, Junji: 127, A2L-D3(xxii)
Klonowski, Wlodzimirz: 473, C1L-B3(xxxxii)
Kobori, Yoshiyuki: 245, B1L-D1(xxv)
Kohda, Tohru: 265, B1L-E2(xxvii), 426, B3L-C4(xxvi)
Kohno, Takashi: 261, B1L-E1(xxvi)
<table>
<thead>
<tr>
<th>Last Name</th>
<th>First Name</th>
<th>Page Numbers</th>
<th>Volumes and Sections</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ogura</td>
<td>Takeshi</td>
<td>27, 469</td>
<td>A1L-B4(xviii), C1L-B2(xxxii)</td>
</tr>
<tr>
<td>Ohno</td>
<td>Shuhei</td>
<td>119</td>
<td>A2L-D1(xxi)</td>
</tr>
<tr>
<td>Oishi</td>
<td>Shin'ichi</td>
<td>35, 135, 144, 469</td>
<td>A1L-C1(xviii), A3L-A1(xxii), A3L-A4(xxxii)</td>
</tr>
<tr>
<td>Ohno</td>
<td>Shuhei</td>
<td>119</td>
<td>A2L-D1(xxi)</td>
</tr>
<tr>
<td>Ono</td>
<td>Aiko</td>
<td>39, 134, 144, 469</td>
<td>A1L-C2(xxvii), A3L-B2(xxviii), A3L-D3(xxxii), 512, C1L-D3(xxxiv)</td>
</tr>
<tr>
<td>Osana</td>
<td>Yuko</td>
<td>181</td>
<td>A3L-D2(xxiii), 185, A3L-D3(xxiii), 512, C1L-D3(xxxiv)</td>
</tr>
<tr>
<td>Oseily</td>
<td>Hassan</td>
<td>480</td>
<td>C1L-B5(xxxiii)</td>
</tr>
<tr>
<td>Oswald</td>
<td>Marion</td>
<td>599</td>
<td>C3L-B3(xxxvii)</td>
</tr>
<tr>
<td>Oseily, Hassan</td>
<td></td>
<td>480</td>
<td>C1L-B5(xxxiii)</td>
</tr>
<tr>
<td>Pareschi</td>
<td>Fabio</td>
<td>63</td>
<td>A1L-D3(xix)</td>
</tr>
<tr>
<td>Pazienza</td>
<td>Giovanni E.</td>
<td>537, 599, 599</td>
<td>C2L-B1( xxxv ), C3L-B3(xxxvii)</td>
</tr>
<tr>
<td>Polychronaki, Georgia Emmanouil</td>
<td>455</td>
<td>C1L-A2(xxxii)</td>
<td></td>
</tr>
<tr>
<td>P</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Quang Vu</td>
<td>Hien</td>
<td>350</td>
<td>B2L-D4(xxix)</td>
</tr>
<tr>
<td>R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Radebach</td>
<td>Alexander</td>
<td>3</td>
<td>A1L-A2(xviii)</td>
</tr>
<tr>
<td>Rak</td>
<td>Adam</td>
<td>161</td>
<td>A3L-B4(xxii)</td>
</tr>
<tr>
<td>Ramig</td>
<td>Lorraine</td>
<td>457</td>
<td>C1L-A3(xxxii)</td>
</tr>
<tr>
<td>Rekeczky</td>
<td>Csaba</td>
<td>95</td>
<td>A2B-B2(xx)</td>
</tr>
<tr>
<td>Righero</td>
<td>Marco</td>
<td>573</td>
<td>C2L-D4(xxxvi)</td>
</tr>
<tr>
<td>Roska</td>
<td>Tamás</td>
<td>593</td>
<td>C3L-B1(xxvii)</td>
</tr>
<tr>
<td>Rosatti</td>
<td>Riccardo</td>
<td>63</td>
<td>A1L-D3(xix)</td>
</tr>
<tr>
<td>Rudolf</td>
<td>Lars</td>
<td>75, 162</td>
<td>A2L-A1(xx)</td>
</tr>
<tr>
<td>Rump</td>
<td>Siegfried</td>
<td>35</td>
<td>A1L-C1(xviii), 139, A3L-A2(xxii)</td>
</tr>
<tr>
<td>Runge</td>
<td>Jakob</td>
<td>3</td>
<td>A1L-A2(xviii)</td>
</tr>
<tr>
<td>Ruszinkó</td>
<td>Miklós</td>
<td>99</td>
<td>A2L-B3(xx)</td>
</tr>
<tr>
<td>S</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sadjadi Alamdari, Seyed Amin</td>
<td>450</td>
<td>B3L-D5(xxxi)</td>
<td></td>
</tr>
<tr>
<td>Sagara</td>
<td>Taizo</td>
<td>422</td>
<td>B3L-C3(xxxi)</td>
</tr>
<tr>
<td>Sahab</td>
<td>Ali Reza</td>
<td>450</td>
<td>B3L-D5(xxxi)</td>
</tr>
<tr>
<td>Saito</td>
<td>Toshimichi</td>
<td>197</td>
<td>A1L-A2(xxiv), 201, B1L-A3(xxiv), 314, B2L-B4(xxvii), 484, C1L-C1(xxxiii), 549, C2L-C2(xxxv), 611, C3L-C3(xxxvii)</td>
</tr>
<tr>
<td>Sakamoto</td>
<td>Yusuke</td>
<td>382</td>
<td>B3L-A2( xxx )</td>
</tr>
<tr>
<td>Samura</td>
<td>Toshikazu</td>
<td>241</td>
<td>B1L-C5(xxv)</td>
</tr>
<tr>
<td>Sanjuán</td>
<td>Miguel A. f.</td>
<td>533, 533</td>
<td>C2L-A3(xxxiv)</td>
</tr>
<tr>
<td>Sasaki</td>
<td>Tomoyuki</td>
<td>390</td>
<td>B3L-A4( xxx )</td>
</tr>
<tr>
<td>Šato</td>
<td>Masatoshi</td>
<td>131</td>
<td>A2L-D4(xxii), 245, B1L-D1(xxv)</td>
</tr>
<tr>
<td>Šato</td>
<td>Shigeo</td>
<td>39, 615</td>
<td>A1L-C2(xviii), C3L-C4(xxxvii), C3L-D1(xxxvii)</td>
</tr>
<tr>
<td>Schulze-Bonhage, Andreas</td>
<td>455, 533</td>
<td>C1L-A2( xxxii )</td>
<td></td>
</tr>
<tr>
<td>Sekikawa</td>
<td>Munehisa</td>
<td>342</td>
<td>B2L-D2(xxviii)</td>
</tr>
<tr>
<td>Sekiya</td>
<td>Hiroo</td>
<td>51</td>
<td>A1L-C5(xix), 362, B2L-E2(xxix), 516, C1L-D4(xxxiv)</td>
</tr>
<tr>
<td>Sendiña-Nadal, Irene</td>
<td>7</td>
<td>A1L-A3(xviii)</td>
<td></td>
</tr>
<tr>
<td>Sengör</td>
<td>Neslihan Serap</td>
<td>354, 533</td>
<td>B2L-D5(xxix)</td>
</tr>
<tr>
<td>Senthilkumar, Dharmapuri Vijayan</td>
<td>83, 83</td>
<td>A2L-A3(xx)</td>
<td></td>
</tr>
<tr>
<td>Šetti</td>
<td>Gianluca</td>
<td>63</td>
<td>A1L-D3(xix)</td>
</tr>
<tr>
<td>Shibata</td>
<td>Kazuaki</td>
<td>301</td>
<td>B2L-A5(xxvii)</td>
</tr>
<tr>
<td>Shimada</td>
<td>Yutaka</td>
<td>581</td>
<td>C3L-A2(xxxvi)</td>
</tr>
<tr>
<td>Shimaciu</td>
<td>Yuki</td>
<td>269</td>
<td>B1L-E3(xxvi)</td>
</tr>
<tr>
<td>Shimizu</td>
<td>Kuniyasu</td>
<td>406</td>
<td>B3L-B3( xxx ), 623, C3L-D2(xxxxvii)</td>
</tr>
<tr>
<td>Shimomura, Sho</td>
<td>504, 533</td>
<td>C1L-D1(xxxiii)</td>
<td></td>
</tr>
<tr>
<td>Shindo</td>
<td>Takuya</td>
<td>193</td>
<td>B1L-A1( xxiv )</td>
</tr>
<tr>
<td>Shintani</td>
<td>Takeshi</td>
<td>434</td>
<td>B3L-D1( xxxi )</td>
</tr>
<tr>
<td>Slot</td>
<td>Krzysztof</td>
<td>476</td>
<td>C1L-B4(xxxii)</td>
</tr>
<tr>
<td>Small</td>
<td>Michael</td>
<td>11</td>
<td>A1L-A4(xviii), 589, C3L-A4(xxxvi)</td>
</tr>
<tr>
<td>Soós</td>
<td>Gergely Balázs</td>
<td>161, 161</td>
<td>A3L-B4(xxii)</td>
</tr>
<tr>
<td>Sobanska</td>
<td>Paulina</td>
<td>59</td>
<td>A1L-D2(xix)</td>
</tr>
<tr>
<td>Sota</td>
<td>Takahiro</td>
<td>615</td>
<td>C3L-C4(xxxvii), 619, C3L-D1(xxxvii)</td>
</tr>
<tr>
<td>Spaanenburg, Lambert</td>
<td>91, 91</td>
<td>A2L-B1(xx)</td>
<td></td>
</tr>
<tr>
<td>Stepan</td>
<td>Holger</td>
<td>585</td>
<td>C3L-A3(xxxvi)</td>
</tr>
<tr>
<td>Stepien</td>
<td>Robert</td>
<td>473</td>
<td>C1L-B3(xxxii)</td>
</tr>
<tr>
<td>Šteifs</td>
<td>Dirk</td>
<td>75</td>
<td>A2L-A1(xx)</td>
</tr>
<tr>
<td>Su</td>
<td>Yuanqi</td>
<td>213</td>
<td>B1L-B2(xxiv)</td>
</tr>
<tr>
<td>Suetsugu</td>
<td>Tadashi</td>
<td>51</td>
<td>A1L-C5(xix)</td>
</tr>
<tr>
<td>Šugimoto</td>
<td>Masaki</td>
<td>504</td>
<td>C1L-D1(xxxiii)</td>
</tr>
<tr>
<td>Suzuki</td>
<td>Takayuki</td>
<td>205</td>
<td>B1L-A4(xxiv)</td>
</tr>
<tr>
<td>Szabo</td>
<td>Vilmos</td>
<td>95</td>
<td>A2L-B2(xx)</td>
</tr>
<tr>
<td>Szolgya</td>
<td>Péter</td>
<td>103</td>
<td>A2L-B4(xx)</td>
</tr>
</tbody>
</table>

| T        |            |              |                     |
| Tachibana, Toshihiro | 297, 297 | B2L-A4(xxvii) |         |
| Tachikawa, Ryota     | 512, 512 | C1L-D3(xxxiv) |         |
| Tadeusiewicz, Michal | 366, 366 | B2L-E3(xxix) |         |
| Taguchi, Yu          | 394, 394 | B3L-A5( xxx ) |         |
| Takahashi, Koichi    | 209, 209 | B1L-B1(xxiv) |         |
| Takahashi, Norikazu  | 358, 358 | B2L-E1(xxix) |         |
Nonlinear Dynamics of Biological Rhythms

Albert Goldbeter

Faculté des Sciences, Université Libre de Bruxelles
Campus Plaine, CP 231, B-1050 Brussels, Belgium

Abstract—Oscillations arise in genetic and metabolic networks as a result of various modes of cellular regulation. In view of the large number of variables involved and of the complexity of feedback processes that generate oscillations, mathematical models and numerical simulations are needed to fully grasp the nonlinear dynamics of biological rhythms. Models are also necessary to comprehend the transition from simple to complex oscillatory behaviour. To illustrate how mathematical modeling contributes to clarify the dynamical bases of biological oscillations, the presentation will focus on circadian rhythms. These autonomous rhythms, with a period close to 24h, are conspicuous by their ubiquity and by the key role they play in allowing organisms to adapt to their periodically changing environment. Mathematical models closely related to experimental observations will be considered for circadian clocks. Models of increasing complexity predict the occurrence of sustained circadian oscillations corresponding to the evolution toward a limit cycle. Chaos can occur in the model either in autonomous conditions, or as a result of periodic forcing by light-dark cycles. Stochastic simulations show how circadian oscillations are affected by molecular noise. Extending the model to circadian rhythms in mammals permits an investigation of the dynamical bases of physiological disorders of the sleep-wake cycle in humans. Finally, the analysis of a model for the mammalian cell cycle shows how the latter can be entrained by the circadian clock.

References

Dynamics on Complex Networks with Time Varying Topology

Juergen Kurths, Jonathan Donges, Norbert Marwan and Yong Zou

Potsdam Institute for Climate Impact Research and Humboldt University Berlin, Institute of Physics, Germany
Email: Juergen.kurths@pik-potsdam.de

Abstract—Recent research has revealed a rich and complicated network topology in various model systems as well as in several fields of applications. It will be discussed whether this approach can lead to useful new insights into rather large complex systems or whether it is fashionable only to interpret various phenomena from this viewpoint and publish papers on that. On one side, among such studies it has become very popular to look for a scale-free behaviour by showing log-log plots. This reminds the hunting for low dimensional chaos in the 80ies of the last millennium. On the other side, many promising approaches have already lead to useful applications, e.g. immunization problems (spreading of diseases), functioning of biological/physiological processes as protein networks, brain dynamics, colonies of thermites, or functioning of social networks as network of vehicle traffic in a region or air traffic. A challenging task is to understand the implications of such network structures on the functional organization of the brain activities. This is studied here basing on dynamical complex networks. We investigate synchronization dynamics on the cortico-cortical network of the cat by modelling each node (cortical area) of the network with a sub-network of interacting excitable neurons. We find that the network displays clustered synchronization behaviour and the dynamical clusters coincide with the topological community structures observed in the anatomical network. Our results provide insights into the relationship between the global organization and the functional specialization of the brain cortex. This approach of a network of networks seems to be of general importance, especially for spreading of diseases or opinion formation in human societies or socio-economic dynamics. Therefore, we next study a network of networks with time varying topology for modelling epidemic spreading. We find qualitatively different behaviour there in dependence on the changes of the topology.

References

Evolving Climate Networks

Norbert Marwan¹, Jonathan F. Donges¹², Alexander Radebach¹², Jakob Runge¹², and Jürgen Kurths¹²

1 Potsdam Institute for Climate Impact Research, P.O. Box 60 12 03, D-14412 Potsdam, Germany
2 Department of Physics, Humboldt-Universität zu Berlin, Robert-Koch-Platz 4, 10115 Berlin, Germany

Email: marwan@pik-potsdam.de

Abstract—We propose a method to reconstruct and analyse an evolving complex network from data generated by a spatio-temporal dynamical system. We study reanalysis surface air temperature data by different complex network measures. This approach reveals a rich internal structure in complex climate networks and allows to study the stability of the climate network and the impacts of teleconnections (e.g., El Niño/ Southern Oscillation). Moreover, the betweenness analysis uncovers peculiar wave-like structures of high information flow, that can be related to global surface ocean currents.

1. Introduction

Climate dynamics is related to spatio-temporal variability on different scales. Various approaches for spatio-temporal analysis are in use for a better understanding of the climate variability, like wavelet or EOF analysis [6, 8]. More recently, the complex network approach has been suggested for a spatio-temporal interaction analysis of climate data [9, 17, 20]. The complex network paradigm has proven to be a fruitful tool for the investigation of complex systems in various areas of science, e.g., the internet and world wide web in computer science, food webs, gene expression and neural networks in biology, and citation networks in social science [5, 16, 19]. The intricate interplay between the structure and dynamics of real networks has received considerable attention [5]. Particularly, synchronisation arising by the transfer of dynamical information in complex network topologies has been studied intensively [3]. The application of complex network theory to climate science is a young field, where only few studies have been reported recently [9, 10, 17, 20]. The vertices of a climate network are identified with the spatial grid points of an underlying global climate data set. Edges are added between pairs of vertices depending on the degree of statistical interdependence between the corresponding pairs of anomaly time series taken from the climate data set. Climate networks enable novel insights into the topology and dynamics of the climate system over many spatial scales ranging from local properties as the number of first neighbours of a vertex (the degree centrality) to global network measures such as the clustering coefficient or the average path length. For example, the betweenness centrality uncovers peculiar wave-like structures of high energy flow, that can be related to global surface ocean currents [9]. These insights are conceptually new and cannot be obtained using classical methods of climatology such as principal component analysis (PCA) or singular spectrum analysis (SSA) of anomaly fields [12, 18], because these are by design local in a network sense and are not suitable to study local flow measures depending on a global network topology.

Because the climate is changing in time, it is obvious, that the climate network should also change in time, i.e., the correlation structure of climatological fields cannot generally be considered to be stationary in a statistical sense. This is related to the concept of evolving complex networks (dynamically changing networks), which have received increasing interest in the last years, as real systems often exhibit variations in the ensembles of elements (vertices) and interrelations (edges) [2, 4, 13]. Evolving networks are marked by the emergence of information, rich dynamics and structure formation, e.g., collective behaviour between some of the elements. They can switch between stability and instability, leading to new qualitative behaviour like robustness or vulnerability. Evolving complex networks have been successfully studied to investigate failure propagation in power-grids [1] or hierarchical structures in the brain [16, 21].

2. Data and data pre-processing

We utilise the 6-hourly global surface air temperature (SAT) field to construct climate networks, that allows to directly capture the complex dynamics on the interface between ocean and atmosphere due to heat exchange and other local processes. SAT, therefore, enables us to study atmospheric as well as oceanic dynamics using the same climate network. We use reanalysis data provided by the National Center for Environmental Prediction/National Center for Atmospheric Research (NCEP/NCAR) [15]. A data set consists of a regular spatio-temporal grid with time series $x_i(t)$ associated to every spatial grid point $i$ at latitude $\lambda_i$ and longitude $\phi_i$. The data starts at January 1948 and ends at December 2009 (744 months). The latitudinal and longitudinal resolution is $\Delta \lambda = \Delta \phi = 2.5\, (N = 10, 224$ vertices).

To minimise the bias introduced by solar forcing common to all time series in the data set, we calculate anomaly time series $a_i(t)$ from the $x_i(t)$, i.e., remove the mean annual cycle by phase averaging. Furthermore, the anomaly time series are normalised to zero mean and unit variance,
and the data field is transformed from the cartesian grid to an icosahedral grid (in order to avoid the bias due to the higher grid point density at the poles).

3. Methodology

The climate interaction network is a representation of the interactions or interrelations \( I_{ij} \) between the time series at the grid points \( i \) and \( j \). Such interactions can be measured, e.g., by cross correlation \( I_{ij} = C_{ij} \) (as done by Yamasaki et al. [20]) or mutual information (MI) \( I_{ij} = M_{ij} \) [9]. MI is a measure from information theory, that can be interpreted as the excess amount of information generated by falsely assuming the two time series \( a_i \) and \( a_j \) to be independent, and is able to detect linear as well as nonlinear relationships [14]. MI can be estimated using

\[
M_{ij} = \sum_{\mu \nu} p_{ij}(\mu, \nu) \log \frac{p_{ij}(\mu, \nu)}{p_i(\mu)p_j(\nu)},
\]

where \( p_{ij}(\mu) \) is the probability density function (PDF) of the time series \( a_i \), and \( p_{ij}(\mu, \nu) \) is the joint PDF of a pair \((a_i, a_j)\).

By definition, \( I_{ij} \) is symmetric, so that \( I_{ij} = I_{ij} \). We can also evaluate time delayed correlation and MI. This is appropriate when studying climate networks on smaller time scales using data sets with (sub-)diurnal resolution [20].

We now construct the climate interaction network by thresholding the interrelation matrix \( I_{ij} \), i.e., only pairs of vertices \((i, j)\) that satisfy \( I_{ij} > \tau \) are regarded as linked, where \( \tau \) is the threshold. Using the Heaviside function \( \Theta(x) \), the adjacency matrix \( A_{ij} \) of the climate network is given by \( A_{ij} = \Theta(1 - \tau) - \delta_{ij} \), where \( \delta_{ij} \) is the Kronecker delta (subtracted in order to remove self-loops). If we consider time delayed interrelations, we choose the maximum value of interrelations within the range of considered time delays and apply the threshold on this maximal value. The resulting climate network is an undirected and unweighted simple graph (\( A_{ij} \) inherits its symmetry from \( I_{ij} \)). The threshold \( \tau \) can be chosen as a fixed value or on dependence of a desired edge density. As the network characteristics (e.g., betweenness centrality) depend on the choice of the threshold \( \tau \), it is sometimes desirable to constrain the edge density \( \rho = 2E/N(N-1) \), where \( E \) gives the total number of edges [9], and to use a corresponding threshold \( \tau = \tau(\rho) \). It was shown recently, that the backbone of the climate network is most clearly observed at small \( \rho \) with corresponding large threshold \( \tau \), that is very unlikely to be exceeded by chance, and that was reassured using various significance tests [9].

Following the idea of evolving networks we divide the reanalysis data into short time epochs and construct the climate networks from these epochs. This way we get a time evolving complex network \( A_{ij}(t) \) and can analyse the time variation of the topology of the interaction patterns and heat transfer in the global climate system.

The climate interaction network can be characterised by global and local network statistics. Several of them have been analysed for climate networks [9, 17, 20].

In order to study the stability of a climate network, Yamasaki et al. have suggested to study the number of robust edges [20]. A robust edge is defined as an edge which remains in the evolving network for some time \( k \). Formally, we calculate a matrix

\[
R_{ij}(t) = \prod_{r=t-k}^{t} A_{ij}(t'),
\]

containing only such edges lasting a period of at least \( k \) time steps. Here we consider a period of \( k = 250 \) days. The number of robust edges \( n \) is then calculated by the sum over this matrix \( n(t) = \sum_{ij} R_{ij}(t) \).

Donges et al. have studied the betweenness centrality (BC) of a climate interaction network [9]. BC includes global topological information by relying on shortest paths between pairs of vertices (communication through the network concentrates on shortest paths). There are \( \sigma_{ij} \) shortest paths connecting \( i \) and \( j \). Vertex \( v \) is an important mediator for communication in the network, if it is traversed by a large number of all existing shortest paths. Mathematically, the betweenness \( BC_i \) can be expressed by

\[
BC_i = \sum_{i,j \neq v} \frac{\sigma_{ij}(v)}{\sigma_{ij}},
\]

where \( \sigma_{ij}(v) \) gives the number of shortest paths from \( i \) to \( j \), that include \( v \) [11], and is normalised by \( \sigma_{ij} \). Because the shortest paths contain only edges corresponding to pairs of highly dynamically interrelated time series, BC is a local measure of dynamical information flow. Since we use it to analyse a temperature field we interpret BC more fundamentally as a measure of the flow of energy (heat).

4. Results

The number of robust edges \( n \) in the climate network has been calculated from a network based on cross correlation and a threshold \( \tau = 2\sigma_{\text{E},(i)} \) (where \( \sigma_{\text{E},(i)} \) is the standard deviation of the ensemble of cross correlations for each vertex \( i \)). The number of robust edges decreases significantly after the onset of an El Niño event (grey arrows).
deviation of the cross correlation function between \( i \) and \( j \)
normalised by its largest value \([20]\). \( n \) varies significantly
for the studied period (Fig. 1). Moreover, we find that the
sudden decrease of \( n \) is strongly related to the onset of El
Niño events. After an El Niño event, the global climate
regime needs several years to recover to the former number
of robust edges.

For the calculation of \( BC_v \), we have used MI and fixed
the edge density at \( \rho = 0.001 \). The time evolving calculation
of \( BC_v \) reveals a strong temporal variation also of
this measure (Figs. 2–4). Furthermore, the spatial vari-
ation of \( BC_v \) forms characteristic patterns and obviously un-
veils paths of strong interactions and interrelations, which
we can interpret as important transport paths (of energy or
heat) within the climate network. For example, the year
1993 after the eruption of the Pinatubo volcano in 1991 is
remarkable (Fig. 3). Paths of higher betweenness start at
the site of the volcano and spread out over this part of the
Earth. Comparing a typical El Niño with a La Niña years
reveals structural differences in the interconnectivity, i.e.,
in the teleconnection patterns, in the climate system, and,
thus, highlights the regional and global impact of the El
Niño/Southern Oscillation (Figs. 2, 4).

In analogy with the internet, we call the network of
these channels of high energy flow the backbone of the
climate network. Several of the backbone features which
lie over the oceans coincide with well known ocean cur-
cents \([9]\). Temperature anomalies in sea surface tempera-
ture (SST) are advected by the surface ocean currents and
transferred to the SAT field via heat flux coupling. There-
fore, ocean currents provide a physical mechanism for the
transport of energy on localised linear structures over large
distances. In \([9]\) it was shown that the betweenness field is
neither correlated to SAT-SST gradients, nor is it statisti-
cally strongly related to the fields of degree and closeness
centrality. Therefore we can underline that the backbone
structures observed in climate networks are neither a trivial
response to local anomalies in the SST-SAT gradient nor ar-
tifacts of chains of hubs with high degree and/or closeness
centrality (highly spatially interrelated regions).

5. Conclusions

We have demonstrated the application of the evolving
complex networks approach for a spatio-temporal analysis
of global climate field data. Applying this approach we
have been able to analyse the global stability of the climate
regime and to unveil pathways of strong interactions and
interrelations (teleconnections). By studying the number
of robust edges we have found that the El Niño/Southern
Oscillation strongly reduces the dynamical interconnectiv-
ity within the global climate regime after the onset of El
Niño events. The betweenness centrality underlines such
changes by depicting pathways of heat exchange. For ex-
ample, just after the eruption of the volcano Pinatubo, these
pathways have been centred around the volcano underlying
that the eruption had a major impact on the global air sur-
face temperature field.
It is important to realise that our complex network approach is an essential ingredient in the discovery of the climate backbone. For example, the main advantage of the new approach is that it takes into account the global network topology of pairwise interrelationships between regions. However, the classical linear methods (PCA, SSA, etc. [12, 18]) readily applied to disclose teleconnection patterns in climatology use information about next neighbours at each grid point, and therefore, only local from a complex network point of view. Our method is promising to study the impact of extreme events such as strong El Niños, extreme Monsoons or volcanic eruptions on the topology of climate interaction networks. In the future it will thereby allow us to obtain new insights into the individual local signature of changes in the energy and information flow structure and stability of the climate system.

Acknowledgments

This work has been financially supported by the German Research Foundation (SFB 555 project C4), the Federal Ministry for Education and Research BMBF (Potsdam Research Cluster for Georisk Analysis, Environmental Change and Sustainability, PROGRESS), and the Leibniz Society WGL (project ECONS). All calculations have been performed on the IBM iDataPlex supercomputer at the Potsdam Institute for Climate Impact Research and by using the software package igraph [7].

References

Synchronization regulation of a complex network
by link rewiring or node pinning

I. Sendiña-Nadal†, J.A. Almendral†, I. Leyva†, J.M. Buldú†, D. Yu‡, and S. Boccaletti§

†Complex Systems Group, Dept. of Signal Theory and Communications,
Universidad Rey Juan Carlos, Camino del Molino s/n, 28943 Fuenlabrada, Madrid, Spain
‡School of Automation Engineering, University of Electronic Science and Technology, Chengdu 610054, China
§Embassy of Italy in Tel Aviv, 25 Hammered Street, 68125 Tel Aviv, Israel, and
CNR-Istituto dei Sistemi Complessi, Via Madonna del Piano 10, Sesto Fiorentino, 50019 Firenze, Italy
Email: irene.sendina@urjc.es

Abstract—Synchronization is essential for the correct functioning of many technological systems as in wireless communications, parallel/distributed computing, or electrical power distribution lines. One of the properties that those systems should have is scalability, i.e., the ability to be augmented without loosing the synchronization performance. In this communication, we describe a method to properly engineer the wiring of a network of dynamical units in order to achieve synchronization. We provide rules to guide the rewiring of the links of a given node in the network or to guide the pinning with a new only node added to the network. In particular, we will focus on sequential regulation processes in order to establish conditions for identifying the minimal number of connections needed for regulating synchrony, as well as a practical way to find the corresponding sequence of connections.

1. Introduction

Regulation of synchronization of networking dynamical units is an issue of the outmost importance because synchronous behavior is a must for the correct functioning of many technological and biological networks [1]. As concerning the former class of networks, synchronous message passing in computer science, is a form of communication used in parallel/distributed computing [2] or in wireless sensor networks [3], while in the later class, a quorum-sensing mechanism is used in many populations of cells to cause expression of genes in gene regulatory networks [4, 5].

A common feature of these systems is that they are continuously growing. Thus, it is essential to understand how this growth it is accompanied without loosing the synchronization performance, in order to optimally design technological networks or synthetic biological gene networks.

Here we propose pinning regulation as a model to understand the regulatory mechanisms underlying this synchronous behavior of a network of dynamical units [6]. We provide a full description on how to engineer an external pinning action on a network of identical dynamical units leaving unchanged its dynamical properties and topology and minimally acting to achieve regulation of synchrony. The pinning interaction comes from an external node which is identical to the nodes of the network and whose only effect is establishing bidirectional connections of the same strength in sequential steps. Our aim is to provide the conditions to achieve this with the minimal number of connections.

2. Pinning regulation

In order to evaluate the pinning regulation of a given network, we consider an initial graph \( G_0 \) of \( N \) bidirectional coupled identical systems each one represented by a \( m \)-dimensional real vector state \( x_i \) \((i = 1, \ldots, N)\), whose evolution is given by the equation:

\[
\dot{x}_i = f(x_i) + \sigma \sum_{j=1}^{N} L_{ij} h(x_j),
\]

and depends on the local function \( f \), on the coupling function \( h \), on the Laplacian matrix \( \mathcal{L} \in \mathcal{M}_N \) associated to the connectivity described by the graph \( G_0 \), and on the fixed coupling strength \( \sigma \). The assumption of a network made of identical systems and a zero-row sum Laplacian ensures the existence of a synchronous state \([x_1(t) = x_2(t) = \ldots = x_N(t) = x_s(t)]\) whose stability can be studied by means of the Master Stability Function (MSF) approach [7]. The MSF approach demonstrates that there are only two classes of systems for a given local and coupling functions that allow stability of the synchronous state [8].

Figure 1 shows the shapes of the maximum Lyapunov exponent \( \Lambda \) for the so called class II, Fig. 1 (central column), and class III systems, Fig. 1 (right column), as a function of a parameter \( \lambda \) which essentially depends on \( f \) and \( h \) [8]. For class II systems, the synchronous state \( x_s \) is stable above a critical \( \lambda^1 = \frac{\sigma}{\mu} \) as \( \Lambda \) is a monotonically decreasing function, while stability for class III with a V-shape \( \Lambda \) function is reached within an interval of values \( \lambda^1 = \frac{\sigma}{\mu} \) and \( \lambda^2 = \frac{\sigma}{\mu} \). In our case, as the coupling strength \( \sigma \) is fixed, the variable moving along the \( x \)-axis is directly related to the eigenvalues of the Laplacian matrix.

We start from a situation in which the network topology gives rise to an unstable synchronous state (as depicted at
the top row of Fig. 1), that is, if we order the eigenvalues (which is possible because the Laplacian matrix is zero-row sum and symmetric), $0 = \lambda_1^0 < \lambda_2^0 < \ldots < \lambda_N^0$, the smallest non zero eigenvalue, $\lambda_1^0$, is outside the stability region for both classes of systems. For the class III system besides, the largest eigenvalue, $\lambda_N^0$, has to be initially located inside the stability region in order to be able to regulate the network to synchrony. Otherwise, if $\lambda_1^0 > \frac{\mu_1}{\sigma}$, the synchronous state is impossible to turn stable by pinning.

In order to regulate the stability of $x_i(t)$, we consider here an interaction between $G_0$ and an external dynamical system, identical to those in $G_0$, that forms, at successive times $t_n\,(n = 1, \ldots, N)$ a series of $\sigma$-strength connections by pinning the nodes in $G_0$ with a given sequence $\{s_1, s_2, \ldots, s_N\}$. This is schematically shown in Fig. 1 at the left column for $t_1$ (middle row) and $t_2$ (bottom row), where the original graph is being pinned by an external node (red links).

This situation is now described by this new equation of motion,

$$x_i = f(x_i) + \sigma \sum_{j=1}^{N+1} L'_{ij}(t) h(x_j),$$

where $L'(t) = \{L'_{ij}(t)\} \in M_{N+1}$ is now the following time dependent Laplacian matrix

$$L'_{11}(t) \quad L'_{12} \quad \ldots \quad L'_{1N} \quad \Theta(t-T_1)$$

$$L'_{21} \quad L'_{22}(t) \quad \ldots \quad L'_{2N} \quad \Theta(t-T_2)$$

$$\vdots \quad \vdots \quad \ddots \quad \vdots \quad \vdots$$

$$\Theta(t-T_N) \quad \ldots \quad \Theta(t-T_N) \quad L'_{N+1,N+1}(t)$$

whose elements are such that:

i) $L'_{ij} = L_{ij}$ for $i \neq j$ and $i, j < N+1$;

ii) $\Theta(t-T_i) = L'_{i,i+N}(t) = L'_{i,N+1}(t)$, being $T_i$ the time at which the $i$th node in $G_0$ is pinned by the interaction with the external node, and $\Theta$ the Heaviside function. Notice that while the index $i$ in $t_i$ refers to a time ordering, the index $i$ in $T_i$ points to the ordering of the pinning sequence, and therefore $t_i = T_n$.

iii) $L'_{ii}(t) = -\sum_{j \neq i} L'_{ij}$.

The effect of pinning the nodes in $G_0$ with the external one is to produce a new set of eigenvalues for the Laplacian matrix $L'$. $0 = \lambda_1^{0'} < \lambda_2^{0'} < \ldots < \lambda_N^{0'} < \lambda_{N+1}^{0'}$. It is clear that the only way to regulate the situation depicted at the top row of Fig. 1 by pinning an external dynamical system is moving $\lambda_2$ inside the stability region for both class II and class III systems and by keeping the largest eigenvalue (now $\lambda_{N+1}$) inside for class III.

While for class II systems, a practical way to regulate the synchronous state is by finding the optimal pinning sequence maximizing $\lambda_2$, for the class III the synchronization stability is ensured for all coupling architectures whose corresponding eigenvalue spectrum is entirely contained within the stability region of the MSF, delimited by the two threshold parameters $\frac{\mu_2}{\sigma}$ and $\frac{\mu_1}{\sigma}$. Then, the way we select the pinning sequence is that to maximize (at each time $t_n$ a new link is formed with the regulating node) the distance $|\lambda_2^{0'} - \lambda_{N+1}^{0'}|$ in order to ensure that we are moving to the right in the spectrum and, at the same time, to minimize $\lambda_{N+1}^{0'} - \lambda_{N+1}^{0'}$ in order to keep the largest eigenvalue far from the second threshold. This is equivalent to maximize the quantity,

$$R_n = \frac{\lambda_2^{0'} - \lambda_{N+1}^{0'}}{\lambda_{N+1}^{0'} - \lambda_N^{0'}}.$$  

Note that, the largest eigenvalue $\lambda_{N+1}$ always increases by the fact we are adding connections, while is not always true for $\lambda_2$.

3. Results

From here on, we will accompany our analytical study using the MSF with numerical examples, corresponding to the case of $N = 400$ nodes arranged in two different network configurations: a small-world network (SW) obtained as in Ref. [9] by initially arranging the $N$ nodes in a ring with connections only between nearest neighbors, and by randomly adding with probability $p = 0.02$.
a connection between unconnected pairs of nodes (i.e. obtaining an average degree \(\langle k \rangle = 2 + pN = 10\), and a scale-free network (SF) obtained by the preferential attachment process of Ref. [10] with the same average degree of \(\langle k \rangle = 10\). Furthermore, in all cases, we will consider \(f(x \equiv (x, y, z)) = [-y - z, z + 0.165y, 0.2 + z(x - 10)]\) in Eq. (1) (i.e. we will refer to the case of networks of coupled Rössler systems [11]), because it is known that such a case allows for a direct comparison of class III networks (when \(h(x \equiv (x, y, z)) = [x, 0, 0]\) with \(\mu_1 = 0.206\) and \(\mu_2 = 5.519\)) and class II networks (when \(h(x \equiv (x, y, z)) = [0, y, 0]\) with \(\mu_1 = 0.178\)).

Let us proceed first with the regulation of the class II system. The behavior of \(\lambda_2\) by pinning both the SW and the SF networks according to the criterion of maximizing \(\lambda_2\) is reported in Fig. 2. This figure allows us to identify a minimum number of links between the regulating node and the rest of nodes. In particular, it is sufficient to pin less than 2% of nodes to make the synchronous state stable. At the same time, we observe that, comparing the two complete different networks regarding their degree distribution but with the same average degree, it is evident that the SW is easier to regulate than the SF.

Figure 3 shows the behavior of the smallest (blue open circles) and largest (red open squares) eigenvalues by maximizing the quantity defined by Eq. (2) as a new link is formed between the regulating node and \(G_0\) during the pinning regulation. Note that, before the pinning starts, the initial smallest eigenvalues (blue full face circles) are below the threshold, so the synchronous state is unstable, and the largest ones (red full face squares), are inside the stability region in order to be able to regulate the system. The main difference with respect to the class II system is that the regulation is only possible up to a maximum length sequence determined by the monotonous increase of the largest eigenvalue. Another difference between regulating networks with different heterogeneities in the degree distribution is that the SF allows for a larger number of regulating sequences.

In order to verify the analytical results we performed numerical simulations with a network of coupled chaotic Rössler dynamical units and monitored the time average synchronization error \(\langle e \rangle\) with respect to the trajectory of the regulating node. Figure 3(c-d) shows for the SW and SF networks how accurately the numerical results reproduce the vanishing of \(\langle e \rangle\) within the range predicted by the theory in Fig. 3(a-b).

To conclude, let us introduce a short discussion regarding our approach of regulation of synchrony. In previous approaches of pinning control of networks [13] (i.e. the situation in which the external node is unidirectionally forcing the dynamics of the original graph), it was argued that the controllability of a generic network behavior towards an assigned synchronous evolution could be enhanced by pinning configurations that imply a decrease of the ratio \(\frac{\lambda_{\text{ext}}}{\lambda_{\text{reg}}}\) associated to the extended network topology [14]. This is so because the MSF states that the more packed are the
Figure 4: Log-linear behavior of the ratio $\frac{\lambda^N_{n+1}}{\lambda^2_n}$, where $\lambda^N_n$ and $\lambda^0_{n+1}$ are those obtained from the regulation process of the SW network of the class III system reported in Fig. 3(a). The sequences within the red rectangle correspond to all those sequences that stabilize the synchronous state, while those within the shaded region correspond to those that, despite corresponding to values of the eigenratio larger than the initial one (horizontal dashed line) and therefore making less compact the spectra, still correspond to sequences stabilizing the synchronous state.

We point out that our conclusions are not limited to processes where regulation is attained by interaction with an external node, but they can also be applied in all cases in which the problem is to regulate synchrony by rewiring the connections of given nodes of a graph. One can, indeed, imagine to start with $N$ networking systems, remove all $\tilde{k}$ connections of a given node, and substitute them with new $\tilde{k}$ connections to the other $N - 1$ units of the graph, following the ranking sequence that our criteria are providing, thus enhancing the synchronization behavior of the original graph while maintaining the same number of nodes and links.

**Acknowledgments**

Work partly supported by EU Contract No. 043309 GABA, by the Spanish Ministry of S&T under Project n. FIS2009-07072, the URJC-CM Project n. 2008-CET-3575, the Community of Madrid under the R&D Program of activities MODELICO-CM/S2009ESP-1691, and by the Chinese NSFC under Grant No. 10602026.

**References**


Why scale-free networks are a good thing for controlling disease transmission

Michael Small

Department of Electronic and Information Engineering
Hong Kong Polytechnic University Hung Hom, Kowloon, Hong Kong
Email: ensmall@polyu.edu.hk

Abstract—The observation of a scale free degree distribution in real disease incidence data tends to suggest that such diseases would be difficult, or in fact impossible to control. Moreover, it has been shown that the fat tail degree distribution for scale free networks implies that (SIS type) disease transmission cannot be eradicated for any nonzero level of infectivity. Nonetheless, we have found that when one considers a voluntary immunisation strategy, or even disease transmission in multiple waves, the presence of hub nodes actually becomes an advantage. Highly connected nodes will be immunised earlier (or conversely, quickly encounter an early and less virulent strain of infection). As a consequence of the frailty of scale-free networks, the removal of hub nodes will actually reduce the infectivity of the disease. In this paper we consider disease transmission on scale free networks and on more stratified networks, motivated by structures observed in society. We find that the level of infection for scale free networks is actually only moderately higher than for equivalent structured networks. Nonetheless, in finite size networks extinction occurs at a higher threshold for structured (non-scale free) networks. Conversely, these structured networks exhibit broader (in time) peaks in the disease outbreak.

1. Introduction

Since the recent rediscovery of small world and scale free networks by Watts and Strogatz [7] and Barabasi and Albert [1] transmission of infectious agents on such networks have been one of the doctrinaire examples. Indeed, it is natural to consider that for diseases for which personal contact is required to support transmission (as opposed to air borne pathogens) that the degree of connectivity between individuals will have an effect on the transmission of the disease.

The importance of this idea was given weight by the work of Boguña and colleagues who showed that disease transmission on a scale free network will persist for any non-zero value of transmission rate [2]. This is in direct contrast with our experience of homogeneous mixing (the standard differential equation based model of disease transmission) for which there always exists a finite non-zero threshold below which an infection will decrease to zero, and above which it will remain endemic.

Nonetheless, it is important to examine the result of Boguña et al. a little more closely. Essentially, the result considers the case of SIS disease transmission on a perfect scale-free network. That is, there are two disease states, susceptible (S) and infected (I). Susceptible individuals become infected with some probability λ if they have an infected neighbour. Without loss of generality, infected individuals recover and become susceptible with rate 1. If one assumes full connectivity then we have the case of disease transmission on a homogeneous system and there exist a critical rate λc > 0 such that if λ < λc the disease will be eradicated. The remarkable result of Boguña et al. is that if one examines disease transmission on a scale free network (i.e. less than full connectivity) then the exists no such λc > 0. However, the fundamental requirement for this to occur is that one must consider a infinite perfect scale free network (in [2], the authors also only consider the case where the degree exponent is between 2 and 3 — but this restriction is not entirely necessary [5]). That is, the network is infinitely large and consists of nodes such that the probability P of a node having degree k is given by

\[ P(k) \approx Ck^{-\gamma} \]  \hspace{1cm} (1)

for 2 < γ ≤ 3.

Boguña and colleagues show that λc may be calculated exactly

\[ \lambda_c = \frac{<k>}{<k^2>} \]  \hspace{1cm} (2)

and, of course, for degree distribution (1) both <k> and <k^2> are infinite (hence, γ < 3 is actually sufficient to ensure their result). Note, however, that this presumes that the network is uncorrelated (in obtaining (2) the authors assume that the degree distribution (1) applies equally to all nodes). Moreover, the network must be infinite (for otherwise a finite sample variance exists).

In [5], Small, Walker and Tse presented the first evidence that such behaviour could arise in the real world. They showed that the degree distribution of a network inferred from the global spatial temporal distribution of avian influenza outbreaks (in animal populations) cases actually did follow a scale free distribution — in fact, the scale exponent of that distribution was about 1.2 [5]. However, it was not clear that the result of Boguña et al. could be extended to this situation. We had observed that the degree distribution was highly assortative (high degree nodes tend to connect to one another) [6] and we proposed a mechanism by which such a network could arise [9]. Moreover,
the network was certainly finite, so the question of whether this was a good sampling of an underlying infinite network, or whether the sample mean and variance would imply a finite $\lambda_c > 0$ was unresolved. Moreover, we must note that this network is only inferred from time series data: we can not be certain that this necessarily captured the relevant structure in the true transmission pathways.

In addition to the connectivity model of [2] being rather idealised, the simple SIS dynamics also lacks some of the complexity of the real world. In [8] Zhang and colleagues consider the case where nodes can choose to receive vaccination, and that the decision to vaccinate is affected by node degree (that is, those with a high degree have more risk and therefore more reason to vaccinate). Under this situation, the system behaviour is the opposite of that described in [2]. That is, scale free networks, with hub nodes being more ready to vaccinate actually inhibit the spread of disease when compared to homogeneous mixing.

In this paper we report out recent attempts to address this problem from another angle. Rather than pursuing the pure scale free networks popular in the physics literature we attempt to construct a communal model of connectivity within a city — the basic definition of which is the localisation of children within schools — and we study the transmission dynamics within this system. This network is structured in such a way that it contains distinct layers (adults and children) and has both characteristics of scale free and small world networks. We examine the behaviour of disease transmission on this network to determine whether the results from computational physics can also be applied to this somewhat more realistic model.

2. The model

The purpose of this paper if to compare the behaviour of networks with more realistic transmission topologies to standard scale free complex networks in the vein of the Barabasi-Albert (BA) model [1]. We assume two distinct populations in the community: adults and children. Each child attends a school and their network connections are constrained by that school structure. Each adult is connected in a scale-free fashion to a certain number of other adults. Some adults are also parents, and in these cases they are connected to their children and to fellow parents within the same school. We assume that parents always occur in pairs, and that each adult can be a member of at most one such pairing.

Our network model necessitates a moderate number of parameters, all of which can be given reasonably “realistic” values: nonetheless, we would like to stress that our results do not sensitively depend on these choices (for a “sensible” range of values). The parameters are listed, along with our chosen values in Table 1. The network has a total of $N$ nodes, $p_{\text{kids}}N$ children (all assumed to be of school age) and the remaining $(1 - p_{\text{kids}})N$ adults. Each family has two adults and a mean $N_{\text{kids}}$ children (following a Poisson distribution). Each child is linked to exactly $\ell$ other children of which $\ell_{\text{class}}$ are in classes other than their own and $\ell_{\text{school}}$ are outside their school (that is, most children are connected only within that same school, and largely within the same class).

This creates a network of children which is both hierarchical and clustered. Children are arranged in classes, with dense links within classes. The classes are then arranged in

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_{\text{kids}}$</td>
<td>0.3</td>
</tr>
<tr>
<td>$N_{\text{kids}}$</td>
<td>2</td>
</tr>
<tr>
<td>$\ell_{\text{class}}$</td>
<td>2</td>
</tr>
<tr>
<td>$\ell_{\text{school}}$</td>
<td>1</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>5</td>
</tr>
<tr>
<td>$N$</td>
<td>$10^6$</td>
</tr>
</tbody>
</table>

Table 1: Model parameters used in the community structured complex network model.
schools, with moderate density of links between schools. Links between schools are relatively sparse, but sufficient to ensure that the resulting network is small world.

Each adult contributes \( y \) links with preferential attachment — leading to a scale free distribution. For parents \( \alpha_{\text{school}} \) of those links are restricted to be chosen from among fellow parents of the same school attended by their children. Hence, viewed in isolation from the children, the parents form an independent scale free network. However, for the adults that are also parents they are then connected directly to the hierarchical small world network of children. A typical adjacency matrix of a network constructed with this scheme is depicted in Fig. 1.

In Fig. 2 we depict the degree distribution for this network and for an equivalent BA network. The societal network has mean degree of 12.4. To construct a preferential attachment scale free network with an equivalent mean degree we add 7 links with each new node — doing so yields a scale free network with a mean degree of 14. Similarly the assortativity for the society network is 0.029; and for the preferential attachment network is −0.0050. The mean path length (computed from a random sampling of pairs of points on each network) is \( 5.57 \pm 0.54 \) and \( 4.48 \pm 0.59 \) (mean ± standard deviation of 100 samples) respectively.

3. SIS disease dynamics

For each of the networks described in the previous section we simulate the effect of SIS transmission dynamics for various rates of infectivity \( \lambda \). In Fig. 3 we report the mean infected population as a function of time over 20 realisations.

Despite the fact that both networks have similar connectivity, similar degree and similar mean path lengths (the scale free network being a little more highly connected) we do see (as anticipated by the theoretical results of [2]) that the scale free network supports a high incidence of infection. Nonetheless, the effect is rather less marked than one might have expected. For moderate of large values of \( \lambda \) the difference is only that the preferential attachment network reaches the steady state infectivity level somewhat quicker.

Admittedly, this value is comparable to the incidence of SARS in Hong Kong in 2003 [4, 3].

In both scale free and the socially structure networks the steady state level is very similar. Certainly, for moderate levels of infectivity, there is a range of values \( \lambda = 0.03, 0.04 \) and 0.05 for which the preferential attachment scale free network supports a contagion that becomes extinct on the non scale free network. However, this range is quite small and the asymptotic level of infection is rather low (between 0.01% and 0.1% of the population).

Finally, for small values of \( \lambda \) we note that both models become extinct. As noted earlier, this is to be expected, even for scale free networks. Since the networks used in this study are finite one must expect \( \lambda_c > 0 \).

Now, we turn to the central issue of the current study — what happens when a disease occurs in waves (and successive wave confer resistance)? (Or, equivalently, what is the effect of vaccination?) We modify the above SIS model by assigning an increased resistance to infection for a node based on previous infection. Let \( \alpha (0 \leq \alpha \leq 1) \) be a constant such that if a node has been infected on \( k \) previous occasions, the rate of successive infection is reduced from \( \lambda \) to \( \alpha^k \lambda \). Fig 4 illustrated typical results. Note that if \( \alpha = 1 \) this reduces to the standard SIS model, and if \( \alpha = 0 \) it is SIR.

As expected, the gradual conferment of immunity means
that the disease eventually becomes extinct. Moreover, the rate of extinction (the total duration of the infection) for the structure community network is consistently less than for the BA network. However, in the structured network we also observe a broader initial peak. That is, the peak of infection is longer for the community network than for the scale free network.

4. Conclusions

The scale free BA network is an appealing model for physicists interested in the study of transport in complex systems, and disease transmission is a useful prototypical application of this model. However, complexities in the real world do not always mirror such neat abstractions. In this work we have focussed on the comparison of this model to (perhaps) more realistic alternatives.

Our results show that the persistence one observes in SIS type dynamics in infinite BA networks is a reasonable proxy for moderately large (but finite) BA simulations. In the case of the community model proposed in this paper, we find that the level of infection is somewhat — but not drastically lower. The most marked distinction is that the disease propagates to and reaches it’s equilibrium more slowly. We also observe a somewhat larger threshold \( \lambda_c \).

These results are also mirror we we consider the case of partial immunisation — a simple proxy for a more complex model of vaccination [8]. Interestingly, in this case, the time course for the disease of both networks is very similar. The main difference being that the BA model exhibits a sharper and narrower peak, while the community model (for comparable parameter values) has a broader and lower maximum. We now need to extend this community model to ensure that it does closely reflect reality. The model structure — with separate communities of adults and children also means that it may be a useful tool to study the transmission of diseases which affect adults and children differently. This may also help us to study disease which typically break out in waves — first affecting mostly children, and later adults.

Acknowledgments

This work is supported in full by funding from the Hong Kong University Grants Council through General Research Fund grant number PolyU5300/09E. A MATLAB implementation of the algorithm to generate the societal network described in this paper is available from the author.

References

Propagation Mechanism of Phase-Inversion Wave in In-and-Anti-Phase Synchronization on 2D Lattice Oscillator

Seiko Kunihiro†, Hitoshi Aburatani†, Masayuki Yamauchi† and Yoshifumi Nishio‡

†Department of Electronics and Computer Engineering, Hiroshima Institute of Technology
2-1-1,Miyake, Saacki-ku, Hiroshima, Japan
Email: seiko.kunihiro@gmail.com
‡Department of Electrical and Electronic Engineering, Tokushima University
2-1, Minami-Josanjima, Tokushima, Japan

Abstract—We analyze synchronization phenomena on coupled oscillators systems as a ladder and a lattice. On the systems, we observed phase-inversion waves, which are phenomena of change phase states between two adjacent oscillators from in-phase synchronization to anti-phase synchronization or from anti-phase synchronization to in-phase synchronization in steady state. Some characteristics of phase-inversion waves are propagations, penetrations, reflections, and disappearances. In this paper, we discover the phase-inversion waves in in-and-anti-phase synchronization. We clarify regions which the phase-inversion wave can be observed in in-and-anti-phase synchronization. We analyze a mechanism of propagation of a phase-inversion wave in in-and-anti-phase synchronization on the lattice system.

1. Introduction
A lot of synchronization phenomena can be observed in nature world. For example, there are biological clocks, schools of sardines, the synchronization of fireflies, and so on. Recently, synchronization phenomena are researched in various fields[1]-[2].

In our previous study, we observed synchronization phenomena on coupled oscillators system. This system is made by using van der Pol oscillators which are coupled by inductor as a lattice[3]. We predicted the time-series data by using this system including nine oscillators[4]. We observed phase-inversion waves on this system including over 25 oscillators. We analyzed a mechanism of disappearance between two phase-inversion waves. Further, we analyzed a mechanism of reflection when two phase-inversion waves arrive at a corner at same time. However, these phase-inversion waves are observed in double-in-phase synchronization which all oscillators synchronize to in-phase for a vertical direction and a horizontal direction. In other hand, on ladder system, the phase-inversion waves are observed in in-and-anti-phase synchronization[5]. In-and-anti-phase synchronization is in-phase and anti-phase synchronizations exist alternately.

In this study, we observe the phase-inversion waves in in-and-anti-phase synchronization. We clarify regions which the phase-inversion wave can be observed in in-and-anti-phase synchronization when N equals 9, and clarify a mechanism of propagation of a phase-inversion wave in in-and-

anti-phase synchronization using instantaneous frequency of each oscillator and phase differences between adjacent oscillators on the lattice system.

2. Circuit model
The van der Pol oscillators are coupled by inductors $L_0$ as a lattice(see Fig. 1). The number of column and row of this system are assumed as “N + 1” respectively. We name each oscillator OSC$(k,l)$. A voltage of each oscillator is named $v_{i(k,l)}$ and a current of each oscillator is named $i_{i(k,l)}$(see Fig. 1). The circuit equations of this circuit model are normalized by Eq. (1), and the normalized circuit equations are shown as Eqs. (2)-(6).

\[
\begin{align*}
\dot{i}_{i(k,l)} &= \frac{1}{L_0} x_{i(k,l)}, \\
v_{i(k,l)} &= \frac{1}{C_0} y_{i(k,l)}, \\
t &= \sqrt{\frac{1}{L_0 C_0}}, \\
\epsilon &= \frac{1}{\sqrt{L_0 C_0}}.
\end{align*}
\]

[Corner–top] (left:$(a,b)=(0,1)$, right:$(a,b)=$(N,N − 1).)

\[
\frac{d y_{(a,b)}}{d t} = y_{(a,b)},
\]

\[
\frac{d x_{(a,b)}}{d t} = -x_{(a,b)} + \alpha(x_{(a,b)} + x_{(a,b)}) + \epsilon(y_{(a,b)} - \frac{1}{3} y_{(a,b)^3}).
\]

[Corner–bottom] (left:$(a,b)=(0,1)$, right:$(a,b)=$(N,N − 1).)

\[
\frac{d y_{(N,a)}}{d t} = y_{(N,a)},
\]

\[
\frac{d x_{(N,a)}}{d t} = -x_{(N,a)} + \alpha(x_{(N,a)} + x_{(N,a)}) + \epsilon(y_{(N,a)} - \frac{1}{3} y_{(N,a)^3}).
\]

[Center] $(0 < k < N, 0 < l < N)$

\[
\frac{d y_{(k,l)}}{d t} = y_{(k,l)},
\]

\[
\frac{d x_{(k,l)}}{d t} = -x_{(k,l)} + \alpha(x_{(k,l)} + x_{(k,l)}) + \epsilon(y_{(k,l)} - \frac{1}{3} y_{(k,l)^3}).
\]
\[ \frac{dx(a,l)}{d\tau} = y(a,l), \quad (5) \]
\[ \frac{dy(a,l)}{d\tau} = -x(a,l) + x(\alpha + 1,l) + x(k,l) - 3x(\alpha,l) + \varepsilon(y(\alpha,l) - \frac{1}{3}y(\alpha,l)^3). \]

(\text{left:}(a,b) = (0,1), \text{right:}(a,b) = (N,N-1), \text{both:} 0 < l < N.)

The \( \alpha \) corresponds to a coupling parameter. The \( \varepsilon \) corresponds to a nonlinearity of each oscillator. This system is simulated by the fourth order Runge-Kutta method and Eqs. (2)-(6). The phase-inversion waves are shown in Fig. 2. The Fig. 2–A expresses an attractor of each oscillator (current vs. voltage). The Fig. 2–B expresses itinerancy of phase difference by which sum of voltages of adjacent oscillators is shown along the time (sum of voltage vs. time).

3. In-and-anti-phase synchronization

In our circuit, an oscillator, which is not an oscillator on the edge, has four adjacent oscillators. When phase states between the oscillator and two of four oscillators are anti-phase synchronization, phase states between the oscillator and other two oscillators are in-phase synchronization. Oscillators on the edges stay in anti-phase synchronization. These phase states are called “in-and-anti-phase synchronization.” The phase-inversion waves in in-and-anti-phase synchronization are classified into two patterns. Pattern A can be observed if \( N \) is an odd number. Odd number’s phase-inversion waves propagate in vertical direction and horizontal direction respectively. Pattern B can be observed if \( N \) is an even number. Even number’s phase-inversion waves propagate in vertical direction and horizontal direction respectively. Simulation results of pattern A and B show in Figs. 2 and 3 respectively. Figure 4 shows regions which the phase-inversion wave can be observed in in-and-anti-phase synchronization when \( N \) equals 9. The coupling parameter \( \alpha \) and nonlinearity \( \varepsilon \) are changed from 0.05 to 1.0, every 0.05. The phase-inversion wave in in-and-anti-phase synchronization can be observed in region (i)(see Figs. 2 and 4). The complex phenomena on in-and-anti-phase synchronization can be observed in region (ii)(see Figs. 4 and 5).

We can observe some characteristics of phase-inversion waves in in-and-anti-phase synchronization. These characteristics are a propagation, a penetration, a reflection at an edge, and a reflection between two phase-inversion waves (see Figs. 2 and 3). These characteristics are shown in Table 1.

4. Mechanism

We analyze a mechanism of propagation of a phase-inversion wave. The mechanism is made clear by using instantaneous frequency of each oscillator and phase differences between adjacent oscillators. Figure 6 shows the signs of the initial values of the voltages and currents of each oscillator. The coupling parameter \( \alpha \) is fixed as 0.05, and nonlinearity \( \varepsilon \) is fixed as 0.15. An equation of the instantaneous frequency of \( OSC(k,l) \) is obtained as follows (see Eq. (7)). The instantaneous frequency is named \( f_{\alpha,l}(a) \) where “a” expresses the number of times of the peak value of the voltage. Time of a-th peak value of the voltage of \( OSC(k,l) \) is assumed as \( \tau_{\alpha,l}(a)(\text{see Fig. 7}) \). Similarly, \( \tau_{\alpha,l+1}(a) \) and \( \tau_{\alpha,l+1}(a) \) are decided.

\[ f_{\alpha,l}(a) = \frac{1}{\tau_{\alpha,l}(a) - \tau_{\alpha,l}(a - 1)}. \]

Three frequencies are observed in this system. To consider of the synchronizations for the vertical direction and for the horizontal direction are needed, because this system is 2 dimensional array. The in-phase synchronization and the anti-phase synchronization exist. Therefore, three types of synchronizations are observed as follows:

1. \( OSC(k,l) \rightarrow OSC(k, l+1), \) \( OSC(k, l) \rightarrow OSC(k, l-1), \)
2. \( OSC(k, l) \rightarrow OSC(k, l+1), \) \( OSC(k, l) \rightarrow OSC(k, l-1), \)
3. \( OSC(k, l) \rightarrow OSC(k, l+1), \) \( OSC(k, l) \rightarrow OSC(k, l-1), \)

...
Phenomena

The phase-inversion waves propagate for vertical direction or horizontal direction. The vertical phase-inversion waves independently move from the horizontal phase-inversion waves.

Penetrations

Two phase-inversion waves arrive at an oscillator from vertical direction and horizontal direction, and each phase-inversion wave penetrates each other.

Reflections at an edge

When a phase-inversion wave arrives at an edge, the phase-inversion wave reflects and propagates to where they came from. Sometime this phenomenon is happened with penetration.

Reflections between two phase-inversion waves

When two phase-inversion waves coming from the opposite directions arrive to two adjacent oscillator at same time, the phase-inversion waves reflect and propagate to where they came from.

Figure 5: An example of complex phenomena in region(ii) ($\alpha=0.05$ and $\varepsilon=0.85$).

Figure 6: Sign of initial value of each oscillator of in-and-anti-phase synchronization.

Table 1: Characteristics of the phase-inversion waves on in-and-anti-phase synchronization.

<table>
<thead>
<tr>
<th>Names of characteristics</th>
<th>Phenomena</th>
</tr>
</thead>
<tbody>
<tr>
<td>Propagations</td>
<td>The phase-inversion waves propagate for vertical direction or horizontal direction. The vertical phase-inversion waves independently move from the horizontal phase-inversion waves.</td>
</tr>
<tr>
<td>Penetrations</td>
<td>Two phase-inversion waves arrive at an oscillator from vertical direction and horizontal direction, and each phase-inversion wave penetrates each other.</td>
</tr>
<tr>
<td>Reflections at an edge</td>
<td>When a phase-inversion wave arrives at an edge, the phase-inversion wave reflects and propagates to where they came from. Sometime this phenomenon is happened with penetration.</td>
</tr>
<tr>
<td>Reflections between two phase-inversion waves</td>
<td>When two phase-inversion waves coming from the opposite directions arrive to two adjacent oscillator at same time, the phase-inversion waves reflect and propagate to where they came from.</td>
</tr>
</tbody>
</table>

Figure 7: The detection method of frequencies and the phase differences.

Figure 8: Transitions of phase difference and frequencies by propagation of a phase-inversion wave on in-and-anti-phase synchronization.

and-anti-phase synchronization. We fix the N as 19. Propagation mechanism is shown in Table. 2(see Fig. 8). In Fig. 8(a), the vertical axis is instantaneous frequency, and horizontal axis is time. In Fig. 8(b), the vertical axis is the phase difference, and the horizontal axis is time.

4.2. Comparison between a propagation in double in-phase synchronization and a propagation in in-and-anti-phase synchronization.

The frequency’s itineraries of propagation of the phase-inversion wave in double in-phase synchronization are Fig. 9. The frequencies are changed from $f_{\text{in}}$ to $f_{\text{anti-anti}}$.

5. Conclusion

We discovered the phase-inversion waves in in-and-anti-phase synchronization. We clarified regions which the phase-
Table 2: Propagation mechanism of a phase-inversion wave (see Fig. 8).

<table>
<thead>
<tr>
<th>no.</th>
<th>Mechanism</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>At this time, ( \Phi(9,3)(9,4) ) and ( \Phi(8,3)(8,4) ) are fixed the in-phase synchronization. In vertical direction, the phase-inversion wave, which changes synchronized state, arrives at the number of row is 10 from the number of row is 18.</td>
</tr>
<tr>
<td>1</td>
<td>A phase difference ( \Phi(9,3)(10,3) ) starts to change from the in-phase synchronization toward the anti-phase synchronization by a phase-inversion wave.</td>
</tr>
<tr>
<td>2</td>
<td>A instantaneous frequency ( f(8,3) ) starts to increase from ( f_{\text{in-anti}} ) toward ( f_{\text{anti-anti}} ) because ( \Phi(9,3)(10,3) ) starts to change from the in-phase synchronization toward the anti-phase synchronization and ( \Phi(8,2)(9,3) ) and ( \Phi(8,3)(9,3) ) are anti-phase synchronization and ( \Phi(9,3)(9,4) ) is in-phase synchronization.</td>
</tr>
<tr>
<td>3</td>
<td>( \Phi(8,3)(9,3) ) starts to change from the anti-phase synchronization toward the in-phase synchronization by ( f(8,3) ).</td>
</tr>
<tr>
<td>4</td>
<td>( f(8,3) ) starts to decrease from ( f_{\text{in-anti}} ) toward ( f_{\text{in-anti}} ) because ( \Phi(8,3)(9,3) ) starts to change from the anti-phase synchronization toward the in-phase synchronization and ( \Phi(8,3)(8,4) ) and ( \Phi(7,3)(8,3) ) are the in-phase synchronization and ( \Phi(8,2)(8,3) ) is the anti-phase synchronization.</td>
</tr>
<tr>
<td>5</td>
<td>( f(8,3) ) doesn’t arrive at ( f_{\text{anti-anti}} ) and ( f(8,3) ) starts to decrease toward ( f_{\text{in-anti}} ) again, because ( \Phi(9,3)(10,3) ) starts to change from the in-phase synchronization toward the anti-phase synchronization and ( \Phi(8,1)(9,3) ) starts to change from the anti-phase synchronization toward the in-phase synchronization and ( \Phi(9,3)(9,4) ) is the in-phase synchronization and ( \Phi(9,2)(9,3) ) is the anti-phase synchronization.</td>
</tr>
<tr>
<td>6</td>
<td>( \Phi(7,3)(8,3) ) starts to change from the anti-phase synchronization toward the anti-phase synchronization by ( f(8,3) ).</td>
</tr>
<tr>
<td>7</td>
<td>( \Phi(8,3)(10,3) ) arrives at the anti-phase synchronization and becomes fix.</td>
</tr>
<tr>
<td>8</td>
<td>( f(8,3) ) doesn’t arrive at ( f_{\text{in-anti}} ) and ( f(8,3) ) starts to increase toward ( f_{\text{in-anti}} ) again, because ( \Phi(8,3)(9,3) ) starts to change from the anti-phase synchronization toward the in-phase synchronization and ( \Phi(7,3)(8,3) ) starts to change from the in-phase synchronization toward the anti-phase synchronization and ( \Phi(8,3)(8,4) ) is the in-phase synchronization and ( \Phi(8,2)(8,3) ) is the anti-phase synchronization.</td>
</tr>
<tr>
<td>9</td>
<td>( f(8,3) ) arrives at ( f_{\text{in-anti}} ) again, because ( \Phi(9,3)(10,3) ) arrives at the anti-phase synchronization and ( \Phi(8,3)(9,3) ) starts to change from the anti-phase synchronization toward the in-phase synchronization and ( \Phi(9,3)(9,4) ) is the in-phase synchronization and ( \Phi(9,2)(9,3) ) is the anti-phase synchronization.</td>
</tr>
<tr>
<td>10</td>
<td>( \Phi(8,3)(9,3) ) arrives at in-phase synchronization and becomes fix, because ( f(8,3) ) arrives at ( f_{\text{in-anti}} ) and becomes fix.</td>
</tr>
<tr>
<td>11</td>
<td>( f(8,3) ) arrives at ( f_{\text{in-anti}} ) again, because ( \Phi(9,3)(9,3) ) arrives at the in-phase synchronization and ( \Phi(7,3)(8,3) ) starts to change from the in-phase synchronization toward the anti-phase synchronization and ( \Phi(8,3)(8,4) ) is the in-phase synchronization and ( \Phi(8,2)(8,3) ) is the anti-phase synchronization.</td>
</tr>
<tr>
<td>12</td>
<td>( \Phi(7,3)(8,3) ) arrives at anti-phase synchronization and becomes fix, because ( f(8,3) ) arrives at ( f_{\text{in-anti}} ) and becomes fix.</td>
</tr>
</tbody>
</table>

The phase-inversion wave propagates by this mechanism.

Figure 9: Transitions of frequencies by propagation of a phase-inversion wave on in-and-in-phase synchronization.

The phase-inversion wave can be observed in in-and-anti-phase synchronization when \( N \) equals 9, and clarified a mechanism of propagation of a phase-inversion wave in in-and-anti-phase synchronization by using instantaneous frequency of each oscillator and phase differences between adjacent oscillators on the lattice system, and compared between a propagation in double in-phase synchronization and a propagation in in-and-anti-phase synchronization. The frequencies of the phase-inversion wave in double in-phase synchronization are changed from \( f_{\text{in-anti}} \) to \( f_{\text{anti-anti}} \). However, the frequencies of the phase-inversion wave in in-and-anti-phase synchronization are changed around \( f_{\text{in-anti}} \) and can not change to \( f_{\text{in-anti}} \) and \( f_{\text{anti-anti}} \). We observed some characteristics of phase-inversion waves on in-and-anti-phase synchronization. These characteristics are a propagation, a penetration, a reflection at an edge, and a reflection between two phase-inversion waves.

Acknowledgements

This research is supported by the Grants-in-Aid for Young Scientists (B) (No. 19760270) from the Japan Society for the Promotion of Science.

References

Cellular Neural Networks with Hopfield Neural Networks 
Considering the Confidence Degree

Yasuhiro Ueda†, Masakazu Kawahara‡, Yoko Uwate† and Yoshifumi Nishio†

†Dept. of Electrical and Electronic Engineering, Tokushima University
2-1 Minami-Josanjima, Tokushima 770-8506, Japan
Email:{yasuhiro, kawahara, uwate, nishio}@ee.tokushima-u.ac.jp

Abstract—In this study, we propose cellular neural networks (CNN) with Hopfield neural networks (Hopfield NN) considering the confidence degree. The Hopfield NN works as an associative memory to retrieve one of the embedded patterns from the local data of the input images. The confidence degree means the difference between the retrieved pattern and the input image. When the difference is small the confidence degree is set to be large. The confidence degree works to enhance the CNN operation. By computer simulations, we investigate the basic property of the proposed method and confirm its effectiveness for example of pattern detection.

In Sec. 2, we review the basic of the standard CNN. In Sec. 3, we review the basic of the standard Hopfield neural network. In Sec. 4, we describe the structure of the space-varying CNN designed by Hopfield NN considering the confidence degree. In Sec. 5, simulation results of binary image processing task are shown. Section 6 concludes this article.

2. Cellular Neural Networks [1]

In this section, we describe the basic structure of CNN. CNN has \( M \) by \( N \) processing unit circuits called cells. Cells are arranged in a reticular pattern to \( M \) line by \( N \) row. We represent a cell \( C(i, j) \) using a variable \( i \) which denotes vertical position and a variable \( j \) which denotes horizontal position. The cell contains linear and nonlinear circuit elements. CNN is an array of cells. Each cell is connected to its neighborhood cells according to a template. Usually, the same template is used for all the cells except for boundary cells. CNN has the features of time continuity, spatial discreteness, nonlinearity and parallel processing capability.

The state equation and the output equation of the cell \( C(i, j) \) are shown as follows.

State equation:

\[
\frac{dv_{ij}}{dt} = -v_{ij} + \sum_{k=i-r}^{i+r} \sum_{l=j-r}^{j+r} A_{(i,j,k,l)} v_{kl}(t) + \sum_{k=i-r}^{i+r} \sum_{l=j-r}^{j+r} B_{(i,j,k,l)} v_{kl}(t) + I. \tag{1}
\]

Output equation:

\[
v_{ij}(t) = \frac{1}{2}(|v_{ij}(t) + I| - |v_{ij}(t) - I|). \tag{2}
\]

where \( v_x \), \( v_y \) and \( v_0 \) represent a state, an output and an input of the cell, respectively. In the Eq. (1), \( A \) is the feedback template and \( B \) is the control template. These templates and threshold value \( I \) are collectively called the general template.
The \( r \)-neighborhood of \( C(i, j) \) in CNN is defined by
\[
Nr(i, j) = \{ C(k, l) \mid \max \{ |k - i|, |l - j| \} \leq r; \quad 1 \leq k \leq M; \quad 1 \leq l \leq N \},
\]
where \( r \) is a positive integer number. In our study, we fix the value of \( r \) as 1.

3. Hopfield Neural Network Working as Associative Memory

In this section, we describe the basic structure of Hopfield Neural Network (Hopfield NN). Associative memory is a system which returns a stored pattern or its reversed pattern that is similar to an input pattern. Noisy patterns can be associated or distorted patterns can be recognized by a well-constructed associative memory.

The Hopfield NN is used as an associative memory by exploiting the property that the network has multiple stable states. Namely, if the parameters of the network can be decided in such a way that the patterns to be stored become stable states of the network, the network produces a stored pattern that is similar to an input pattern. The energy function of the Hopfield NN with \( N \) neurons and \( P \) stored binary patterns is defined by the following equation.
\[
E = -\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} w_{ij} x_i x_j + \sum_{i=1}^{N} \theta_i x_i,
\]
where \( w_{ij} \) is the weight between \( i \)-th neuron and \( j \)-th neuron, and \( \theta_i \) is the threshold of the \( i \)-th neuron. The weight \( w_{ij} \) is given as follows.
\[
w_{ij} = \begin{cases} \sum_{p=1}^{P} x_i^{(p)} x_j^{(p)} & (i \neq j) \\ 0 & (i = j) \end{cases}
\]
The weight \( w_{ij} \) is 0 when \( i = j \), because all the units have combined with all the units of the others except themselves. The states of the neurons are asynchronously updated due to the following difference equation:
\[
x_i(t+1) = \text{sgn}(\sum_{i \neq j} w_{ij} x_j(t)),
\]
where \( \text{sgn} \) is an output function as follows:
\[
\text{sgn}(a) = \begin{cases} 1 & (a \geq 0) \\ -1 & (a < 0) \end{cases}
\]

4. Space-Varying Cellular Neural Networks Considering the Confidence Degree

In the previous study, we have proposed space-varying cellular neural networks which is designed by using the ability of the associative memory of Hopfield NN. In general, the design of space-varying systems is not easy. However, we can set one of prepared existing templates on each cell of CNN according to the retrieved pattern by Hopfield NN to which some typical local image structures are embedded. Namely, we need only some existing templates and their associated patterns. The design method is described as follows. Firstly, we prepare some two-dimensional arbitrary patterns for the memory of Hopfield NN and some existing templates related with these patterns, respectively. Secondly, Hopfield NN memorize the some two-dimensional arbitrary patterns. Thirdly, Hopfield NN compare between memorized the some arbitrary patterns and each pixel and its two neighborhood pixels of a input image. Hence, from some arbitrary patterns, Hopfield NN associate the certain arbitrary pattern which is the most similar to the pattern of each pixel and its two neighborhood pixels. Fourth, from the prepared some existing templates, the template of each cell in CNN is designed based on the associated pattern. Finally, input image is processed by using the space-varying CNN designed by Hopfield NN. Each templates are applied to each cell by these steps.

In this study, we propose the developing system by considering the confidence degree to space-varying CNN. We calculate the Hamming difference between the retrieved pattern and the input image. When the Hamming difference is small the confidence degree is set to be large. The confidence degree works to enhance the CNN operation. By using this system, it is possible to detect the some objects by changing the threshold value when several similar objects exist in the input image. The threshold value \( I_{cd} \) is defined as following in Eq. 8 and the characteristics of the threshold value \( I_{cd} \) with the Hamming distance and the confidence degree is shown in Fig. 1.

\[
I_{cd} = \frac{13 \times \text{Confidence Degree}[\%]}{100(1 + \exp(HD))}
\]

In Eq. (8), \( HD \) is calculated by Eq. (9), and Confidence Degree[\%] is a parameter to change the characteristics of the threshold value \( I_{cd} \).
\[
HD = -\frac{1 \times \text{Hamming Distance} + 13}{5}
\]

5. Simulation Results

In order to confirm the effectiveness of this proposed method, the CNN application of the object detection is simulated. In this simulation, we use “Logic AND” template. This template reads out the black when input value and initial value are black. The others situation, the output becomes white. Therefore, when the input image and the initial state are same, the output is same with input image. However, if the threshold value of “Logic AND” template
is changed smaller than -2.1, CNN outputs white regardless of the input image and the initial state.

In this section, we show some simulation results using the proposed CNN. The input binary image is 4 and its size is 64 by 64. First, we describe the memorized patterns and related templates. And templates are found in [8].

"Logic AND" template:

\[
A = \begin{bmatrix}
0 & 0 & 0 \\
0 & 2 & 0 \\
0 & 0 & 0
\end{bmatrix}, \quad B = \begin{bmatrix}
0 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 0
\end{bmatrix}, \quad I = -1. 
\] (10)

"White filler" template:

\[
A = \begin{bmatrix}
0 & 0 & 0 \\
0 & 2 & 0 \\
0 & 0 & 0
\end{bmatrix}, \quad B = 0, \quad I = -4. 
\] (11)

This template of Eq. (11) changes all outputs into -1.

The memorized patterns are set as shown in Fig. 2. Four patterns which are Fig. 2 from the left are related with “Logic AND” template. These patterns corresponds to the part of objects in the input image. The rightmost pattern is related with “White Filler” template. This pattern describe the uniform region. By using these patterns, wiring weights of Hopfield NN is decided and each templates are applied to each cells.

Moreover, by changing the confidence degree which is 50 [%], 70 [%] and 80 [%], the extra threshold value \( I_{cd} \) is decided like Eq. 8 and Fig. 1. State equation of CNN updated is described as follows.

\[
\frac{dv_{xj}}{dt} = -v_{xj} + \sum_{k=i-r}^{i+r} \sum_{l=j-r}^{j-r} A_{n(i,j,k,l)}v_{ykl}(t)
\]

\[
+ \sum_{k=i-r}^{i+r} \sum_{l=j-r}^{j-r} B_{n(i,j,k,l)}v_{ykl}(t) + I - I_{cd}. \tag{12}
\]

By using the extra threshold value \( I_{cd} \), simulation results are changed as shown in Fig. 4 (a), (b) and (c), respectively. Therefore, some points of the object are detected by changing the confidence degree. Additionally, some points which have same pattern with memorized pattern is detected irrespectively the confidence degree.

![Figure 3: Input binary image to CNN and Hopfield NN.](image)

![Figure 4: Output image by using proposed system.](image)

(a) Confidence degree is 50 [%]. (b) Confidence degree is 70 [%]. (c) Confidence degree is 80 [%].

By using this system, we detected some points of each objects. Additionally, by changing the confidence degree, the number of the detected points are changed. Moreover, we carry out “Recall” template with conventional CNN to Fig. 4.
Figure 5: Change of detected points by Confidence degree.

"Recall" template:

\[
A = \begin{bmatrix}
0.3 & 0.3 & 0.3 \\
0.3 & 4 & 0.3 \\
0.3 & 0.3 & 0.3
\end{bmatrix}, \quad B = \begin{bmatrix}
0 & 0 & 0 \\
0 & 5.1 & 0 \\
0 & 0 & 0
\end{bmatrix}, \quad I = 0.
\] (13)

Simulation results are shown as follows (Fig. 6).

(a) (b) (c)

Figure 6: Output image by applying the "Recall" template with conventional CNN to Fig. 4. (a) Recall the input image in Fig. 4 (a). (b) Recall the input image in Fig. 4 (b). (c) Recall the input image in Fig. 4 (c).

From this simulation results, we can change the type of the detected objects by changing the confidence degree. Additionally, we could detect the object which is including the same pattern with the memorized pattern.

6. Conclusions

In this study, we proposed cellular neural networks (CNN) with Hopfield neural networks (Hopfield NN) considering the confidence degree. The Hopfield NN worked as an associative memory to retrieve one of the embedded patterns from the local data of the input images. The confidence degree meant the difference between the retrieved pattern and the input image. When the difference was small the confidence degree was set to be large. The confidence degree worked to enhance the CNN operation. By computer simulations, we investigated the basic property of the proposed method and confirmed its effectiveness for several examples.

References

Characteristics of Cellular Neural Networks with Dynamic Template for Motion Pictures

Masakazu KAWAHARA†, Yoko UWATE† and Yoshifumi NISHIO†

Email:{kawahara, uwate, nishio}@ee.tokushima-u.ac.jp
†Dept. Electrical and Electronic Eng., Tokushima University

Abstract—In our previous research, cellular neural networks with dynamic template (D-CNN) have been proposed. In D-CNN, the wiring weights of template are dynamically changed at each update by learning. In this study, we investigate the characteristics of the update template in D-CNN when input images are changed during D-CNN process. We express the variation of the update template through the gray scale. We also calculate the increasing or decreasing ratios of element in the update template. From these obtained results, we confirmed that D-CNN is effective for motion pictures.

1. Introduction

Cellular neural networks (CNN) were proposed by Chua and Yang in 1988 [1]. The idea of CNN was inspired from the architecture of the cellular automata and the neural networks. Unlike the original neural networks, the CNN has local connectivity property. Wiring weights of the cells are established by parameters called the template. The performance of the CNN is decided by the template. Also, the CNN has been successfully used for various high-speed parallel signals processing applications such as image processing application [2][3]. Usually, the templates of all the cells in the CNN are identical and those values do not change during the processing. This is good for implementation but restrict the performance, namely the conventional CNN cannot perform image processing based on the local features of input images.

In the previous study, we have proposed cellular neural networks with dynamic template (D-CNN) [4]. In D-CNN, template is dynamically changed at each update by learning. This learning method is inspired from the rank order learning. The updated template depends on the value of cells. From the simulation results of the previous study, we confirmed that the converged value of each cell is divided to two or three values. Also, convergence process is much more rapid than that of the conventional CNN. Then, we investigate update template in D-CNN for motion pictures [5]. However, the mechanism of D-CNN has not been made clear. In this study, we investigate the characteristic of the updated template using CNN with dynamic template in detail. We also set to threshold value initial template for comparison of the updated template. We expect the appearance of characteristics in updated template by changing input image. From the investigation of the value of the updated template, we confirmed the characteristic of updated template in D-CNN by changing input images. The rest of this paper is structured as follows. In the Sec. 2, we review the basic of the standard CNN. In the Sec. 3, we explain the algorithm of the proposed D-CNN. In the Sec. 4, we show the characteristic of the updated template in simple binary images using D-CNN. In the Sec. 5, we show the characteristic of the updated template in real gray scale images using D-CNN. The Section 6 concludes the article.

2. Cellular Neural Networks [1]

In this section, we explain the basic structure of the CNN. The CNN has $M$ by $N$ processing unit circuits called cells. Cells are arranged in a reticular pattern to $M$ line $N$ row. We represent a cell $C(i, j)$ using a variable $i$ which denotes vertical position and a variable $j$ which denotes horizontal position. The cell contains linear and nonlinear circuit elements. The CNN is an array of cells. Each cell is connected to only its neighboring cells according to a template. Usually, the template is the same for all cells except for boundary cells. The CNN has the features of time continuity, spatial discreteness, nonlinearity and parallel processing capability.

State equation:

$$\frac{dv_{xij}}{dt} = -v_{xij} + \sum_{k=i-r}^{i+r} \sum_{l=j-r}^{j+r} A_{i,j,k,l}v_{xkl}(t)$$
$$+ \sum_{k=i-r}^{i+r} \sum_{l=j-r}^{j+r} B_{i,j,k,l}v_{ukl}(t) + I.$$  \hspace{1cm} (1)

Output equation:

$$v_{xij}(t) = \frac{1}{2}(l|v_{xij}(t) + 1| - |v_{xij}(t) - 1|).$$  \hspace{1cm} (2)

where $v_{e}$, $v_{s}$ and $v_{a}$ represent a state, an output and an input of cell, respectively. In the equation (1), $A$ is the feedback template and $B$ is the control template. These and bias $I$ are collectively called general template. In this equation, the control template depends on input value.
3. CNN with dynamic template

In this section, we explain the algorithm of D-CNN. In our research, we change input images when a certain calculation times comes. In our D-CNN, the templates are updated at every iterations by rank order learning. The learning steps in our D-CNN are described as follows.

**STEP 1:** The state values and the output values of all the cells in D-CNN are updated according to the discretized model of Eqs. (1) and (2).

**STEP 2:** Calculate the comparison of the output value of each cell with the one-step-past outputs of the cell and its neighbor cells. The comparison equation for the cell \((i, j)\) is described Eq. (3).

**Comparison Equation:**

\[
\text{Diff}(i; j; k, l) = |y_{\text{past}}^{i,j,k,l} - y_{\text{now}}^{i,j,k,l}|.
\]

**STEP 3:** Among the 9 calculated values of \(\text{Diff}(i; j; k, l)\), the cells with the smallest and the second smallest values are defined as “winner” and “second”, respectively. In our update algorithm, we change the learning rate in two elements. By this step, we find the position of cells with the nearest and the second nearest values to the corresponding cell \((i, j)\).

**STEP 4:** Update the elements of the template corresponding to the positions of the “winner” and the “second”. Note that in our proposed learning algorithm only two elements are updated. The update method and the update function are described as follows.

**Update Method:**

Assume that the template before update is given as Eq. (4).

**Template now:**

\[
A_{\text{now}}^{(i,j)} = \begin{bmatrix}
    a_{11}^{\text{now}} & a_{12}^{\text{now}} & a_{13}^{\text{now}} \\
    a_{21}^{\text{now}} & a_{22}^{\text{now}} & a_{23}^{\text{now}} \\
    a_{31}^{\text{now}} & a_{32}^{\text{now}} & a_{33}^{\text{now}}
\end{bmatrix},
\]

\[
B_{\text{now}}^{(i,j)} = \begin{bmatrix}
    b_{11}^{\text{now}} & b_{12}^{\text{now}} & b_{13}^{\text{now}} \\
    b_{21}^{\text{now}} & b_{22}^{\text{now}} & b_{23}^{\text{now}} \\
    b_{31}^{\text{now}} & b_{32}^{\text{now}} & b_{33}^{\text{now}}
\end{bmatrix},
\]

\[
p_{\text{now}}^{(i,j)} = p_{\text{now}}.
\]

For example, we consider the case that the “winner” is \((i, j)\) and the “second” is \((i - 1, j - 1)\). In that case, only \(a_{22}^{\text{now}}, b_{22}^{\text{now}}, a_{11}^{\text{now}}\) and \(b_{11}^{\text{now}}\) in Eq. (4) are updated. The threshold value \(l\) is not updated in our learning method.

In our update algorithm, we change the learning rate in two elements. The learning rates of the “winner” and the “second” are shown as follows.

**Learning rate:**

\[
R_1 = R_{10} \left(1 - \frac{\text{Number of calculation}}{\text{Number of calculation}_{\text{max}}} \right),
\]

\[
R_2 = R_{20} \left(1 - \frac{\text{Number of calculation}}{\text{Number of calculation}_{\text{max}}} \right).
\]

In this study, we decide \(\text{Number of calculation}_{\text{max}}\) in Eqs. (5) and (6) to be set to 10. Namely, the learning rates of “winner” and “second” are changed until 10 calculations. Then, after \(\text{Number of calculation}_{\text{max}}\) becomes over 10, the learning rates of “winner” and “second” become 0 and the templates are not updated. By using the learning rate, the elements of the template are updated according to the following update equation.

**Update Equation:**

\[
a_{\text{winner}}^{\text{updated}} = a_{\text{winner}}^{\text{now}} - R_1 (v_{\text{past}}^{(i,j)} - v_{\text{now}}^{(i,j)}),\]

\[
a_{\text{second}}^{\text{updated}} = a_{\text{second}}^{\text{now}} - R_2 (v_{\text{past}}^{(i,j)} - v_{\text{now}}^{(i,j)}).
\]

\(R_1\) and \(R_2\) decrease according to the Eqs. (5) and (6). The initial learning rates are given as follows.

**Initial Learning rate:**

\[
\text{Winner} : R_{10} \quad (0 \leq R_{10} \leq 0.1).
\]

\[
\text{Second} : R_{20} = R_{10}/4.
\]

After the update using Eqs. (7) and (8), the updated template is shown as follows. In Eq. (11), \(a_{11}^{\text{updated}}\) and \(a_{22}^{\text{updated}}\) are the updated values. Also, \(b_{11}^{\text{updated}}\) and \(b_{22}^{\text{updated}}\) are updated similarly.

**Template updated:**

\[
A_{\text{updated}}^{(i,j)} = \begin{bmatrix}
    a_{11}^{\text{updated}} & a_{12}^{\text{updated}} & a_{13}^{\text{updated}} \\
    a_{21}^{\text{updated}} & a_{22}^{\text{updated}} & a_{23}^{\text{updated}} \\
    a_{31}^{\text{updated}} & a_{32}^{\text{updated}} & a_{33}^{\text{updated}}
\end{bmatrix},
\]

\[
B_{\text{updated}}^{(i,j)} = \begin{bmatrix}
    b_{11}^{\text{updated}} & b_{12}^{\text{updated}} & b_{13}^{\text{updated}} \\
    b_{21}^{\text{updated}} & b_{22}^{\text{updated}} & b_{23}^{\text{updated}} \\
    b_{31}^{\text{updated}} & b_{32}^{\text{updated}} & b_{33}^{\text{updated}}
\end{bmatrix},
\]

\[
p_{\text{updated}}^{(i,j)} = p_{\text{now}}.
\]

**STEP 5:** The steps from 1 to 4 are repeated.

These learning steps inspired from the rank order learning.

4. Investigation Results

In this section, we show the investigation results for two types of motion pictures. We express the variation of the updated template in D-CNN through the graduation value. In the first step of this investigation, an initial template is set to D-CNN. The elements of the initial template are updated by using updated method in the previous section. The initial template used in this study is described as follows.

**Initial Template:**

\[
A = \begin{bmatrix}
    0 & 0 & 0 \\
    0 & 0 & 0 \\
    0 & 0 & 0
\end{bmatrix},
\]

\[
B = \begin{bmatrix}
    0.07 & 0.1 & 0.07 \\
    0.1 & 0.32 & 0.1 \\
    0.07 & 0.1 & 0.07
\end{bmatrix},
\]

\(I = 0.\)

4.1. The Updated Template in Binary images

Firstly, we investigate updated template for simple binary image. We change 4 input images every 10 \(\tau\) like a motion picture.
Figure 1: Motion pictures 1. (a) Input image 1. (b) Input image 2. (c) Input image 3. (d) Input image 4.

Figure 1 shows binary input images for motion pictures. In Fig. 1, we recognize the black object appeared and moved up to down through Figs. 1(a) to (d). By changing these input images, we investigate the characteristic of the updated templates.

In this investigation, we show that how to update the dynamical template. Namely, we calculate the difference of elements in updated template from the initial template. We set to the threshold value in updated template. The threshold value of A and B templates are described as follows.

Threshold value of template:

\[
Th_A(i, j) = \sum_{i=1}^{3} \sum_{j=1}^{3} a_{ij}
\]  
(12)

\[
Th_B(i, j) = \sum_{i=1}^{3} \sum_{j=1}^{3} b_{ij}.
\]  
(13)

Namely, the threshold value is summed all of the elements in the each template. From our update algorithm, the rate of update is same rate in \(Th_A\) and \(Th_B\).

Figure 2 shows the variation of the update template through the gray scale. In Fig. 2, if the average of update template is over \(Th\), the pixels near to black. While, if the average of update template is under \(Th\), the pixels near to white. From these simulation results, we can confirm that the dynamical templates of the changing area from the previous image is more updated than the constant image area.

Figure 3 shows the percentage of updated template whether the value of all the elements of updated template under \(Th\) and over \(Th\). In these graphs, we can say that the number of the update template is varied by changing input image.

4.2. The Updated Template in Gray Scale Images

Next, we investigate updated template for gray scale image. Similar to the previous investigation, we change 4 input images every 10 \([\tau]\) like a motion pictures.

Figure 4 shows the input images as motion pictures. We confirm the doll is appeared and gradually move left to right through Figs. 3(a) to (d). Using these input images, we investigate the characteristic of update template in D-CNN.

Figure 5 shows the variation of the update template through the graduation value with gray scale. In Fig. 5, the area of the edge is most changed by changing input image. From these results, we can say that the edge of new object is more updated by inputted next input image.

Next, we investigate the process of the number of the updated template. Figure 6 shows the the number of the updated template is over \(Th\) and under \(Th\). In Fig. 6, the number of template over \(Th\) decreases gradually. On the other hand, the number of template under \(Th\) increases gradually. From these results, we confirm that the average of the changing rate in updated template is varied widely when the next input image is inputted. Even if next input image is inputted, the updated template influence of before input image. Therefore, we can say that our proposed CNN is effective for the motion picture processing.

5. Conclusions

In this study, we have investigated the characteristics of the update template in D-CNN when input images were changed during D-CNN process. In D-CNN, the template was changed by rank order learning. We have expressed the variation of the update template through the gray scale. We also calculate the increasing or decreasing ratios of element in the update template. From instigation results, we have confirmed that the D-CNN is effective for the application of motion pictures.
Figure 3: The number of updated template in Fig. 1. (a) The percentage of template where under $T h_{(i,j)}$. (b) The percentage of template where over $T h_{(i,j)}$.

Figure 4: Input images. (a) Input image1. (b) Input image2 (after 10 $\tau$). (c) Input image3 (after 10 $\tau$). (d) Output image before input image is changed.

Figure 5: The graduation of the updated template. (a) The updated template during simulation. (b) The final updated template.

Figure 6: The number of updated template in Fig. 5.

References


DT-CNN Annealing with Additive Noise Generated by Class 3 CA: A Comparison with Chaos Annealing

Tomohiro FUJITA† and Takeshi OGURA†

†Ritsumeikan University 1-1-1 Noji-higashi, Kusatsu, Shiga, Japan
Email: [tfujita, togura]@se.ritsumei.ac.jp

Abstract — An annealing for global optimization on discrete time cellular neural network (DT-CNN) with additive noise is investigated. In our method the additive noise is generated using chaotic behavior of class 3 cellular automata (CA). The optimization with our method shows higher performance than with uniform random noise. In this paper we investigate the relation between our method and chaos annealing, which is another superior optimization method with neural network. The auto-correlations of the CA noise and logistic map, which is often used in chaos annealing, are evaluated, and we can see the similarity of both systems. The annealing performances are also evaluated, and the results with the CA noise and logistic map show both methods exhibit higher performances under the same condition. These results may suggest that the CA noise and logistic map have common basis and it makes the ability of global search high.

1. Introduction

Combinatorial optimization problems arise in a lot of scientific and technological fields. In this paper we investigate a global optimization with discrete time cellular neural network (DT-CNN).

Many researches apply the optimization on DT-CNN to various problems to date. One of examples is solving minimization problem of spin glass energy by DT-CNN [2]. In statistical physics, with the Ising model [3] the minimization problem of spin glass energy is formulated as a quadratic assignment problem which is an NP-hard problem. This formulation has the same structure of Lyapunov function of DT-CNN, so this can be solved by DT-CNN. In recent years the Ising model is applied in probabilistic information processing [4], and in some applications this method achieves great success. Therefore solving this class of problems is paid much attention from the image processing field. Another example is an image coding and decoding [5]. The image coding and decoding are formulated as an optimization problem and are solved by DT-CNN.

The drawback of the optimization with DT-CNN is that in many cases the state of the network is trapped at a local minimum, and thus a global solution cannot be found. To overcome this difficulty we proposed an annealing method on a DT-CNN that realizes global optimization [6, 7]. In this scheme, noise is induced into network dynamics then gradually reduced. In this process, the state of the network is initially random but eventually becomes convergent. Due to the randomness of the noise, the network escapes from local minima.

In previous work [8] we proposed a hardware-oriented method of noise generation. The noise is generated using the chaotic behavior of class 3 cellular automata (CA) [1] on Cellular AutoMata on Content Addressable Memory (CAM²). CAM² is a dedicated hardware for CA and CNN [9, 10], so that the noise can be generated easily. We also showed the annealing performance with the noise generated by chaotic behavior of CA (CA noise) is superior to that with uniform random noise [7].

Preceding studies reported that the Hopfield Neural Network with the chaotic noise as the additive noise increases the optimization ability [11, 12]. It was pointed out that this ability relates to the auto-correlation of the chaotic noise [13, 14]. Hasegawa and Umeno showed the noise with auto-correlation which has negative auto-correlation at first data lag and gradually decays has high solving ability of minimization [14]. Our system also uses the chaos of class 3 CA, and so the CA noise may have the common property which the chaos noise has.

In this paper we take up logistic map as a representation of chaos, and investigate whether both our method and logistic map have common statistical property. The auto-correlation of the CA noise is compared with that of logistic map. The performance of annealing is also compared CA noise and logistic map.

2. Noise Induced DT-CNN Model

DT-CNN is a temporally discretized CNN. The DT-CNN consists of an $M \times N$ rectangular array of cells $C(i, j), i = 1, 2, \ldots, M, j = 1, 2, \ldots, N$. These cells have three variables $u_{i,j}, x_{i,j}$ and $y_{i,j}$, denoting input, state and output, respectively. The dynamics of the DT-CNN takes the following form:

$$x_{i,j}(t+1) = \sum_{C(k,l)\in N(i,j)} A(i,j,k,l) y_{i,j}(t)$$
\[
\sum_{C(k,l) \in N_r(i,j)} B(i,j;k,l) u_{i,j} + I
\]

\[
y_{i,j}(t) = \frac{1}{2} \left( |x_{i,j}(t) + 1| - |x_{i,j}(t) - 1| \right) 
\]
where \( N_r(i,j) \) is a set of neighborhood cells and \( A(i,j;k,l) \) and \( B(i,j;k,l) \) are parameters called templates. \( I \) is also a parameter called the threshold value.

We can define the Lyapunov function \( E(t) \) of the DT-CNN as follows [15].

\[
E(t) = \frac{1}{2} \sum_{(i,j)} y_{i,j}^2(t) 
- \frac{1}{2} \sum_{(i,j)} \sum_{(k,l)} A(i,j;k,l) y_{i,j}(t)y_{k,l}(t) 
- \sum_{(i,j)} \sum_{(k,l)} B(i,j;k,l) u_{i,j} + I 
- \sum_{(i,j)} I y_{i,j}(t) 
\]

The amplitude of noise is controlled by \( a(t) \), which decreases exponentially in accordance with Eq. (5). \( \delta \) controls the speed of damping.

Noise is added to the dynamics of the DT-CNN to search for a global minimum. Since state \( x_{i,j} \) fluctuates randomly in the presence of noise, escaping from local minima becomes possible. As the noise becomes smaller, the state of the DT-CNN becomes stable at an optimal minimum or suboptimal minima.

3. Noise Generator

To implement the DT-CNN with noise on hardware, a generating system of noise \( n_{i,j} \) is required. We propose the use of two-dimensional CA as a noise generator [8]. CA are computational models proposed by Neumann [1] and consist of lattice-shaped cells. Since the architecture of CA is similar to that of the DT-CNN, we can implement CA on a universal CNN machine, CAM2. Figure 1 shows the concept of the DT-CNN with noise, which has two layers: a DT-CNN layer and a CA layer.

The states of the cells vary on the basis of the state transition function \( F \), which is determined by the rule number \( R \) as follows:

\[
R = \sum_{i,j=0}^{N} F(v_{i,j}, n) \times 2^{v_{i,j} + 2n} 
\]

where \( v_{i,j} \) is the state of cell \( (i,j) \) and takes binary values, \( n \) is the number of neighborhood cells with state \( v_{i,j} \) equal to 1 and \( N \) is the total numbers of neighborhood cells. Wolfram [1] sorted state transition functions into four classes. The functions in class 3 have chaotic behavior. Since we require disordered noise, we use CA in class 3 as the noise generator. In this paper, \( R = 143954 \).

We generate noise \( n_{i,j} \) from the CA as follows.

\[
n_{i,j}(t) = \left( \frac{v_{i,j-1} - 1}{2^8} + \frac{v_{i,j-1} + 1}{2^6} + \frac{v_{i,j-1}}{2^5} + \frac{v_{i,j+1} + 1}{2^3} \right) \times 2^{v_{i,j+1} - 1} - 1 
\]

The range of \( n_{i,j} \) is \([-1, 1]\). Figure 2 shows the process by which the noise is generated.

4. Auto-correlation of CA noise

Hasegawa et al. [13] analyzed which characteristic of the chaos affects on the annealing performance using surrogates. From their experimental results, the noise which
preserves the auto-correlation of chaotic noise exhibits high solving ability. This shows that temporal structure of the chaos affects on the annealing performance. We analyze the auto-correlation of CA noise and investigate whether CA noise has the same temporal structure as logistic map.

The auto-correlation of time-series \( \{x_0, x_1, \cdots \} \) is defined as
\[
A_k = E[(x_0 - \mu)(x_k - \mu)],
\]
where \( k \) is a time-shift from original data. Figure 3 shows the auto-correlation of CA noise. In Fig. 3 the auto-correlations of logistic map and uniform random number are also depicted. The auto-correlation of uniform random number is 0 at \( k \neq 0 \), because there is no correlation among the time series of the random number. On the other hand, in the logistic map, the correlation coefficients at \( k = 1 \) is negative and it gradually decays. Hasegawa and Umeno pointed out that this characteristic shape of the auto-correlation makes the solving ability of optimization high [14].

The auto-correlation of CA noise also takes negative auto-correlation coefficient at \( k = 1 \). That feature is similar with logistic map, but the auto-correlation decays so fast that the coefficients is nearly equal to zero at \( k \geq 2 \).

5. Annealing with CA noise and logistic map

We use three types of noise sources, namely, CA noise, logistic map and uniform random number, as the noise in Eq. (4). In this experiment we use the following templates.

\[
A = \begin{bmatrix} 0.0 & 1.0 & 0.0 \\ 1.0 & 3.0 & 1.0 \\ 0.0 & 1.0 & 0.0 \end{bmatrix}, \quad B = \begin{bmatrix} 0.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \end{bmatrix}, \quad I = 0
\]

(8)

The input variable \( u \) is randomly generated within \([−U, U]\) to create various problems, where \( U \) is the range of random numbers. We apply our model (Eq.(4)) on DT-CNN with cell size of 100 × 100.

Figure 4 is the result of the 100 trials with various initial conditions, and in this case \( U \) is equal to 1. The x axis is the serial number for the trials and the y axis is the convergent energy of the network. In Fig. 4 the green, red and blue lines show the result of the annealing with the CA noise, logistic map and uniform random number, respectively. This results show in CA noise and logistic map the 100 trials converge to one solution and these convergent energies with two methods are exactly same. We also tried to solve the same problem in GA, but we could not find better solution than this. Comparing with the uniform random number, CA noise and logistic map find the better solution in all 100 trials.

We change the parameter \( U \) into 4, and the experimental results by this parameter show in Fig. 5. In Fig. 5, the results using the CA noise and logistic map do not show the convergency into one solution. In all 100 trials, the convergent energies with the CA noise and logistic map are worse than that with the random noise.

Randomly generating the input variable \( u \) within the range of \([−U, U]\), we also try the same evaluation on the 100 various problems. When \( U \) is equal to 1, in the all problems, the annealing results with the CA noise and logistic map show the convergence to one solution, and they obtain better solution than the random number. On the other hand, when \( U = 4 \), in all problems, the convergence to one solution does not reveal, and the random number obtains better solutions than the CA noise and logistic map.

We summarize the above results. When \( U \) is equal to 1, the annealing with both the CA noise and logistic map shows higher performances than with the uniform random number. In the case with the CA noise and logistic map, the network converges to exact one solution in all trials from 100 initial conditions. As \( U \) changes into 4, the annealing performances with the CA noise and logistic map degrade. Their performances are worse than the random number.

The results with the CA noise and logistic map show the same tendency in annealing performance. That may suggest that they have the common basis and it makes solving ability higher.

6. Conclusion

From our analyses the CA noise has some common natures with logistic map. Firstly, the auto-correlation coefficient of the CA noise takes negative value at the first time lag. Logistic map also has this shape of the auto-correlation and the noise which takes this shape of auto-correlation has high ability of global search. Secondly, both the CA noise and logistic map reveal high performances of DT-CNN annealing under the same condition. This may suggest the common principle of both systems makes the ability of global search high.
Figure 4: Convergent energy in the case of $U = 1$ (Cell size = 100 × 100)

Figure 5: Convergent energy in the case of $U = 4$ (Cell size = 100 × 100)

References


Can you achieve any function with a 2-neuron CNN?

Mireia Vinyoles-Serra† and Xavier Vilasí-Cardona†
†LIFAELS, laSalle, Universitat Ramon Llull
Pg. Bonanova 8, 08022 Barcelona, Spain
Email: mireitav@salle.url.edu, xvilas@sales.url.edu

Abstract—We recover the analysis of the response of the 2-neuron CNN to its external inputs in the stable symmetric case. From this we study which binary functions can be implemented either directly, either by composing templates. The results show that some particular combinations can not be achieved. One of this functions is found when implementing a universal Turing machine.

1. Introduction

Cellular Neural Networks (CNNs), as introduced in [1], [2] are nonlinear dynamical systems completely stable for certain parameter range. Their behavior is defined by the CNN parameters usually called cloning template yet sometimes a single cloning template is not enough to solve a particular problem. In this case, a template combination can be used. But the necessary templates in order to reproduce some input-output functional relations is not clear yet. So we study the simplest but rich case: the two neuron CNN.

Taking a symmetric set of weights, and the self-feedback coefficient larger than one, the state variables always converge to ±1. This stability results allow to establish relations between the CNN parameters and the final outputs [3], and so aboard the template design and template composition problems. From this relations we set which combinations are possible with a single template, composing templates and which are unreachable. For instance, we shall see how one of the state transitions of the 4-symbol, 7-state universal Turing machine [4] is impossible to be implemented with a two neuron CNN.

2. Convergence map

Focusing our study in the two neuron Cellular Neural Network, we first define our notation for the piecewise linear CNN system as:

\[
\begin{align*}
\dot{x}_0 &= -x_0 + sy_0 + p_x y_1 + b_0 u_0 + b_x u_1 + I, \\
\dot{x}_1 &= -x_1 + sy_1 + p_{-} y_0 + b_{-} u_0 + b_0 u_1 + I,
\end{align*}
\]

(1)

where \( x_i \) are the internal states of the neuron and are taken in \([-1, 1]\). Variables \( y_i \) are the external states defined by the piecewise linear function, \( f(x_i) = \frac{1}{2}((x_i + 1) - |x_i(t) - 1|), i = 0, 1 \). The external inputs are \( u_i \) and they shall be constant in time. Our analysis uses \( u_i \)'s as the input variables and the final stable state \( y_i \)'s as output. The other parameters \((s, p_{\pm}, b_0, b_\pm, b_{\pm}, I)\) configure the network cloning template. Along the paper, we will use the notation,

\[
\begin{pmatrix}
 u'_0 \\
u'_1
\end{pmatrix}
= \begin{pmatrix}
b_0 & b_x \\
b_{-} & b_0
\end{pmatrix}
\begin{pmatrix}
 u_0 \\
u_1
\end{pmatrix}
+ \begin{pmatrix}
 I \\
1
\end{pmatrix}
\]

(2)

so that, the action of \( B \) on \((u_0, u_1)\) to obtain \((u'_0, u'_1)\) will be called \( B\)-transformation.

In order to study the template influence on the CNN function, we must work in a parameter range where the system converges to a fixed-point. From Lyapunov theory, it is known that for \( s > 1 \) and \( p_\pm = p_- = p \), the system converges to one of the four corner points \( S = \{±1, ±1\}\). This particular convergence set allows to abstract classification problems using the Lyapunov function defined as

\[
L(y_0, y_1) = -py_0y_1 - \frac{1}{2}(y_0^2 + y_1^2) - u'_0y_0 - u'_1y_1. 
\]

(3)

\(L(y_0, y_1)\) is a monotone decreasing function and bounded from below, so the CNN system converges to the point where \(L(±1, ±1)\) is minimum [1]. The comparison of this four values gives us the necessary convergence conditions and let’s choose the adequate CNN parameters in order to guarantee some desired input-output relation. To do it, we first fix the initial conditions at \((0, 0)\) and then compare the four possible output values of \(L(±1, ±1)\) in order to find in which \(L\) takes lower value.

The Lyapunov function (3) takes the minimum value at \((±1, ±1)\), this is \(L(±1, ±1) \leq L(i, j)\) \( i, j = 1, −1 \) if and only if parameters fulfill equations: \( u'_0 \geq -p \), \( u'_1 \geq -p \) and \( u'_0 + u'_1 \geq 0 \). Plotting this region in the \((u'_0, u'_1)\)-plane we obtain a convergence map and then applying the B-transformation (2) in order to find the correspondent region in the \((u_0, u_1)\)-plane, we obtain a convergence map in Figure 1.

From these maps, we may fix the parameters of the CNN to obtain the relation between the external inputs and the final outputs. Taking inputs \( u_i \) inside a convergence region where \(L(i, j)\) is minimum, the system will converge the output value \((i, j)\) in \( S \). The shape of regions \(L(i, j)\) depends only on four parameters: two slopes and two intersection points, instead of the six apparent free parameters in equation (1). Knowing for example the slopes of the external lines \(m_0 = -b_0/b_\pm, m_1 = -b_0/b_{\pm}\), the slope of the line
Figure 1: Convergence maps in the \((u_0', u_1')\)-plane and in the \((u_0, u_1)\)-plane for \(p > 0\).

connecting them is already determined and, knowing the two intersection points which depend on \(m_0, m_1, I\) and \(p\), we complete the construction of the convergence map. For instance, on Table 1 we can see the line equations defining the boundaries of the different convergence maps for a positive value of parameter \(p\).

<table>
<thead>
<tr>
<th>((u_0, u_1)) plane</th>
<th>((u_0', u_1')) plane</th>
</tr>
</thead>
<tbody>
<tr>
<td>((x_p, y_p) = \frac{(m_0(1 - y_p))}{(x_p)})</td>
<td>((-p, p))</td>
</tr>
<tr>
<td>((x_p, y_p) = \frac{(m(m_0 - x_p))}{((m_0 - y_p))})</td>
<td>((p, -p))</td>
</tr>
<tr>
<td>((u_1 - y_p) = m_1(u_0 - x_p))</td>
<td>(u_1' = p)</td>
</tr>
<tr>
<td>((u_1 - y_p) = m_0(u_0 - x_p))</td>
<td>(u_0' = -p)</td>
</tr>
<tr>
<td>((u_1 - y_p) = m_1(u_0 - x_p))</td>
<td>(u_1' = -p)</td>
</tr>
<tr>
<td>((u_1 - y_p) = m_0(u_0 - x_p))</td>
<td>(u_0' = p)</td>
</tr>
<tr>
<td>((u_1 - y_p) = m_1(u_0 - x_p))</td>
<td>(u_1' + u_0' = 0)</td>
</tr>
</tbody>
</table>

Table 1: Intersection points and boundary lines of the convergence regions for \(p > 0\).

From all this study, we also note that we can make the system converge where we want if we compose different templates. So for example, composing two templates, the first one will drive the system to one of the four points \(y_i \in S\). Using this points as the external inputs \(u_i = y_i \in S\) for the second template, we construct another convergence map to make each of this new inputs correspond to a new output.

### 3. Fixing the CNN parameters

The particular shape of the convergence map seems to limit the kind of problems which can be solved using a two neuron CNN. In order to determine whether this limitation is apparent or real we shall examine the repeated action of the cloning template. For this we can restrict the input choice \((u_0, u_1)\) to \(\pm 1\) without loss of generality. Using the B-transformation (2) we shall find their images \(B(\pm 1, \pm 1)\), and the necessary parameter conditions to place them into a pre-established convergence region in the \((u_0', u_1')\)-plane (Figure 1). This may done by studying if each one of the image points \(B(i, j), i, j = \pm 1\), are equal to one of the four possible outputs \(S\) located in each of the four different convergence regions. If the output points are located on a boundary line dividing different convergence regions, we shall translate the input point \((i, j) + (\varepsilon, \varepsilon), i, j = -1, 1, \varepsilon \neq 0\), and proceed as we have explained before. To simplify the notation, let’s rename \(S\)-points with the correspondence shown in Table 2, and the convergence regions \(L(i, j)\) will then be \(R(k)\), for \(k = 1, 2, 3, 4\).

<table>
<thead>
<tr>
<th>((1, 1))</th>
<th>((1, -1))</th>
<th>((-1, -1))</th>
<th>((-1, 1))</th>
</tr>
</thead>
<tbody>
<tr>
<td>((1, 1))</td>
<td>((1, -1))</td>
<td>((-1, -1))</td>
<td>((-1, 1))</td>
</tr>
</tbody>
</table>

Table 2: \(S\)-points correspondence.

For example, let us study the convergence of input 1 for a positive parameter \(p\). If we take 1 converging to itself, \(B(1) = (1, 1)\), this condition implies that for \(p > 0\), \(B\)-parameters fulfill:

\[
\{b_0 + b_+ + I = 1, \quad -b_+ - b_0 + I = 1\}. \quad (4)
\]

Next we consider the different outputs where input 3 can converge. If \(B(3) = (-1, 1)\), we have

\[
\{-b_0 - b_+ + I = -1, \quad -b_- - b_0 + I = 1\}. \quad (5)
\]

Solving the system equations (4) and (5), parameter \(I\) must be equal to 0 and 1. Therefore, such an association can not be achieved by one single 2-neuron CNN. However, if \(B(3) = (1, 1)\), we have

\[
\{-b_0 - b_+ + I = 1, \quad -b_- - b_0 + I = 1\}. \quad (6)
\]

Solving the system equations (4) and (6), we find parameters \(I = 1, b_+ = b_- = b_0\). This relation is compatible with a 2-neuron CNN. Now we study the four possible outputs for inputs 2 and 4 where \(B(2) = (1 - 2b_0, 1 + 2b_0)\) and \(B(4) = (1 + 2b_0, 1 - 2b_0)\), summarized in Table 3.

<table>
<thead>
<tr>
<th>((1, -1))</th>
<th>((-1, 1))</th>
<th>parameter conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>(R(1))</td>
<td>(R(1))</td>
<td>(p &gt; \max(-1 - 2b_0, -1 + 2b_0))</td>
</tr>
<tr>
<td>(R(2))</td>
<td>(R(4))</td>
<td>(p &lt; -1 + 2b_0)</td>
</tr>
<tr>
<td>(R(3))</td>
<td>(R(3))</td>
<td>(\times)</td>
</tr>
<tr>
<td>(R(4))</td>
<td>(R(2))</td>
<td>(p &lt; -1 - 2b_0)</td>
</tr>
</tbody>
</table>

Table 3: Convergence study for points \((-1, 1)\) and \((1, -1)\).

Let us note that the parameter conditions are incompatible for certain values of \(b_0\). Using the parameter conditions shown in Table 3, we obtain that input point 2 can converge to outputs \((1, 1), (1, -1)\) depending on parameters \(p, b_0\) (7). Otherwise for \(0 < b_0 < 1/2\) and \(p > 0\), input point 2 can only converge to \((1, 1)\).

\[
\begin{align*}
B(2) &= (1, 1) \quad \Leftrightarrow \quad p > -1 + 2b_0, b_0 > 1/2, \\
B(2) &= (1, -1) \quad \Leftrightarrow \quad p < -1 + 2b_0, b_0 > 1/2.
\end{align*}
\quad (7)
\]

Like in the first case, input 3 can not converge to \((-1, -1)\) because there is no solution for the system equations obtained from (4) and \(B(3) = (1, -1)\). Finally, input 3 converges to itself for parameters, \(I = 0\) and \(b_+ = b_- = 1 - b_0\). The other square point \(B\)-images are \(B(2) = (-1 + 2b_0, 1 - 2b_0)\), \(B(4) = (1 + 2b_0, 1 - 2b_0)\), \(B(2) = (1 - 2b_0, 1 + 2b_0)\), and \(B(4) = (-1 - 2b_0, -1 + 2b_0)\) for \(b_0 < 1/2\).
$2b_0 = -B(4)$ and lay on a boundary line of the convergence map. In this case, we apply a translation to the image points in order to solve $B(\pm 1, \pm 1) = (\pm 1 + \varepsilon, \pm 1 + \varepsilon), \varepsilon \in \mathbb{R} - \{0\}$. 

From the equations obtained, we find parameters $I = \varepsilon, b_s = b_\varepsilon = 1 - b_0$. The other square point B-images are then $B(2) = (-1 + 2b_0 + \varepsilon, 1 - 2b_0 + \varepsilon)$ and $B(4) = (1 - 2b_0 + \varepsilon, -1 + 2b_0 + \varepsilon)$. Using the convergence map, $B(2) = (1, 1)$ if and only if conditions in (8) are fulfilled:

$$\begin{cases} 
\pm(-1 + 2b_0) + \varepsilon > -p \\
1 - 2b_0 + \varepsilon \geq 1 - 2b_0 - \varepsilon \Rightarrow \varepsilon \geq 0 
\end{cases}$$

(8)

Parameter conditions are then $p > \max\{\pm(-1 + 2b_0) - \varepsilon\}$. Doing a similar study for the other convergence regions we obtain the rest.

Using similar arguments we find all possible output values for the case where the first input point converges to itself and $p > 0$ in Table 4. We use the two row notation in order to describe the rearrangement of the input-output relations obtained.

<table>
<thead>
<tr>
<th>input-output</th>
<th>parameter conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 2 3 4</td>
<td>$p &gt; \max{-1 - 2b_0, -1 + 2b_0}$</td>
</tr>
<tr>
<td>1 1 1 1</td>
<td>$0 &lt; p &lt; -1 + 2b_0$</td>
</tr>
<tr>
<td>1 2 3 4</td>
<td>$0 &lt; p &lt; -1 - 2b_0$</td>
</tr>
<tr>
<td>1 4 1 2</td>
<td>$0 &lt; p &lt; \min{-1 + 2b_0 \pm \varepsilon}$</td>
</tr>
<tr>
<td>1 2 3 4</td>
<td>$0 &lt; p &lt; \min{-1 - 2b_0 \pm \varepsilon}$</td>
</tr>
<tr>
<td>1 2 3 4</td>
<td>$p &gt; \max{\pm(-1 + 2b_0) - \varepsilon}, \varepsilon &gt; 0$</td>
</tr>
<tr>
<td>1 1 3 1</td>
<td>$p &gt; \max{\pm(-1 + 2b_0) - \varepsilon}, \varepsilon &lt; 0$</td>
</tr>
</tbody>
</table>

Table 4: Possible outputs for the case $B(1) = (1, 1), p > 0$.

Now, from the study of all the direct input-output relations, we have found 25 possible convergence options with their correspondent templates. Let’s note that the parameter conditions in order to reproduce a desired input-output relation, are determined by $p$ and $b_0$. The rest, $b_s$ and $b_\varepsilon$, depend in each particular case, on $b_0$. From the different system equations, parameter $I$ gives us the key point in order to discuss the existence of a solution.

Moreover, composing this different templates, we obtain all the possible relation between the four points in $S$ using a two neuron CNN. This relation can be classified with those converging to one, two, three or four different outputs.

For example, there are four elements $T_i, i = 1, \ldots, 4$ converging to a single output. A template $T_1$ making the system converge to output 1, can be defined for parameters $I = 1, b_s = b_\varepsilon = -b_0, p > \max\{-1 \pm 2b_0\}$ and $s > 1$. Choosing $b_0 = 2, p = 4 > 3$ and $s = 3$ we find $(s, p, b_0, b_\varepsilon, b_\varepsilon, I) = (3, 4, 2, -2, -2, 1)$. Composing this template with another one $T_j$ where input 1 converges to 2 we obtain $T_2 = T_j \circ T_1$.

$$T_2 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 2 & 2 & 2 \end{pmatrix}$$

Let’s note that using these results, a two neuron CNN can also realize Boolean functions $F : \mathbb{R}^2 \rightarrow \mathbb{R}$ defined for example as $F(u_0, u_1) = y_0(\infty), (u_0, u_1) \in S$ like in [6]. Using a single template, linearly separable Boolean functions can be solved while for the rest, template composition must be used.

4. Impossible relations

In the remaining of the paper we shall focus on the only but bijective input-output relations summarized in Table 5. From $p_1$ to $p_8$, they are obtained by the action of a single template except $p_3$ and $p_8$ which come from the composition of two templates.

<table>
<thead>
<tr>
<th>inputs</th>
<th>$p_1$</th>
<th>$p_2$</th>
<th>$p_3$</th>
<th>$p_4$</th>
<th>$p_5$</th>
<th>$p_6$</th>
<th>$p_7$</th>
<th>$p_8$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 5: Cases where the CNN converges to four different outputs.

In this particular case, $p_1$ can be written as permutations of four different objects: the input points in $S$.

Remark that we have found only eight bijective relations, while using four elements $S$, we should find the set of all possible permutations, the symmetric group $S_4$ of $4! = 24$ elements. To shed light in the number of different templates which perform a functional relation between all the four elements, we compose the eight ones described in Table 5.

Let’s first rewrite the eight permutation templates $p_i$ using the cycle notation, and compose them.

$$p_1 = Id \quad p_3 = (12)(34) \quad p_5 = (13) \quad p_7 = (1432)$$

$$p_2 = (24) \quad p_4 = (1234) \quad p_6 = (13)(24) \quad p_8 = (14)(23)$$

(9)

The result of all the composition templates represented by product permutations is shown in Table 6.

Let’s note that we have found a special subset of group $S_4$ that fulfill the group properties, this is a subgroup.

With this results we set a Convergence Lemma. Let’s consider a two neuron CNN defined by equations (1) where parameters fulfill $s > 1$ and $p_s = p_\varepsilon = p$. Let’s name $S = \{(\pm 1, \pm 1)\}$ the four possible output values set where the CNN can converge. There exist only eight different cases where the CNN system converges to the four different outputs $S$ summarized in Table 5.
One example where we can see this restrictions is trying to implement Minsky’s 7-state 4-color universal Turing machine illustrated in Figure 2.

We want the two neuron CNN to represent the header action of the Turing machine on the tape so we shall use the values ±1 to code the colors. Since a symmetric two neuron CNN has four possible outputs for s > 1, it can be used to modify the color of an active cell of the Minsky’s universal Turing machine. The input color will be coded on the u_i’s while the output color will be obtained from the final state y_i’s of the neurons (Figure 3).

In this way, each state of the machine corresponds to a template or to a combination of templates relating the four possible input symbols to their correspondent output ones. To design these templates we shall use the convergence map studied before. Taking the fifth state (Figure 2) which can be written as s_5 = (34), we see from the convergence lemma that this case can not be performed using a two neuron CNN because it is not one of the eight possible permutations found (9). Of course, this result seem to depend on the choice between colors and S-points. State s_5 is the only permutation of colors so one may think on choosing a different relation between S-points and the Turing machine colors to implement the state. However, in all the cases, the associations fulfilling s_i, do no fulfill some other state s_i, i ≠ 1, . . . , 7, where the system converges to three different outputs. So, a 2-neuron CNN can not be used to reproduce this header action of the universal Turing machine.

5. Conclusions

We have seen that a 2-neuron CNN can not perform all logical bijective function but only a subgroup. The particular geometry of the convergence map limits the kind of problems which can be solved yet, allows to classify them into those converging to one, two, three or four different outputs. In this way, we may know if a specific problem, can be solved using a two neuron CNN. In the line of [6], we have checked that all single-output Boolean functions can be reproduced. Linearly separable ones, just need one template while non linearly separable ones require the composition of two. We have also seen as an example the problem to reproduce the header action of the universal Turing machine.

From [5] we know that a CNN is a universal Turing machine in higher dimensions. So this leads to the discussion on which is the minimal CNN being a universal Turing machine. Moreover, using y_0(∞) as output, we can reproduce any Boolean function just like a universal CNN cell but, using (y_0(∞), y_1(∞)) as output we loose universality.

Acknowledgments

This work is financially supported by FUNITEC.

References


Table 6: Template composition for all the permutations p_i founded in a two neuron CNN.

<table>
<thead>
<tr>
<th></th>
<th>p1</th>
<th>p2</th>
<th>p3</th>
<th>p4</th>
<th>p5</th>
<th>p6</th>
<th>p7</th>
<th>p8</th>
</tr>
</thead>
<tbody>
<tr>
<td>p1</td>
<td>p1</td>
<td>p2</td>
<td>p3</td>
<td>p4</td>
<td>p5</td>
<td>p6</td>
<td>p7</td>
<td>p8</td>
</tr>
<tr>
<td>p2</td>
<td>p2</td>
<td>p1</td>
<td>p7</td>
<td>p8</td>
<td>p6</td>
<td>p5</td>
<td>p3</td>
<td>p4</td>
</tr>
<tr>
<td>p3</td>
<td>p3</td>
<td>p4</td>
<td>p1</td>
<td>p2</td>
<td>p7</td>
<td>p8</td>
<td>p5</td>
<td>p6</td>
</tr>
<tr>
<td>p4</td>
<td>p4</td>
<td>p3</td>
<td>p5</td>
<td>p6</td>
<td>p8</td>
<td>p7</td>
<td>p1</td>
<td>p2</td>
</tr>
<tr>
<td>p5</td>
<td>p5</td>
<td>p6</td>
<td>p4</td>
<td>p3</td>
<td>p1</td>
<td>p2</td>
<td>p8</td>
<td>p7</td>
</tr>
<tr>
<td>p6</td>
<td>p6</td>
<td>p5</td>
<td>p8</td>
<td>p4</td>
<td>p3</td>
<td>p1</td>
<td>p2</td>
<td>p7</td>
</tr>
<tr>
<td>p7</td>
<td>p7</td>
<td>p8</td>
<td>p2</td>
<td>p1</td>
<td>p4</td>
<td>p6</td>
<td>p5</td>
<td>p3</td>
</tr>
<tr>
<td>p8</td>
<td>p8</td>
<td>p7</td>
<td>p6</td>
<td>p5</td>
<td>p4</td>
<td>p3</td>
<td>p2</td>
<td>p1</td>
</tr>
</tbody>
</table>

Figure 2: Generalization of Minsky’s 7-state 4-color universal Turing machine made by Macura [4].

Figure 3: One correspondence between the Turing machine colors and the four possible states of the 2-neuron CNN.

\[ (-1,1) \quad (1,1) \quad (-1,-1) \quad (1,-1) \]
Generation Method of Extremely Ill-conditioned Integer Matrices

Tetsuo Nishi†, Siegfried Rump‡ and Shin’ichi Oishi§

†School of Science and Engineering, Waseda University
2-4-12 Okubo, Shinjuku-ku, Tokyo, 169-0072 Japan
‡Hamburg University of Technology
21071 Hamburg, Germany
Email: nishi-t@waseda.jp, rump@tu-harburg.de, oishi@waseda.jp

Abstract—This paper proposes an innovative generation method of extremely ill-conditioned integer matrices. This method is superior to the conventional Rump’s method, i.e., the former has a simpler algorithm and can generate more variety of ill-conditioned matrices than the latter. ∴

1. Introduction

Extremely ill-conditioned matrices are required to examine the quality of accuracy-guaranteed algorithms for solving linear simultaneous equations[1]–[6]. Here an ill-conditioned matrix implies that its condition number is $10^{16} \sim 10^{100}$ or larger in the double precision arithmetic.

Once S. Rump[7] proposed a method to generate extremely ill-conditioned matrices. His method utilized the Pell equation, which is well-known in the number theory[12]. The method is most well-known and is used as a standard tool to generate an ill-conditioned matrix with an arbitrary condition number in the INTLAB, but the variety of generated matrices is not so large because the number of solutions of the Pell equation is not so many. So we want other methods to obtain more variety of matrices. From this point of view we proposed [8][10][11] several methods which are considered as extensions of Rump’s method[7].

In this paper we propose another method to generate ill-conditioned matrices. It has the following features in comparison with Rump’s method[7] and its extensions [8][10][11]: (i) it has a simpler algorithm, (ii) the obtainable condition number is roughly the same as previous ones, (iii) it generates much variety of matrices. The obtained matrices are somewhat similar to the companion matrix.

2. Preliminaries

Let $\mu$ be a large positive integer such as $10^5$, $10^{10}$ or $2^{53}$ (but $\mu = 10$ or $\mu = 2$ may also be permissible theoretically) and let an $n \times n$ integer matrix $A = [a_{ij}]$ to be determined satisfy $|a_{ij}| \leq \mu$. It is very probable that the maximum condition number of $A$ is large, as $\mu$ is large. Our purpose is to generate an integer matrix $A = [a_{ij}]$ such that $|a_{ij}| \leq \mu$ and $\text{Cond}(A) = ||A|| ||A^{-1}||$ is extremely large.

2.1. Outline of Rump’s method

One of the key points of Rump’s method is to find a $2 \times 2$ integer matrix $V^1$ s.t.

$$V = \begin{bmatrix} P & kQ \\ Q & P \end{bmatrix}, \quad |V| = \left| \begin{array}{cc} P & kQ \\ Q & P \end{array} \right| = 1 \quad (1)$$

The condition $|V| = 1$ is important for an ill-conditioned matrix. The integers $P$ and $Q$ are extremely large such as $10^{50}$ and are chosen so as to satisfy the Pell equation:

$$P^2 - kQ^2 = 1 \quad (2)$$

Thus $|V| = 1$ in Eq.(1) is satisfied. Then utilizing $V$ in Eq.(1), he proposed a $(2n + 2) \times (2n + 2)$ integer matrix $A$ and showed by rather tricky calculations that $A$ satisfies

$$\text{Cond}_{\infty}(A) = ||A||_{\infty} ||A^{-1}||_{\infty} \geq (P + Q)(P + kQ) \sim 4\mu^{2n+1} \quad (3)$$

The last term $(4\mu^{2n+1})$ is obtained under certain reasonable assumptions. Since $A$ is a $(2n + 1) \times (2n + 1)$ matrix, we see that the condition number in Eq.(3) per degree is approximately $\mu$, i.e., $|\text{Cond}_{\infty}(A)|^{1/(2n+1)}$ $\sim \mu$.

2.2. Previous extensions of Rump’s method

We showed[8][10][11] that Rump’s algorithm can easily be generalized by replacing $V$ in Eq.(1) with the following two kinds of matrices.

2.2.1 Replacing $V$ by more general type of a $2 \times 2$ matrix

The matrix $V$ was generalized as:

$$V' = \begin{bmatrix} P & F \\ Q & G \end{bmatrix}, \quad |V'| = PG - QF = 1 \quad (4)$$

Prescribed $P$ and $Q$ having no common factor, e.g.,

$$P = 2^k, \quad Q = 3^n \quad P = 2^{k1}3^{k2}11^{k3}, \quad Q = 2^{m1}7^{m2}$$

we can find $F$ and $G$ satisfying Eq.(4) by using the Euclid algorithm[13].

2.2.2 Replacing $V$ by a $3 \times 3$ matrix

3. Generation of ill-conditioned matrices similar to the companion matrix

3.1. Generation method

In this section we consider the generation of an ill-conditioned matrix, which is similar to a companion ma-
trix. Let $A$ be an $n \times n$ integer matrix such that

$$A = \begin{bmatrix}
a_1 & a_2 & a_3 & a_4 & \cdots & a_{n-1} & a_n \\
1 & -\sigma_1 & 0 & 0 & \cdots & 0 & 0 \\
0 & 1 & -\sigma_2 & 0 & \cdots & 0 & 0 \\
0 & 0 & 1 & -\sigma_3 & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & 0 & \cdots & 1 & -\sigma_{n-1} \\
0 & 0 & 0 & 0 & \cdots & 0 & 1
\end{bmatrix}$$

(5)

Without loss of generality we can assume

$$0 < \sigma_i < \mu \quad (i = 1, 2, \cdots, n-1) \quad (6)$$

$$|a_i| < \mu \quad (i = 1, 2, \cdots, n) \quad (7)$$

In this paper we determine $a_i$ ($i = 1, \cdots, n$) such that

$$((a_1\sigma_1 + a_2)\sigma_2 + a_3)\sigma_3 + \cdots \sigma_{n-1} + a_n = 1$$

(8)

Eq.(8) corresponds to $|V| = 1$ in Eq.(1). From the above we see that $a_i$ ($i = 1, 2, \cdots, n$) necessarily include both positive and negative values. Referring to Eq.(8), we will describe how we determine $a_i$.

### 3.2. Determination of $a_i$

We determine $a_i$ by the following three steps:

**Step 1:** From Eq.(8) we have

$$1 - a_n \equiv 0 \pmod{\sigma_{n-1}}$$

(9)

from which we have

$$\frac{1 - a_n}{\sigma_{n-1}} =: k_{n-1}, \quad (k_{n-1} = 0, \pm 1, \pm 2, \pm 3, \cdots)$$

(10)

Thus we have

$$a_n = 1 - \sigma_{n-1} k_{n-1}$$

(11)

Therefore $k_{n-1}$ has to satisfy from Eq.(7) the following equation:

$$|a_i| = |1 - \sigma_{n-1} k_{n-1}| < \mu \quad (i = 1, 2, \cdots, n-1)$$

(12)

i.e., $-\mu < 1 - \sigma_{n-1} k_{n-1} < \mu$. This can be rewritten as

$$1 + \mu > \sigma_{n-1} k_{n-1} > -\mu + 1$$

(13)

from which we have

$$\frac{1 + \mu}{\sigma_{n-1}} > k_{n-1} > \frac{-\mu + 1}{\sigma_{n-1}}$$

(14)

From the above and Eq.(14) we see that

$$k_{n-1} = \frac{1 + \mu}{2\sigma_{n-1}} \quad (> 0) \quad \text{or} \quad \frac{1 - \mu}{2\sigma_{n-1}} \quad (< 0)$$

(15)

are appropriate candidate of $k_{n-1}$, but $k_{n-1}$ is not limited to Eq.(15). We have $a_n$ from Eq.(11) by Eq.(11).

**Step 2:** Quite similarly we can derive equations corresponding to Eqs.(9)-(15) successively. We take $j = 1, 2, \cdots, n-1$ in the order. Then we have $k_{n-2}, k_{n-3}, \cdots, k_1, a_{n-1}, a_{n-2}, \cdots, a_2$ as follows:

$$k_{n-j} - a_{n-j} \equiv 0 \pmod{\sigma_{n-j-1}}$$

(16)

$$k_{n-j} - a_{n-j} = k_{n-j-1}, \quad (k_{n-j-1} = 0, \pm 1, \pm 2, \cdots)$$

(17)

$$a_{n-j} = a_{n-j} - k_{n-j-1} k_{n-j-1}$$

(18)

$$|a_{n-j}| = |k_{n-j} - \sigma_{n-j-1} k_{n-j-1}| < \mu$$

(19)

$$k_{n-j} + \mu > \sigma_{n-j-1} \quad (> 0) \quad \text{or} \quad -\mu + k_{n-j} \quad (< 0)$$

(20)

We therefore have:

$$k_{n-j-1} = \left[ \frac{k_{n-j} + \mu}{2\sigma_{n-j-1}} \right] \quad (> 0) \quad \text{or} \quad \left[ -\frac{\mu + k_{n-j}}{2\sigma_{n-j-1}} \right] \quad (< 0)$$

(21)

are appropriate candidates of $k_{n-j}$.

**Step 3:**

$$a_{n-j} \equiv k_{j-1}$$

(22)

Since $a_i$ have to satisfy Eq.(8), we assume

**Assumption 1:** We choose $a_i$ as

$$a_{2j} > 0, \quad a_{2j+1} < 0 \quad \cdots \quad k_{2j} < 0, \quad k_{2j+1} > 0 \quad (j = 1, 2, \cdots)$$

(23)

or

$$a_{2j} < 0, \quad a_{2j+1} > 0 \quad \cdots \quad k_{2j} > 0, \quad k_{2j+1} < 0 \quad (j = 1, 2, \cdots)$$

(24)

For convenience let

$$k_n = 1$$

(23)

Then we can calculate $a_j$ and $k_j$ recursively for $j = n-1, n-2, \cdots, 2$ in this order by both Eq.(24) and Assumption 1.

$$\frac{k_{j+1} + \mu}{\sigma_j} > k_j > \frac{-\mu + k_{j+1}}{\sigma_j}$$

(24)

$$k_j = \left[ \frac{k_{j+1} + \mu}{2\sigma_j} \right] \quad (> 0) \quad \text{or} \quad \left[ \frac{k_{j+1} - \mu}{2\sigma_j} \right] \quad (< 0)$$

(25)

is reasonable candidates of $k_j$.

$$a_{j+1} = k_{j+1} - \sigma_j k_j$$

(26)

Eq.(25) is only an example of choice and we can also choose many other values.

### 3.3. Condition number of $A$ in Eq.(5)

In this section we calculate the condition number of $A$ in Eq.(5) with the $\infty$-norm. For this purpose we need to calculate the inverse matrix $A^{-1}$, which can be easily calculated in a similar way as in [8] as follows:

Let

$$H = \begin{bmatrix}
1 & 0 & 0 & 0 & \cdots & 0 & \prod_{j=1}^{n-1} \sigma_j \\
0 & 1 & 0 & 0 & \cdots & 0 & \prod_{j=1}^{n-1} \sigma_j \\
0 & 0 & 1 & 0 & \cdots & 0 & \prod_{j=1}^{n-1} \sigma_j \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & 0 & \cdots & 1 & \sigma_{n-2} \sigma_{n-1} \\
0 & 0 & 0 & 0 & \cdots & 0 & \sigma_{n-1} \\
0 & 0 & 0 & 0 & \cdots & 0 & 1
\end{bmatrix}$$

(27)

Then we have

$$A' = AH = \begin{bmatrix}
a_1 & a_2 & a_3 & a_4 & \cdots & a_{n-2} & a_{n-1} & a_n \\
1 & -\sigma_1 & 0 & 0 & \cdots & 0 & 0 & 0 \\
0 & 1 & -\sigma_2 & 0 & \cdots & 0 & 0 & 0 \\
0 & 0 & 1 & -\sigma_3 & \cdots & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & \cdots & 1 & -\sigma_{n-2} & 0 \\
0 & 0 & 0 & 0 & \cdots & 0 & 1 & -\sigma_{n-1}
\end{bmatrix} \times \begin{bmatrix}
1 & 0 & 0 & \cdots & 0 & \prod_{j=1}^{n-1} \sigma_j \\
0 & 1 & 0 & \cdots & 0 & \prod_{j=1}^{n-1} \sigma_j \\
0 & 0 & 1 & \cdots & 0 & \prod_{j=1}^{n-1} \sigma_j \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & 1 & \sigma_{n-2} \sigma_{n-1} \\
0 & 0 & 0 & \cdots & 0 & \sigma_{n-1} \\
0 & 0 & 0 & \cdots & 0 & 1
\end{bmatrix}$$

(27)
Let

\[
A' = \begin{bmatrix} U \end{bmatrix}
\]

\[
U = \begin{bmatrix} a_1 & a_2 & a_3 & \cdots & a_{n-1} \\
1 & -\sigma_1 & 0 & 0 & \cdots & 0 \\
0 & 1 & -\sigma_2 & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & 1 & -\sigma_{n-2} \\
0 & 0 & 0 & \cdots & 0 & 1
\end{bmatrix}
\]

\[
W = \begin{bmatrix} W^{-1} \\
1 & -UW^{-1}
\end{bmatrix}
\]

\[
(A')^{-1} = \begin{bmatrix} 0 \\
W^{-1} \\
1 & -UW^{-1}
\end{bmatrix}
\]

\[
W^{-1} = \begin{bmatrix} 1 & \sigma_1 & \sigma_1 \sigma_2 & \cdots & \sigma_1 \sigma_2 \cdots \sigma_1 \sigma_2 \cdots \\
0 & 1 & \sigma_2 & \sigma_2 \sigma_3 & \cdots & \sigma_2 \sigma_3 \cdots \\
0 & 0 & 1 & \sigma_3 & \cdots & \sigma_3 \cdots \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & 1 & \sigma_{n-2} \\
0 & 0 & 0 & \cdots & 0 & 1
\end{bmatrix}
\]

\[
-\text{UW}^{-1} = \begin{bmatrix} -[ a_1 & a_2 & a_3 & \cdots & a_{n-1}] \\
1 & \sigma_1 & \sigma_1 \sigma_2 & \cdots & \sigma_1 \sigma_2 \cdots \\
0 & 1 & \sigma_2 & \sigma_2 \sigma_3 & \cdots & \sigma_2 \sigma_3 \cdots \\
0 & 0 & 1 & \sigma_3 & \cdots & \sigma_3 \cdots \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & 1 & \sigma_{n-2} \\
0 & 0 & 0 & \cdots & 0 & 1
\end{bmatrix}
\]

We therefore have

\[
\|A^{-1}\|_{\infty} > \max \left\{ \prod_{i=1}^{n} \sigma_i, \prod_{i=1}^{n} \sigma_i, \prod_{i=1}^{n} \sigma_i \right\}
\]

(29)

Finally we have

\[
\text{Cond}_{\infty}(A) > \left( \prod_{i=1}^{n} \sigma_i \right) \sum_{i=1}^{n} |a_i|
\]

This corresponds to the Rump’s result in Eq.(3). If we choose

\[
|a_i| \sim \mu, \quad \nu \sim \sqrt{\nu}
\]

then we see that \(K_{n-1} \sim \mu^{n-1}\). We therefore see from Eqs.(31) and (32) that

\[
\text{Cond}_{\infty}(A) > \mu^{n-1} \cdot \mu^{\frac{n}{2}} \cdot (n-1) \mu \sim \mu^{n+\frac{1}{2}}
\]

(33)

Since the size of A is n, the condition number per degree is approximately \(\mu\), that is, Eq.(31) is appropriately equal to that in [7].

3.4. Considerations through examples

Example 1: Let

\[
\mu = 10, \quad n = 4, \quad \sigma_1 = \sigma_2 = \sigma_3 = 5
\]

(34)

We choose \(k_i, a_i\) using Eqs.(9)–(22).

Since \(1 - a_4\) must be divided by 5, we have

\[
a_4 = 1, -4, 6, -9, \cdots
\]

So we choose \(a_4 = -9\) as an example. Then \(k_3 = (1 - a_4)/5 = 2\).

Since \(k_3 - a_3\) must be divided by 5, we choose \(a_3 = 7\) and therefore \(k_2 = 1\). Since \((k_2 - a_2) = 0\) and \(k_1 = a_1 = 1\). Then \((1 \times 5 + (-6) \times 5 + 5 \times 5 + (-9)) = 1\) surely holds.

We therefore have

\[
A = \begin{bmatrix} 1 & 6 & 7 & -9 \\
1 & -5 & 0 & 0 \\
0 & 1 & -5 & 0 \\
0 & 0 & 1 & -5
\end{bmatrix}
\]

(35)

\[
A^{-1} = \begin{bmatrix} 125 & -124 & 130 & -225 \\
25 & -25 & 26 & -45 \\
5 & -5 & 5 & -9 \\
1 & -1 & 1 & -2
\end{bmatrix}
\]

(36)
from which we see that

$$\text{Cond}_2(A) = ||A|| \cdot ||A^{-1}|| = 13892 \quad (37)$$

We have $|A| = -1$ and the singular values of $A$ are about 14.1, 5.14, 4.42, and 0.00312. Therefore we have

$$\text{Cond}_2(A) = \frac{14.109418}{0.0031213} \approx 4520.2995$$

**Remark 1:** From the above results, we see that (i) $A$ has a considerably large condition number even for such small $\mu$ and $n$, (ii) $A$ has three large singular values and an extremely small one, and (iii) $A^{-1}$ is very near to a matrix with rank one.

**Example 2:** Let

$$\mu = 1000, \quad n = 4, \quad \sigma_1 = \sigma_2 = \sigma_3 = 50 \quad (38)$$

In a similar way as in Example 1 we choose $k_i$ and $a_i$ using Eqs.(9)–(22).

Since $1 - a_2$ must be divided by 50, we choose $a_2 = -799$ and $k_1 = (1 - a_4)/50 = 16$.

Since $k_3 - a_3$ must be divided by 50, we choose as $a_3 = 716$ and therefore $k_2 = -14$. Since $(k_3 - a_2)$ must be divided by 50, we choose as $a_2 = -864$ and $k_1 = a_1 = 17$. Then ($\binom{17 + 50 + (-864)}{50 + 716} \times 50 + 716 \times 50 + (-799) = 1$ holds.

We therefore have

$$A = \begin{bmatrix} 17 & -864 & 716 & -799 \\ 1 & -50 & 0 & 0 \\ 0 & 1 & -50 & 0 \\ 0 & 0 & 1 & -50 \end{bmatrix}$$

$$A^{-1} = \begin{bmatrix} 125000 & -2124999 & 1750050 & -1997500 \\ 2500 & -42500 & 35001 & -39950 \\ 50 & -850 & 700 & -799 \\ 1 & -17 & 14 & -16 \end{bmatrix}$$

from which we see that

$$\text{Cond}_2(A) = ||A|| \cdot ||A^{-1}|| \approx 13 \times 10^8 \quad (39)$$

We have $|A| = -1$, and singular values are about 1380, 50.0, 49.4 and 0.0000003. Then we have

$$\text{Cond}_2(A) = \frac{1378.5521}{0.0000003} \approx 4.693 \cdot 10^9$$

Thus $A$ has a considerably large condition number for $\mu = 1000$ and $n = 4$ and a similar remark as Remark 1 in Example 1 holds.

We see from the above examples that we do not obtain uniform singular values distribution. The reason is omitted here due to the lack of space. As a trial to obtain more uniform singular value distribution, we can choose $\sigma_1 = 1$ but the result is not necessarily good. The desirable singular value distribution will be discussed in near future.

### 4. Conclusion

This paper proposed an innovative generation method of extremely ill-conditioned integer matrices. This method is superior to the conventional Rump’s method in some respects.

### Acknowledgments

This research was supported in part by the Ministry of Education, Science, Culture, Sports, Science and Technology, Grant-in-Aid for Specially Promoted Research, no. 17002012, 2005-2010, on “Establishment of Verified Numerical Computation” and by the Grant-in-Aid for Scientific Research (C) (No. 20560374, 2008-2010) of the Ministry of Education, Science, Culture, Sports, Science and Technology of Japan.

### References

Performance of Adiabatic Quantum Computation using Neuron-like Interconnections

Shigeo Sato †, Aiko Ono †, Mitsunaga Kinjo †, and Koji Nakajima †

†Research Institute of Electrical Communication, Tohoku University
2-1-1 Katahira, Aoba-ku, Sendai, 980-8577 Japan
‡Department of Electorical and Electronic Engineering, University of the Ryukyus
1 Senbaru, Saibaru-cho, Okinawa, 903-01 Japan
Email: shigeo@riei.tohoku.ac.jp

Abstract—Quantum computation algorithms indicate possibility that non-deterministic polynomial time problems can be solved much faster than classical methods. Farhi et al. have proposed an adiabatic quantum computation (AQC) for solving the three-satisfiability (3-SAT) problem. We have proposed a neuromorphic quantum computation algorithm based on AQC, in which an analogy to an artificial neural network (ANN) is considered in order to design a Hamiltonian. However, in the neuromorphic AQC, the relation between its computation time and success rate has not been clear. In this paper, we study residual energy and the probability of correct answers as a function of computation time. The residual energy behaves as expected from the adiabatic theorem. On the other hand, the success rate strongly depends on energy level crossings of excited states during Hamiltonian evolution. The results indicate that computation time must be adjusted according to a target problem.

2. Adiabatic Quantum Computation with Neuron-like Interconnections

2.1. Adiabatic Quantum Computation

The Hamiltonian of the adiabatic quantum computation changes as time goes on from an initial Hamiltonian \(H_I\) whose ground state is given as the superposition of all states, to a final Hamiltonian \(H_F\) whose ground state includes solutions for a given problem. (see [1], [8] for details.) The evolution of the Hamiltonian is given as

\[
H(t) = \left(1 - \frac{t}{T}\right)H_I + \frac{t}{T}H_F, \tag{1}
\]

\[
H_I = \sum_{i=0}^{2^n-1} \sigma_x^{(i)}, \tag{2}
\]

where \(n\) and \(\sigma_x^{(i)}\) are the number of qubits and the \(x\)-component of the Pauli spin matrix acting on the \(i\)-th qubit, respectively. \(T\) denotes the computation time in which the Hamiltonian evolves. The initial quantum state is set to the superposition of all states and is given as

\[
|\psi(0)\rangle = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |i\rangle, \tag{3}
\]

where \(|i\rangle \equiv |x_n\rangle \cdots |x_1\rangle |x_0\rangle \equiv |x_n \cdots x_1 x_0\rangle\) and each qubit \(|x_i\rangle\) takes \(|0\rangle\) or \(|1\rangle\), exclusively. We can control the speed of state changes to be suitable for finding
solutions. If a sufficiently large $T$ is chosen, the evolution becomes adiabatic. The adiabatic theorem [6] tells that a quantum state remains close to each ground state. Therefore, solutions can be found in the final state $|\psi(T)\rangle$. However, if there is any degeneracy or level crossing during the Hamiltonian evolution, the quantum state may not stay in a ground state. These behaviors depend on a Hamiltonian strongly.

2.2. Neurocomputing Method for Designing Final Hamiltonians

The method of designing a final Hamiltonian $H_F$ has not been known for a general case. First of all, we consider a Hopfield neural network (HNN) [10]. The energy function $E$ is defined for an HNN because it has symmetrical synaptic connections. It is given as

$$E = -\frac{1}{2} \sum_{i,j} w_{ij} o_i o_j,$$

where $o_i$ and $w_{ij}=w_{ji}$ are the output of the $i$-th neuron and the synaptic weight between the $i$-th and the $j$-th neurons, respectively. The network state changes to lower energy states as time goes on. It has been known that an HNN can be applied to a combinatorial optimization problem [10]. If we can obtain the synaptic weights by comparing the energy function with the cost or penalty function of an optimization problem, the HNN can work for solving such a problem. However, if the network is trapped at a local minimum, its final state does not correspond to a solution.

If we know a method to convert a synaptic weight matrix $W$ to a final Hamiltonian $H_F$, AQC can be applied to an optimization problem in which quantum dynamics must be used for a negative (positive) synaptic weight. The sum of the upper and lower Hamiltonians reflecting a synaptic matrix $W$ is employed as a final Hamiltonian of neuromorphic AQC. Please note that these Hamiltonians can be replaced by some suitable ones arbitrary. Both parameters $\lambda$ and $A$ are constants, and their magnitudes should reflect the value of a synaptic weight $w_{ij}$. In the followings, however, we do not pay much attention to the magnitude of there parameters because the synaptic weights in this paper take values 0 or -1 only. For example, suppose we have a synaptic weight matrix

$$W_1 = \begin{pmatrix} 0 & -1 & -1 & 1 \\ -1 & 0 & -1 & 0 \\ -1 & -1 & 0 & -1 \\ 1 & -1 & -1 & 0 \end{pmatrix}.$$  

This matrix represents one excitatory and five inhibitory interactions. Then a final Hamiltonian $H_{F1}$ is given according to the rules described in Table I as

$$H_{F1} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix}$$

where $\lambda$ and $A$ are arbitrary constants.

3. Simulation Results

3.1. Residual Energy

For discussing the performance of the neuromorphic AQC, we investigate residual energy $\Delta E$. It is defined as the energy difference between the nominal ground energy calculated from a final Hamiltonian and the actual energy after the state evolution, and given as

$$\Delta E = \langle \psi'(T) | H_F | \psi'(T) \rangle - \langle \psi(T) | H_F | \psi(T) \rangle,$$  

where $\langle \psi'(T) \rangle$ represents a theoretical final state. If there is no degeneracy in the ground energy level of $H_F$, $|\psi'(T)\rangle$ is identical with $|\psi_g(H_F)\rangle$, which corresponds to the single ground state obtained analytically. On the other hand, for the case $H_F$ has degenerated ground states, we define $|\psi'(T)\rangle$ as a linear combination of the ground states $|\psi_g(H_F)\rangle$, namely

$$|\psi'(T)\rangle = \sum_i C_i^g |\psi_g(H_F)\rangle,$$

where $i$ denotes the index of the degenerated ground states. The normalization condition

$$\sum_i |C_i^g|^2 = 1.$$
gives an obvious result
\[
\langle \psi'(T) | H_F | \psi'(T) \rangle = \langle \psi'_g(H_F) | H_F | \psi'_g(H_F) \rangle \tag{10}
\]
for all \( i \). Therefore, \( \Delta E \) does not depend on the choice of \( C_k \)’s.

Figure 1: Energy change of a 4-qubit system with the Hamiltonian \( H_{F1} \). Red line denotes each instantaneous ground state.

If there is no degeneracy or level crossing in the ground state, the relation
\[
\Delta E \propto O\left( \frac{1}{T^2} \right) \tag{11}
\]
in the long \( T \) limit should be found as expected from the adiabatic theorem. Figure 1 shows the energy change of a 4-qubit system during the Hamiltonian evolution from \( H_I \) to \( H_{F1} \), where \( H_{F1} \) is obtained from \( W_1 \). It can be seen that there is no degeneracy or level crossing, so that an adiabatic change is realized. Also, the residual energy shows \( O(1/T^2) \) dependence as shown in Fig.2. It agrees well with the results for quantum annealing reported by S. Suzuki[11]. Also, the results for additional two 3-qubit systems have been shown in the figure. Their qubit interactions are obtained from synaptic weight matrices
\[
W_2 = \begin{pmatrix} 0 & -1 & 1 \\ -1 & 0 & -1 \\ 1 & -1 & 0 \end{pmatrix}, \tag{12}
\]
and
\[
W_3 = \begin{pmatrix} 0 & -1 & -1 \\ -1 & 0 & -1 \\ -1 & -1 & 0 \end{pmatrix}, \tag{13}
\]
respectively.

### 3.2. Probability of Correct Answers

In the previous section, it has been confirmed that the neuromorphic AQC shows a similar behavior as the original AQC in terms of the residual energy. The investigation only of the residual energy is not sufficient for performance evaluation of the neuromorphic AQC because both solution and error states are often mixed together in a ground state of \( H_F \). Namely, one of ground states at \( t = T \) is given as
\[
|\psi'_g(H_F)\rangle = \sum_{j \in \Omega} C_j |j\rangle + \sum_{j' \in \Omega} C_{j'} |k\rangle, \tag{14}
\]
where \( \Omega \) denotes a set of solution states. In general, this mixing is not avoidable because a final Hamiltonian is obtained from qubit-qubit interactions. Therefore, it is necessary to evaluate successful rates as a function of computation time \( T \) apart from the residual energy.

First, we define the probability difference as
\[
\Delta P_g \equiv 1 - P^*,
\]
\[
= 1 - |\langle \psi(T) | \psi^* \rangle|^2
\]
\[
= 1 - \sum_j |C_j|^2 |\langle \psi(T) | \psi'_g(H_F) \rangle|^2 \tag{15}
\]
where \( P^* \) is the actual probability of the quantum state \( |\psi(T)\rangle \) found in the merged ground state \( |\psi^*\rangle \) in Eq.(8). Figure 3(a) shows the change of \( \Delta P_g \) for the final Hamiltonians \( H_{F1} \), \( H_{F2} \) and \( H_{F3} \). When we assume \( \Delta P_g \propto O(1/T^\zeta) \), \( \zeta \sim 2 \) is obtained again. However, both solution and error states are mixed together in the final quantum state \( |\psi(T)\rangle \). We should investigate probabilities of solution states, and define a probability difference given as
\[
\Delta P_s \equiv || \sum_{j \in \Omega} P'_j - P_j ||
\]
\[
= || \sum_{j \in \Omega} (|\langle j | \psi'(T) \rangle|^2 - |\langle j | \psi(T) \rangle|^2)||, \tag{16}
\]
where \( P_j \) is the actual probability of the solution state \( |j\rangle \) found in the final quantum state \( |\psi(T)\rangle \), and \( P'_j \) is the theoretical probability given from the final Hamiltonian \( H_F \).
Figure 3(b) shows the change of $\Delta P_s$ for the final Hamiltonians $H_{F1}$, $H_{F2}$ and $H_{F3}$. $\zeta \sim 2$ is obtained for $W_1$ and $W_3$. On the other hand, $\zeta \sim 1$ is obtained for $W_2$ when $T$ is large enough. Therefore, we can suppose that $\zeta$ depends on a final Hamiltonian $H_F$. This difference is related to the behavior of excited states. For $H_{F2}$ the first and second excited states are exchanged during Hamiltonian evolution as shown in Fig. 4. On the other hand, for $H_{F1}$ and $H_{F3}$ such exchange is not found. Not large but finite population changes in these excited states seem to affect to the performance significantly.

4. Conclusion

In order to evaluate the performance of AQC with neuron-like interactions (neuromorphic AQC), we have studied the residual energy and the probability of correct answers as a function of computation time $T$. Unlike the original AQC, the final state is composed of solution and error states in the neuromorphic AQC. Therefore, the performance of neuromorphic AQC cannot be evaluated only by a relation between residual energy and the computation time. It has been confirmed that the residual energy change is given as $O(1/T^2)$, and the $\Delta P_s$ change is given as $O(1/T^2)$ or $O(1/T)$. The success rate depends on the characteristics or structure of a final Hamiltonian. The results indicate that computation time must be adjusted according to a target problem.

References

Relaxation Oscillation in Single-Electron Transistor with Resistively-Shunted Gate

Yoshinao Mizugaki

Department of Electronic Engineering, The University of Electro-Communications
1-5-1 Chofugaoka, Chofu, Tokyo 182-8585, Japan
Email: mizugaki@ee.uec.ac.jp

Abstract—The author presents that a single-electron transistor with a resistively-shunted gate (RSG-SET) has an operation mode exhibiting relaxation oscillation. A shunting resistor added parallel to the gate capacitance is the key. Single-electron tunneling carries electric charge discretely into the island, while the dissipation in the shunting resistor reduces the island charge continuously. The combination of discrete and continuous charge transfer eventually results in the relaxation oscillation of the island charge. Numerical Monte Carlo simulation demonstrates the relaxation oscillation. Conditions for the bias voltage are also derived.

1. Introduction

The recent development of nanofabrication techniques has enabled us to manipulate single electrons in solid-state circuits, leading to a new field called “single electronics” [1]. Single-electron (SE) devices are essentially composed of arrays of tiny (submicron to nm order) tunnel junctions. The operation of SE devices is based on the Coulomb blockade that appears if the charging energy of a single electron, \( E_C = e^2/2C \), is much larger than the thermal energy \( k_B T \), where \( e \), \( C \), \( k_B \), and \( T \) are the fundamental charge, tunnel junction capacitance, Boltzmann constant, and absolute temperature, respectively.

Among various SE devices, a single-electron transistor (SET) shown in Fig. 1(a) is the most important device in single electronics. It comprises two small tunnel junctions connected in series. The island electrode between the tunnel junctions has a gate capacitor. The current through the junctions is controlled by sub-electron external charge, and hence, an SET is the most sensitive device to electric charge. So far, the charge sensitivity as high as \( 10^{-6} e / \sqrt{Hz} \) has been demonstrated [2].

In this paper, the author presents that a single-electron transistor with a resistively-shunted gate (RSG-SET) has an operation mode exhibiting relaxation oscillation. In the RSG-SET, a shunting resistor is added parallel to the gate capacitor as shown in Fig. 1(b). Its operation principle and numerical waveforms are demonstrated below.

Figure 1: (a) Configuration of a single-electron transistor (SET). The current through the two small tunnel junctions is controlled by sub-electron external charge, and hence, an SET is the most sensitive device to electric charge. So far, the charge sensitivity as high as \( 10^{-6} e / \sqrt{Hz} \) has been demonstrated [2].

2. Operation Principle of Relaxation Oscillation

The electric charge \( Q_0 \) and potential \( V_0 \) of the island electrode in the conventional SET shown in Fig. 1(a) are given as

\[
Q_0 = ne
\]  

(1)

and

\[
V_0 = \frac{Q_0 + C_1 V_b + C_2 V_f}{C_1 + C_2 + C_g}
\]  

(2)

where \( n \) is an integer. Its Coulomb blockade conditions are expressed as follows for the zero initial island charge, i.e.,
n = 0 [3].

\[
V_b > (Q_g - e/2)/(C_2 + C_g) \\
V_b < (Q_g + e/2)/(C_2 + C_g) \\
V_b > -(Q_g + e/2)/C_1 \\
V_b < -(Q_g - e/2)/C_1
\]

Eq. (2) is rewritten as

\[
-R_g \frac{dQ_b(t)}{dt} = \frac{Q_b(t) + C_1 V_b + C_g V_g}{C_1 + C_2 + C_g}.
\]

Besides, SE tunneling through the junctions makes a discrete shift of ±e in \(Q_b\).

Figure 3 provides a simplified explanation for the first cycle of sequential SE tunneling in the RSG-SET. When the bias voltage \(V_b\) is increased slightly beyond the boundary of the condition (4), an electron tunnels from the island to the bias voltage source through the left junction. Then, the island charge \(Q_0\) becomes \(+e\). Unlike the conventional SET, due to the dissipation in the shunting resistor, the island charge \(Q_0\) decays from \(+e\) to \(+e - \Delta\) before the subsequent SE tunneling occurs in the right junction. As a result, the residual charge of \(-\Delta\) remains in the island after the SE tunneling in the right junction. That is, one cycle of sequential SE tunneling in the junctions leaves finite charge of \(-\Delta\) in the island.

Repeating such sequential SE tunneling accumulates the residual charge in the island. Finally, even after the SE tunneling in the left junction, the accumulated charge prevents the subsequent SE tunneling in the right junction, and the sequential tunneling is suspended. In other words, the accumulated residual charge equivalently shifts the external charge until the operation point enters the Coulomb diamond.

During the intermittent of SE tunneling, the island charge decays due to the dissipation in the shunting resistor. When the island charge decreases enough to start again the SE tunneling in the left junction, the sequential SE tunneling in the RSG-SET resumes.

To obtain the relaxation oscillation, the bias voltage \(V_b\) must satisfy two conditions. Firstly, \(V_b\) must be smaller than the peak of the Coulomb diamond, that is,

\[
V_b < e/(C_1 + C_2 + C_g).
\]

Secondly, SE tunneling must occur in the left junction even when the island potential is as low as the ground level, which gives the condition of \(V_b > (-C_1 V_b + e/2)/(C_2 + C_g)\). That is,

\[
V_b > e/2(C_1 + C_2 + C_g).
\]

3. Numerical Simulation

3.1. Simulation Conditions

To verify the relaxation oscillation in the RSG-SET, the author simulated the device operation using a simulator for SE devices and circuits (SIMON) on the basis of the Monte Carlo method [4]. In the simulation, the zero temperature condition was assumed and higher-order co-tunneling processes were excluded. The device parameters were set as follows: \(C_1 = C_2 = C_g\), \(R_1 = R_2\), \(R_g = 100R_1\), and \(V_g = 0\). The time constant of the resistively-shunted gate is
100 times larger than those of the tunnel junctions, that is, $C_1 R_g = 100 C_1 R_1$.

Besides the operation of the RSG-SET shown in Fig. 1(b), the operation of the conventional SET in Fig. 1(a) was also simulated for reference.

### 3.2. Simulated Waveforms of Island Charge $Q_0$

#### 3.2.1. Results for $V_b > e/(C_1 + C_2 + C_g)$

The condition of $V_b > e/(C_1 + C_2 + C_g)$ means that the bias voltage is greater than the peak of the Coulomb diamond. Figure 4 shows the simulated waveforms of the island charge $Q_0$ for $V_b = 0.437 e/C_1$, which is greater than the peak of the Coulomb diamond of $0.333 e/C_1$. (See Fig. 2.)

The island charge $Q_0$ oscillates continuously in both the conventional SET and the RSG-SET. $Q_0$ in the conventional SET has two values of 0 and $+e$, whereas the $Q_0$ curve of the RSG-SET is tilted due to the dissipation in the shunting resistor.

#### 3.2.2. Results for $e/2(C_1 + C_2 + C_g) < V_b < e/(C_1 + C_2 + C_g)$

The conditions $e/2(C_1 + C_2 + C_g) < V_b < e/(C_1 + C_2 + C_g)$ are necessary for the relaxation oscillation in the RSG-SET. Because of $C_1 = C_2 = C_g$, the conditions are given as $e/(6C_1) < V_b < e/(3C_1)$. Results for $V_b = 0.312 e/C_1$ and $V_b = 0.187 e/C_1$ are shown in Fig. 5.

In Fig. 5(a), the island charge $Q_0$ in the SET oscillates continuously, whereas the $Q_0$ in the RSG-SET exhibits relaxation oscillation. Two lines of $Q_0 = 0.188 e$ and $0.124 e$ in Fig. 5(a) presents the boundaries of Coulomb diamond for $V_b = 0.312 e/C_1$. It can be seen that the $Q_0$ oscillation occurs only outside the Coulomb diamond.

In Fig. 5(b), the relaxation oscillation is demonstrated more clearly, while the island charge $Q_0$ in the conventional SET stays at zero. No oscillation occurs in the conventional

---

**Figure 3:** Schematic drawing of the simplified first cycle of sequential SE tunneling in the RSG-SET. $V_b$ is assumed to be slightly beyond the boundary of the Coulomb blockade condition. (a) An electron tunnels through the left junction. (b) The island charge $Q_0$ becomes $+e$. (c) Before the subsequent SE tunneling occurs in the right junction, the island charge $Q_0$ decreases from $+e$ to $+e - \Delta$ due to the dissipation in the shunting resistor. (d) After the SE tunneling in the right junction, the residual charge of $-\Delta$ remains in the island.

**Figure 4:** Simulated waveforms for $V_b = 0.437 e/C_1$. The bias voltage is greater than the peak of the Coulomb diamond of $0.333 e/C_1$. Dashed and solid curves are the results for the conventional SET and the RSG-SET, respectively.
of 0

3.2.3. Results for $V_{b}$

Figure 5: Simulated waveforms for (a) $V_{b} = 0.312e/C_{1}$ and (b) $V_{b} = 0.187e/C_{1}$. The both bias voltages satisfy the necessary conditions of $e/(6C_{1}) < V_{b} < e/(3C_{1})$. Dashed and solid curves are the results for the conventional SET and the RSG-SET, respectively. Two lines of $Q_{0} = 0.188e$ and $0.124e$ in (a), and two lines of $Q_{0} = 0.313e$ and $-0.125e$ in (b) are the thresholds of the Coulomb blockade.

SET, because $V_{b} = 0.187e/C_{1}$ is smaller than the threshold of $0.250e/C_{1}$.

3.2.3. Results for $V_{b} < e/(2C_{1} + C_{2} + C_{g})$

The island charge $Q_{0}$ does not oscillate under the small bias condition of $V_{b} < e/(2C_{1} + C_{2} + C_{g})$, of which results are shown in Fig. 6. The island charge $Q_{0}$ in the conventional SET stays at zero. Although $Q_{0}$ in the RSG-SET approaches to $-C_{1}V_{b}$, it does not exceed the threshold.

4. Conclusion

In this paper, the author demonstrated the relaxation oscillation in the RSG-SET. The RSG-SET has a shunting resistor parallel to the gate capacitor. Because of the dissipation in the shunting resistor, the island charge $Q_{0}$ becomes continuous, while the SE tunneling carries electric charge into the island discretely in units of $e$. Thus, even when the operation point is initially set outside the Coulomb diamond, sequential SE tunneling moves the operation point eventually into the Coulomb diamond, resulting in relaxation oscillation. After the operation principle

Figure 6: Simulated waveforms for $V_{b} = 0.156e/C_{1}$. The bias voltage is smaller than the critical value of $0.167e/C_{1}$. Dashed and solid curves are the results for the conventional SET and the RST-SET, respectively. Two lines at $Q_{0} = 0.344e$ and $-0.188e$ are the thresholds of the Coulomb blockade.

was schematically explained, several numerical waveforms were demonstrated. The conditions of the bias voltage for the relaxation oscillation were also derived.

Acknowledgments

The author would like to thank Prof. S. Sato of Tohoku University for fruitful discussion.

References


Rotation Angle Measurement System using Printed Spiral Inductor and Attractor of Chua’s Circuit

Takahiro KUROKOU†, Kazuhisa YOSHIMATSU†, Masayuki YAMAUCHI‡, and Mamoru TANAKA‡

†Dept. of Electronics and Computer Eng., Hiroshima Institute of Technology
2-1-1 Miyake, Saeki-ku, Hiroshima, 731-5193 Japan
‡Dept. of Information and Communication Sciences, Sophia University
7-1 Kioi-cho, Chiyoda-ku, Tokyo, 102-8544 Japan
Email:takahiro.kuroko@gmail.com

Abstract—Loosening of some hidden screws in a mechanism is a big problem. Serious troubles might be induced. Therefore, a rotation angle detection system is needed. In this paper, we suggest a rotation angle measurement system with printed spiral inductor and an attractor of Chua’s circuit. The shape of this printed spiral inductor is a chaos attractor. We call this Printed Spiral Inductor(PS-Inductor) Printed Chaos Spiral Inductor(PCS-Inductor).

1. Introduction
We think that a big trouble might be induced by a hidden screw which looses by vibration of a machine and so on. Therefore, we can say that automatic angle detection system of rotations is very important. A rotation angle detection method, which is cheap and good accuracy, is desired. Therefore, researches of rotation angle indicators are carried out up to now[1]–[4]. For example, there are measurement methods with resolver[1], optical rotation sensors[2], magnetic angle sensors[3], and the image recognition methods[4].

We suggest a rotation angle measurement system using two PCS-Inductors. Because, spiral inductor is very thin. Therefore, to overlap some spiral inductors is easy, and to use mutual inductances between spiral inductors is easy.

A spiral inductor, which is called a primary PCS-Inductor, is overlapped on another one, which is called secondary PCS-Inductor. If the secondary PCS-Inductor is fixed, and the primary PCS-Inductor is rotated, a mutual inductance between PCS-Inductor and the secondary PCS-Inductor is changed, and a rotation angle can be measured. An inductance is depend on a spacing between conductors of the spiral inductor[6]. Therefore, we can think that the mutual inductance between primary PCS-Inductor and secondary PCS-Inductor is depend on overlapping area. If PS-Inductor, which are made coupling semicircles, are used for the system, we can think that a variation of the mutual inductance is small because a variation of the overlapping area is small. Therefore, in this paper, a chaos attractor of Chua’s Circuit is used for shape of a spiral inductor. The rotation angle measurement systems with PS-Inductors and with PCS-Inductors are actually made, and these performances are investigated. We are compared the system of PCS-Inductor to the system of PS-Inductor.

2. Printed Spiral Inductor
In this study, we create two kind of spiral inductors on a printed board.
1. A spiral inductor is built up of semicircles, and is called Primary PCS-Inductor or PS-Inductor.
2. A spiral inductor is created using a chaos attractor. We name it Printed Chaos Spiral Inductor(PCS-Inductor).

2.1. PS-Inductor
Structure of the PS-Inductor is decided by six parameters (see Fig. 1). The specification of PS-Inductor are set as follows:
Maximum diameter:$D_{max}=40[mm]$, Conductor width:$W_m=0.1[mm]$, Spacing:$W_s=0.1[mm]$, Number of half-turn:$41$[half-turn], Thickness of Conductor:$D_m=35[\mu H]$, Thickness of substrate:$d_s=1.58[mm]$, Material of substrate:Bakelite.
A dextral PS-Inductor is shown in Fig. 1. In this study, we make a dextral PS-Inductor and a sinistral PS-Inductor.
The each parameter of each PS-Inductor is same value. A PS-Inductor, which is called secondary PS-Inductor, is overlapped on another one, which is called primary PS-Inductor, as like Fig. 2, and a mutual inductance between PS-Inductors is measured while a secondary PS-Inductor is rotated.

A state, which two PS-Inductors are overlapped exactly, is assumed as 0 degree (see Fig. 2). Therefore, dextral and sinistral PS-Inductors are made. Measurement results of these actual PS-Inductors are as below. An inductance of dextral PS-Inductor is 17.24 µH. An inductance of sinistral PS-Inductor is 17.07 µH.

2.2. Printed Chaos Spiral Inductor

<Chua’s circuit>

Figure 3(a) shows Chua’s circuit model, and differential equations of Chua’s circuit are shown Eqs. (1)–(3).

\[ C_1 \frac{dv_{c_1}}{dt} = \frac{1}{R}(v_{c_2} - v_{c_1}) - g(v_{c_1}), \]  
\[ C_2 \frac{dv_{c_2}}{dt} = \frac{1}{R}(v_{c_1} - v_{c_2}) + i_L, \]  
\[ L \frac{di_L}{dt} = -v_{c_2}, \]  

where \( v_{c_1} \) and \( v_{c_2} \) are voltages of \( C_1 \) and \( C_2 \) respectively, \( i_L \) is a current of \( L \), and \( g(v_{c_1}) \) is a characteristic of Chua’s diode. The characteristic of Chua’s diode is shown in Fig. 3(b), and an equation of the characteristic is shown below.

\[ g(v_{c_1}) = m_0 v_{c_1} + \frac{1}{2}(m_1 - m_2)|v_{c_1} + b_p| + \frac{1}{2}(m_0 - m_1)|v_{c_1} - b_p|. \]  

These differential equations are simulated by using fourth-order Runge-Kutta method, and we get a chaos attractor. These parameters are shown in Table 1.

<Optimization of the attractor>

PCS-Inductor using the chaos attractor is needed to fulfill below conditions for our rotation angle measurement system because an inductance of the PCS-Inductor should be large value. If the inductance is too small value, we must use very high frequency, and cost of our system becomes high.

- The line must not intersect with the line when the attractor is mapped to the X-Y plane.

An inductance is depend on a spacing between conductors. We can think that the mutual inductance between primary PCS-Inductor and secondary PCS-Inductor is depend on overlapping area of the spacing. Therefore, the shape of PCS-Inductor like a circle is desired. The chaos attractor used in our system is shown in Fig. 4.

Each coordinate of the attractor is assumed as \((x_k, y_k)\), and each coordinate of an optimization attractor is assumed as \((X_k, Y_k)\). The “\( k \)” is changed from 0 to the number of total points of the attractor. Each coordinate of the base attractor is remap below equations.

\[ X_k = 33x_k, Y_k = 99y_k \]  

The dextral PCS-Inductor are shown in Fig. 5. We make dextral and sinistral PCS-Inductors using the optimized chaotic attractor. Measurement results of each actual PCS-Inductor are shown below. When measurement frequency is 60 MHz, an inductance of dextral PCS-Inductor is 4.12 µH, and an inductance of sinistral PCS-Inductor is 4.32 µH. These PCS-Inductors are overlapped each other in a similar way to PS-Inductors (see Fig. 2).
3. Rotation Angle Measurement System

In our system, the secondary PS-Inductor(or PCS-Inductor) is fixed, and the primary PS-Inductor(or PCS-Inductor) is rotated. Rotation angles are measured by voltage transmission efficiency which changed by using a variation of mutual inductance between primary and secondary PS-Inductor(or PCS-Inductor).

3.1. Circuit model

Our measurement circuit model is shown in Fig. 6. The filter circuit is constructed by using two inductors, a mutual inductor, three resistors, and two capacitors.

3.2. Rotation angle measurement system with PS-Inductors

Frequency characteristics of the system with two PS-Inductors, which are exactly overlapped each other, are shown in Fig. 7. Our rotation angle measurement system needs large variation when the mutual inductance is changed a little by rotating the primary PS-Inductor. Therefore, the frequency of $v_{in}$ is fixed 16 MHz.

When the primary PS-Inductor is rotated from 0 degree to 360 degrees, the voltages $v_{in}$ and $v_{out}$ are measured every 5 degrees, and the voltage transmission gains is calculated(see Fig. 8). We need large variation of voltage transmission gains, but, large variations can not be obtained by the system with PS-Inductors.

3.3. Rotation angle measurement system with PCS-Inductors

Frequency characteristics of the system with two PCS-Inductors, which are exactly overlapped each other, are shown in Fig. 9. A resonance frequency of our system is around 57.8 MHz.

We think that large variation can be obtained by small variation of the mutual inductance when a frequency of $v_{in}$ is fixed as around 57.8±10 MHz. The voltage transmission gains are investigated, when the frequency of $v_{in}$ is fixed as 1 MHz, 56 MHz, or 67 MHz. The voltage transmission gains of 1 MHz is shown in Fig. 10. The voltage transmission gains is obtained from 0 degree to 360 degrees every 10 degrees because a large variation can not be observed. We can think that detection of the rotation angle is hard if the frequency is not close to the resonance frequency. The voltage transmission gains of 56 MHz and 67 MHz are investigated from 0 degree to 360 degrees every 1 degree, and are shown in Fig. 11. The voltage transmission gains is widely changed by few rotation angles. Our system needs large variation of the voltage transmission gains, and therefore, the frequency of $v_{in}$ fixed to around 67 MHz.

4. Conclusion

In this study, we developed a rotation angle measurement system using two PCS-Inductors. A shape of the Printed Chaos Spiral Inductor(PCS-Inductor) was used a Chua’s circuit attractor. The voltage transmission gains was widely
changed by rotating the primary PCS-Inductor when the frequency of the system was fixed to 56 MHz or 67 MHz. Especially, the frequency should be fixed to 67 MHz for our system. If combinations between angles and voltage transmission gains are already known and a start angle of rotation is already known, we made clear that the angle can be known by our system.

References


Inter-connection of Parallel Connected Class D Amplifiers Operated at Different Switching Frequencies

Yusuke Ishikawa†, Ryuta Yamamoto†, Hiroyuki Uchiyama†, Xiuqin Wei‡, Hiroo Sekiya‡, and Tadashi Suetsugu†
†Department of Electronics Engineering and Computer Science
Fukuoka University, Fukuoka, Japan
‡Chiba University, Chiba, Japan
suetsugu@fukuoka-u.ac.jp

Abstract—This paper researches about influence of multi-frequency operation of parallel connected class D amplifiers. It can be seen that variation in dc supply voltage of one amplifier leg does not influence to output waveform of another amplifier leg. However, parallel connection of two amplifiers influences output power of both amplifiers due to change in impedance of output filter. It is proposed to insert decoupling filter to the output of higher frequency amplifier to reduce the coupling. It is shown that decoupling filter successfully improved coupling between amplifiers.

I. INTRODUCTION

Class D amplifier is one of next generation power source for plasma generator due to its high efficiency characteristics [1][2]. In the plasma generators, RF power is often imposed with combination of several frequency components. For example, an RF generator for semiconductor processing inputs 2 MHz and 13.56 MHz RF power to the chamber [3]. In order to impose RF powers with several frequencies, several power amplifiers with different operating frequencies are operated in parallel. If power amplifiers are connected in parallel, there is electric coupling between the amplifiers. Operation of class D amplifier [4] is strongly influenced by condition of load impedance. It is important to investigate the influence of parallel operation of amplifiers with different frequencies. However, this influence is not well researched for class D amplifiers yet. In this paper, parallel connected two class D amplifiers which are operated at different frequencies is simulated with PSPICE to obtain influence of coupling of different frequencies.

Fig. 1. Parallel connected class D amplifier with different operating frequencies.
II. PARALLEL CONNECTED CLASS D AMPLIFIERS WITH DIFFERENT OPERATING FREQUENCIES

Parallel connected class D amplifier analyzed in this paper is shown in Fig. 1. Two class D amplifiers with different operating frequencies are connected in parallel. One leg of class D amplifier is a voltage switching type class D amplifier shown in Fig. 2. The class D amplifier is composed of dc supply voltage \( V_{DD} \), two MOSFET transistor switches, output resonant LC filter \( C-L \), and load resistance \( R \). The output LC filter \( C-L \) has a high loaded quality factor and a resonant frequency \( f_o \) near the switching frequency \( f \) so that the output current can be pure sinusoid at the switching frequency. The MOSFET transistors switch in turn periodically and it produces rectangular voltage excitation at drain voltage \( v_S \). The output LC filter \( C-L \) extracts the fundamental frequency component of drain voltage and output to the load resistance \( R \).

Fig. 3 shows drain voltage waveform and output voltage waveform of the class D amplifier. When the duty ratio is 0.5, the circuit parameters are [2]

\[
R = \frac{2V_{DD}^2 \cos^2 \varphi}{\pi^2 P} \tag{1}
\]

\[
C = \frac{1}{\omega_o QR} \tag{2}
\]

\[
L = \frac{QR}{\omega_o} \tag{3}
\]

where \( P \) is output power (100% power efficiency is supposed), \( \varphi \) is phase angle of output current, \( Q \) is loaded quality factor of output LC filter, and \( \omega_o \) is

\[
\omega_o = 2\pi f_o = \frac{4\pi}{\tan \varphi + \sqrt{\tan^2 \varphi + 4}} \tag{4}
\]

In general, \( f_o \) is set to less than \( f \) in order to ensure inductive load current in transistor. If \( \varphi = 30 \text{ deg} \), \( V_{DD} = 250 \text{ V} \), \( P = 190 \text{ W} \), and \( Q = 10 \) at 13.56 MHz, circuit parameters are calculated as follows; \( C = 24.2 \text{ pF} \), \( L = 6.0 \text{ µF} \), and \( R = 50 \Omega \). And at 27.12 MHz, \( C = 12 \text{ µF} \), \( L = 3 \text{ µF} \), and \( R = 50 \Omega \). At 40.68 MHz, \( C = 8 \text{ pF} \), \( L = 2 \text{ µF} \), and \( R = 50 \Omega \). At 60.00 MHz, \( C = 5.5 \text{ pF} \), \( L = 1.4 \text{ µF} \), and \( R = 50 \Omega \).

III. SIMULATION RESULTS

In the simulation, 3 circuits who have different patterns of frequency combinations were tested; they are (1) 13.56 MHz + 27.12 MHz, (2) 13.56 MHz + 40.68 MHz, and (3) 13.56 MHz + 60.00 MHz. Simulated circuit has 50 \( \Omega \) load resistance. In this simulation, IXYS DEI175-102N12A was used for MOSFET.

In the first, supply voltage of leg #1, \( V_{DD1} \) was varied from 0 V to 250 V. Fig. 4 shows time domain waveforms of drain voltage of \( S1 \) \( v_{S1} \), drain voltage of \( S2 \) \( v_{S2} \), and output voltage \( v_o \) when leg #1 was operated at 13.56 MHz and leg #2 was operated at 27.12 MHz. It can be seen waveform of \( v_{S1} \) was proportional to \( V_{DD1} \). However, waveform of \( v_{S2} \) was not varied with \( V_{DD1} \). Waveform of output current \( i_o \) was superimposing of two different frequency sinusoids. Fig. 5 shows FFT of waveforms in Fig. 4. It can be seen that spectrum of \( v_{S1} \) has third harmonic component but does not have second harmonic component. Therefore, there is very low influence between \( v_{S1} \) and \( v_{S2} \). It can be also seen that on frequency component of \( i_o \) at 13.56 MHz varied with \( V_{DD1} \). But, frequency component at 27.12 MHz was not varied with \( V_{DD1} \). Namely, input voltage (input power) of low frequency component did not influence to the output power of high frequency component when 13.56 MHz and 27.12 MHz were imposed.

Same simulations were done for circuit (2) and (3). Fig. 6 shows plots of FFT results of (a) 13.56 MHz components and (b) 27.12 MHz, 40.56 MHz, and 60.00 MHz components of output current \( i_o \) of (1), (2), and (3) circuits. As seen in Fig. 6(a), 13.56 MHz component was proportional to \( V_{DD1} \). (Some plots were not obtained due to PSpice convergence problem.) From Fig. 6(b), it can be seen that frequency components at 27.12 MHz, 40.56 MHz, and 60.00 MHz did not varied with \( V_{DD1} \). These two plots indicate that variation of input voltage
for 13.56 MHz did not influence to output power of other amplifier leg. However, three plots in Fig. 6(a), i.e., 13.56 MHz + 27.12 MHz, 13.56 MHz + 40.56 MHz, and 13.56 MHz + 60.00 MHz, have different values. This difference is due to change in impedance of output filter of amplifier #1. Namely, connection of amplifier #2 influences impedance of output filter seen from amplifier #1.

Similar simulation were done for variation of $V_{DD1}$ for 13.56 MHz did not influence to output power of other amplifier leg. However, three plots in Fig. 6(a), i.e., 13.56 MHz + 27.12 MHz, 13.56 MHz + 40.56 MHz, and 13.56 MHz + 60.00 MHz, have different values. This difference is due to change in impedance of output filter of amplifier #1. Namely, connection of amplifier #2 influences impedance of output filter seen from amplifier #1.

Similar simulation were done for variation of $V_{DD2}$. Fig. 7 shows plots of FFT results of (a) 13.56 MHz components and (b) 27.12 MHz, 40.56 MHz, and 60.00 MHz components of output current $i_o$ of (1), (2), and (3) circuits when $V_{DD2}$ was varied from 0 V to 250 V. From Fig. 7(a), it can be seen that frequency components at 13.56 MHz did not varied with $V_{DD2}$. This plot indicates that variation of input voltage $V_{DD2}$ did not influence to output power of other amplifier leg.

IV. REDUCTION OF COUPLING BETWEEN PARALLEL CONNECTED AMPLIFIERS

In order to reduce coupling between parallel connected amplifiers, it is proposed to insert decoupling filter to the output of one amplifier which has higher output frequency. Fig. 8 shows proposed circuit which has decoupling filter. Decoupling filter is a band cut filter which has notch frequency 13.56 MHz, i.e., output frequency of amplifier #1. Fig. 9 shows FFT results of 13.56 MHz component of output current $i_o$ of improved circuits. In this simulation, values of decoupling filter were $C_C = 23\mu\text{F}$, $L_C = 6\mu\text{H}$. Comparing with Fig. 6(a), difference between three plots are reduced.

V. CONCLUSIONS

This paper influence of multi-frequency operation of parallel connected class D amplifiers were researched. It can be seen that variation in dc supply voltage of one amplifier leg does not influence to the output waveform of another amplifier.
However, parallel connection of two amplifiers influence output power of both amplifiers due to change in impedance of output filter. It was proposed to insert decoupling filter to the output of higher frequency amplifier to reduce the coupling. It was seen that decoupling filter successfully improved coupling between amplifiers.

VI.  REFERENCE


Stochastic Resonance in a Simple Electric Circuit having a Double-Well Potential

~Circuit Experiments with a Single Operational Amplifier~

Akira Utagawa†, Tetsuya Asai, and Yoshihito Amemiya

†: utagawa@lalsie.ist.hokudai.ac.jp
Graduate School of Information Science and Technology, Hokkaido University.
Kita 14, Nishi 9, Kita-ku, Sapporo, 060-0814 Japan.

Abstract—In this report, we propose a double-well potential system that can easily be implemented by a single operational amplifier. The system is described by the same dynamics as traditional analog neurons. We first introduce a potential function obtained from the proposed dynamics and explain the bistable conditions. Then we describe how we constructed an electronic SR system that is implemented by a single operational amplifier, show experimental results of the proposed circuit, and demonstrate that the circuit exhibits the same SR behavior demonstrated in traditional double-well potential systems [1].

1. Background

Stochastic resonance (SR) is a phenomenon where a system can stochastically detect a weak input signal with the help of external noise when the input signal is below the system’s threshold [1]. SR has been observed in many systems such as threshold systems [2], monostable systems [3], and bistable systems (double-well potential systems) [4]. Moreover, the SR has also been found in electrical applications such as threshold circuits [5], bistable circuits [6], and semiconductor laser [7]. SR may be utilized for weak signal detection in electric circuits [8]. Especially, SR in double-well potential electronic systems, i.e., bistable electronic systems, can be used not only for weak signal detection, but also for logic memories.

SR-utilizing logic memory circuits may be useful when the supply voltage of the circuit is extremely low. Because electric power consumption of digital circuits is proportional to the square of the supply voltage, decreasing the supply voltage is very effective to reduce power consumption. However, decreasing the supply voltage causes data writing to fail or stored data to be lost due to threshold voltage deviation of MOSFETs. If the threshold deviation becomes the dominant factor, SR would decrease SRAM cell failure rates. When threshold deviation disturbs data writing, noise and fluctuations change the potential barrier of double-well potential of a memory cell, and the data writing would succeed. Noise sources exist everywhere. For instance, power supply noise in LSIs [9] can be used as a noise source.

Memory cells of SRAMs must be designed with minimal size because integration density of the cell must be high. The most widely used memory cell is a latch circuit composed of two inverter circuits. In order to construct SR-utilizing memory cells, obtaining potential function of the latch circuit is required to evaluate SR characteristics. However, obtaining the potential function is not easy. So, we propose a mathematical model whose potential function can be obtained theoretically and construct an electric circuit which is equivalent quantitatively to the latch circuit.

2. Stochastic resonance in double-well potential system

In this paper, we will use the following dynamics,

$$\tau \frac{du}{dt} = -u + f_\beta(u - I),$$  \hspace{1cm} (1)

where \( f_\beta(\cdot) \) is the sigmoid function whose slope factor is \( \beta \) and \( I \) the external input signal. Suppose that \( \beta \) is large enough, so \( u \to 1 \) when \( u > I \), whereas \( u \to 0 \) when \( u < I \). Thus one can be convinced that this system is a bistable system.

Next, potential function of this system \( H \) will be given. When the following equation,

$$\frac{\partial H}{\partial t} = \frac{du}{dt} \frac{\partial H}{\partial u} < 0,$$  \hspace{1cm} (2)

is satisfied, the system is considered to be stable. One can easily notice that

$$\frac{\partial H}{\partial u} = -\tau \frac{du}{dt},$$  \hspace{1cm} (3)

is a candidate that satisfies the condition. By substituting Eq. 1 to the above equation, the following equation,

$$\frac{\partial H}{\partial u} = u - f_\beta(u - I),$$  \hspace{1cm} (4)
Figure 1: Double-well potential function of proposed system

is obtained. Integrating this by \( u \) leads to the following potential function,

\[
H = \frac{1}{2} u^2 - \frac{1}{\beta} \ln(\exp(\beta u) + \exp(\beta I)) + C,
\]

where \( C \) the integral constant.

Figure 1 plots the obtained potential function (\( \beta = 30, \ C = 0 \)). Here consider \( I = A \sin(t) + B \), where \( A \) the amplitude and \( B \) the offset. When \( I = 0.5 \) [Fig. 1(a), (c)], the middle potential barrier becomes the highest. In this condition, the system holds current state \( u = 1 \) or \( u = 0 \). Setting the external input \( I \) to 0 (or 1) changes the internal state to 1 (or 0). Figure 1(b) and (c) show the potential curves when \( I \) is set to 0.2 and 0.8. When the system accepts noises, the transition may occur. This small barrier can be surpassed by applying moderate noises. Small amounts of noise cannot cause the transition, whereas excessive amounts of noise cause uncorrelated transitions of the state with the input signal. This phenomenon is called “stochastic resonance in the double-well potential system” [1].

3. Electric circuits having double-well potential

The double-well potential system proposed in the previous section was implemented in electric circuits with only one operational amplifier. A fundamental property of the amplifier is reviewed briefly here. The operational amplifier has 2-input (\( V_+ \) and \( V_- \), for example) and it amplifies the voltage difference between \( V_+ \) and \( V_- \) with gain \( A_v \). The output \( A_v \cdot (V_+ - V_-) \) is clamped at supply voltages (\( V_{dd}, V_{ss} \)). Thus, when the gain is large enough and \( V_{ss} = 0 \), the amplifier’s output is approximated by \( V_{dd} \cdot \theta(V_+ - V_-) \), where \( \theta(\cdot) \) is the step function.

Suppose the time constant \( \tau \) in Eq. (1) is small enough, then we obtain \( u \approx f_\beta(u - I) \) from Eq. (1). When \( \beta \) is large enough, we can say \( f_\beta(\cdot) \approx \theta(\cdot) \). When we consider \( V_{out} \) as \( u \cdot V_{dd} \) and \( V_{in} \) as \( v \cdot V_{dd} \), we obtain the approximate equation,

\[
V_{out} \approx V_{dd} \cdot \theta(V_{out} - V_{in} - V_{offset}).
\]

The right side of the above equation is equal to the estimated output voltage of the amplifier where “+” node is set to \( V_{out} \) and “−” node is set to \( V_{in} \). Thus, by connecting the output node of the amplifier and “+” node, the system for Eq. (1) is implemented in electric circuits.

4. Simulation and experimental results

Figure 2 shows the potential function of the proposed circuit. Time courses of \( V_{out} \) was obtained from SPICE simulations (TSMC 0.18 \( \mu \)m CMOS parameter, \( V_{dd} = 1.8 \) V, \( V_{offset} = 0.3 \) V) and the potential function in Fig. 2 was numerically obtained by calculating \( dV_{out}/dt \) from the simulation results and integrating \( dV_{out}/dt \) by \( V_{out} \).

Figure 3 shows the circuit configuration. A CMOS full-swing (Rail-To-Rail) operational amplifier (National semiconductor, LMC6482) was used and supply voltage \( V_{dd} \) was set to 3.0 V. A input signal applied to the amplifier was \( V_{in} = V_{in} + V_n \), where \( V_{in} \) the sinusoidal input and \( V_n \) the noise voltage. \( V_{in} \)
was generated by a resistive voltage divider ($R = 1\ \text{k}\Omega$). $V'_\text{in}$ was applied to "-" node (an input node of the proposed circuit). Furthermore, $V_{\text{in}}$ was given by $V_{\text{cm}} + V_A \cdot \sin (2\pi f_0 t)$, where $V_{\text{cm}}$ was 1.5 V, $V_A$ was 1 V, and $f_0$ was 200 Hz. $V_{\text{in}}$ was the time varying Gaussian noise voltage whose average and standard deviation were 0 V and $\sigma$ V. Both $V_{\text{in}}$ and $V_{n}$ were given by a waveform generator (HIOKI, 7075). Pseudo-random sequences were generated by using the Box-Muller method from a computer simulation. $V_{n}$ was the Pseudo-random sequence that was imported to the waveform generator with frequency limitation of 19 kHz. Note that our circuit has the simpler structure than the double-well potential electronic system proposed in [10]. We observed waveforms of the input and output voltages ($V'_{\text{in}}$ and $V_{\text{out}}$) by a oscilloscope (Techtronix, TDS784D) and sampling rate was 100 kHz. We also observed power spectrum of the output voltage ($V_{\text{out}}$) by a FFT module, which is equipped with the oscilloscope (averaged over 1000 times in frequency domain), and obtained SNR by subtracting the background noise level on $f_0$ from the signal level on $f_0$.

Figure 4 shows experimental results (waveform screens of the oscilloscope) where $\sigma$ was set to 0.3 V, 0.75 V, and 1.5 V. Each figure, (a), (b), and (c), contains time courses of $V'_{\text{in}}$ (upper), time courses of $V_{\text{out}}$ (middle), and power spectrum of $V_{\text{out}}$ (lower). When $\sigma = 0.3$ V, probability of $V_{\text{out}}$ transition was small [Fig. 4(a)]. Because the offset and amplitude of $V_{\text{in}}$ were 1.5 V and 2 Vpp ($V_{\text{in}} = 0.5$–2.5 V), the minimal voltage of $V_{n}$ for transitions from ‘1’ to ‘0’ and from ‘0’ to ‘1’ are +0.5 V (when $V_{\text{in}}$ is 2.5 V) and -0.5 V (when $V_{\text{in}}$ is 0.5 V), respectively. This indicates that when $\sigma$ were 0.3 V, a possibility of $V_{n} > 0.5V$ (or $V_{n} < -0.5V$) is very small. When $\sigma$ was 0.75 V (Fig. 4(b)), SNR was 21.5 dB@200 Hz. When $\sigma$ was 1.5 V (Fig. 4(c)), SNR was 18.7 dB@200 Hz.
V_n < −0.5 V) and the transition possibility were small [Fig. 4(a)]. We measured signal and background level at 200 Hz in power spectrum and found that SNR was 6.4 dB. When σ = 0.75 V, possibility of V_n > 0.5 V (or V_n < −0.5 V) was higher than the transition possibility of σ = 0.3 V, and transition possibility was also higher [Fig. 4(b)]. The SNR in this case was 21.5 dB. The important fact is that the possibility of the transition from 0 to V_{dd} became high when V_in is low, and the possibility of the transition from V_{dd} to 0 became high when V_in was high. In other words, although V_in didn’t have the amplitude required for the transition, the noise stochastically helped the transition of V_out to V_{dd} (or 0) depending on V_in. The experimental results in σ = 1.5 V are shown in Fig. 4(c). In this case, the SNR was 18.7 dB. The SNR didn’t decrease suddenly as it seems, but the frequent transition of V_out occurred. This is because the noise level almost always surpassed the signal level required for transition.

Figure 5 shows the experimental SNR curve that was obtained by varying standard deviation of noise σ from 0 to 4 V. The maximum SNR was 21.5 dB (σ = 0.75 V).

5. Conclusion

We proposed a double-well potential system that can easily be implemented by a single operational amplifier. We first obtained a potential function of the system and its bistable conditions. Then, we constructed a simple electric circuit based on the system. And we conducted experiments of the circuit. We confirmed the same SR behavior observed in conventional double-well potential systems [1].

Acknowledgments

This study was supported by a Grant-in-Aid for Scientific Research on Innovative Areas [20111004] from the Ministry of Education, Culture Sports, Science and Technology (MEXT) of Japan.

References

New Idea of the Pseudo-Inverse Maps in Optimal Pre-Correction of Nonlinear Systems as the Result of Modeling and Optimal Past-Correction

Grzegorz Ciesielski† and Paulina Sobanska‡

†Division of Theoretical Electrotechnics, Institute of Theoretical Electrotechnics, Metrology and Materials Sciences, Technical University of Lodz, Ul. B. Stefanowskiego 18/22, 90-924 Lodz, Poland
‡INWAT Ltd., Research and Design Company, Ul. H. Sienkiewicza 101/109, 90-301 Lodz, Poland
Emails: gregulus@p.lodz.pl, gregulus@toya.net.pl or gregulusus@gmail.com and pola.sob@gmail.com

Abstract — This paper presents the new idea of the pseudo-inverse maps applied to the optimal pre-corrections of nonlinear systems. This concept is a result of search for optimal models and optimal past-correctors of nonlinear systems from perspective of the Functional Theory of Nonlinear Systems which is discussed in this article. Considered systems are multidimensional, all of input and output signals are real or complex valued and their sets are finally equipped with the structure of the Hilbert spaces. All maps used in this paper are functions for the static systems, convolutions for the linear time-invariant systems, and nonlinear operators for the nonlinear systems. It is shown, that the nonlinear system past- and pre-corrections can be reduced to the modeling tasks of some systems which can be reduced further to the generalized least mean square (LMS) approximations.

1. Introduction

The beginning of the Theory of Nonlinear Systems dates back to 1887 when Vito Volterra announced Theory of his Series which was developed in the course of subsequent research. The Volterra Series, although inconvenient from the numerical point of view, play the role in the Theory of Nonlinear Systems as significant as Convolutions in Modeling of Time-Invariant Linear Systems. Another giant step in the Theory of Nonlinear Systems took place around 1920, when Norbert Wiener, the creator of Cybernetics, conducted research on Orthogonal Operators with random inputs, nowadays known as Wiener Operators. These operators are magnificent from the numerical point of view due to their orthogonality. Later, since about 1962, the significant progress in the Theory of Nonlinear Systems has been made by Martin Schetzen, the brilliant follower of Wiener (M. Schetzen [9]). In 1970s an equally great contribution to the Theory of Nonlinear Systems was made by Russians K. A. Pupkov, W. I. Kapalin and A. S. Jushtchenko (K. A. Pupkov et al. [8]). Afterwards outstanding papers by Irvin W. Sandberg, Rudi J. P. de Figueiredo et al. appeared. One of the authors of this article has dealt with past-correction of static systems since 1977, initially in a purely electronic way and then numerically with the use of the Splines. His further work led him eventually to the Theory of Nonlinear Systems in which he applied methods resulting from advanced Functional Analysis, Nonlinear Analysis, Theory of Lebesgue Measure and Integral, Algebra and Topology, although he is not a mathematician by education. In 1994 this work led him to the Functional Theory of Nonlinear Systems (G. Ciesielski [4]), which is a generalization of the classical approach used in the Theory of Nonlinear Systems. This uniform generalized Theory shows among other things that there is no need to treat Volterra and Wiener Theories of Nonlinear Systems as two separate theories as wrote Martin Schetzen in his book. The second author of this article has recently begun her scientific experience, starting with a successful application of neural networks to modeling of highly complex real systems, such as steam turbines (P. Sobanska, P. Szczepaniak [10]).

2. Optimal Modeling

Let us assume further in the whole of article that $F$ is the real or complex number field, $U_F$ is the normed space over the number field $F$ of some maps with the norm $\|\cdot\|_U$ and $V_{[k]} := (v_i \in U_F)_{[k]}^k$ is a system of some maps from $U_F$. For the given modeled system described by the map $\phi \in U_F$ we search for its model $\phi_k \in \text{span} V_{[k]} \subset U_F$ as the linear combination of elements of the system $v_{[k]}$ not necessarily linearly independent, so $\phi_k = c_{[k]}(v) \in \text{span} V_{[k]}$. Let on $U_F$ the functional $\sigma_{\text{mod}} \text{span} V_{[k]}(\phi) := \min_{v \in \text{span} V_{[k]}} \|\phi - v\|_U$. The coefficients $c_{[k]}$ of the optimal model $\phi_k$ should satisfy the equation: $\|\phi - \phi_k\|_{U} = \sigma_{\text{mod}} \text{span} V_{[k]}(\phi)$. As we can see now, this modeling task leads us to the problem of approximation of the system $\phi$ by the linear combination of elements of the system $\phi_k$ what can be very easily computed.

3. Optimal Past-Correction

Let us assume further in the whole of article that $V_F$ and $W_F$ are the normed spaces of some maps with the norms $\|\cdot\|_V$ and $\|\cdot\|_W$ adequately. These spaces satisfy the condition: $\forall u \in U_F \forall v \in V_F (v \circ u \in W_F)$. Let $V_{[k]} := (v_i \in V_F)_{[k]}^k$ is a system of some maps from $V_F$. For the given corrected system $\phi \in U_F$ and required system $\eta \in W_F$ we search for the past-corrector $\gamma_k := c_{[k]}(v) \in \text{span} V_{[k]} \subset V_F$ as the linear combination of elements of the system $\gamma_k$ not necessarily linearly independent, so $\gamma_k = c_{[k]}(v) \in \text{span} V_{[k]}$. Similarly as previously, let the func-
tional $\sigma_{\text{past span } v[k]}(\phi, \eta) := \min_{F \in \text{span } v[k]} ||\eta - y \circ \phi||_W$ on $U_F \times W_F$. The coefficients $c[k]$ of the optimal past-corrector $y_k$ satisfy the equation: $||\eta - y_k \circ \phi||_W = \sigma_{\text{past span } v[k]}(\phi, \eta)$.

This is the unconstructive form of determination of the past-corrector of the system $\phi$ by the linear combination of the maps $v[k]$ for the given required system $\eta$. The unconstructiveness of this form arises from the fact that there is no well-known mathematical method of solving this problem. But after a bit of consideration, we can notice that this approach can be rearranged to the constructive form of determination of the past-corrector of the system $\phi$. This equivalent constructive form consists in the modeling of the required system $\eta \in W_F$ by the linear combination of the maps $\omega[k] := (\omega_i \in W_F)_k^{i} = (v_i \circ \phi \in W_F)_k^{i}$. Therefore, for the given required system $\eta$ we search for its model $\eta_k \in \text{span } \omega[k]$ as the linear combination of elements of system $\omega[k]$, perhaps linearly dependent, so $\eta_k = c[k]^{i} \omega[i] := \sum_{i=1}^{k} c[i] \omega[i]$. The coefficients $c[i]$ of the optimal past-corrector $y_k$ are the same as for the optimal model $\eta_k$ of the required system $\eta$ and satisfy equation: $||\eta - \eta_k||_W = \sigma_{\text{mod span } \omega[k]}(\eta)$. Obtained proceeding can be easily computed.

4. Pseudo-Inverse System

Let us assume that the identity map $\text{id} \in W_F$ or some its approximation $\tilde{\text{id}} \in W_F$ which is chosen by us. We can notice from the optimal past-corrector task that if $\eta = \text{id}$ in the exact case we get $y_k = \phi^{-1}$, if inverse map $\phi^{-1}$ exists, because as we well know the inverse map $\phi^{-1} \circ \phi = \text{id}$. From mathematical analysis we know that the inverse map $\phi^{-1}$ exists if and only if the map $\phi$ is a bijection. This theorem limits us to such a simple approach to find inverse system $\phi^{-1}$ in the exact case. But, if we make such approach to the optimal past-corrector task previously discussed we can obtain the pseudo-inverse system $\phi^*$. So, let us define on $U_F$ the functional $\sigma_{\text{inv } v[k]}(\phi) = \sigma_{\text{past } v[k]}(\phi, \text{id})$. The system denoted by $\phi^*$ which satisfies the equation: $||\text{id} - \phi^* \circ \phi||_W = \sigma_{\text{inv } v[k]}(\phi)$ we will call the pseudo-inverse system for the system $\phi$ in $V_F$. We will denote the compound map $\phi^* \circ \phi$ as $\text{id}$. In approximate approach, for the given system $\phi \in U_F$ we search for the pseudo-inverse system $\phi^*_k \in \text{span } v[k]$ as the linear combination of elements of system $v[k]$ not necessarily linearly independent, so $\phi^*_k = c[k]^{i} v[i] := \sum_{i=1}^{k} c[i] v[i]$. The coefficients $c[k]$ of the optimal pseudo-inverse system $\phi^*_k$ satisfy equation: $||\text{id} - \phi^*_k \circ \phi||_W = \sigma_{\text{inv span } v[k]}(\phi)$. Let us denote the compound map $\phi^*_k \circ \phi$ as $\text{id}_k = c[k]^{i} \omega[i] := \sum_{i=1}^{k} c[i] \omega[i]$. Similarly as previously, the coefficients $c[k]$ of the optimal pseudo-inverse system $\phi^*_k$ are the same as for the optimal model $\text{id}_k$ of the identity map $\text{id}$ and should satisfy equation: $||\text{id} - \text{id}_k||_W = \sigma_{\text{mod span } \omega[k]}(\phi)$. Obtained proceeding can be easily computed.

5. Optimal Pre-Correction

For the given corrected system $\phi \in V_F$ and required system $\eta \in W_F$ we search for the pre-corrector $y_k \in \text{span } v[k] \subset U_F$ as the linear combination of elements of system $v[k] := (v_i)_k^{i}$ not necessarily linearly independent, so $y_k = c[k]^{i} v[i] := \sum_{i=1}^{k} c[i] v[i]$. Let the functional $\sigma_{\text{pre span } v[k]}(\phi, \eta) := \min_{v[k]} ||\eta - \phi \circ \gamma||_W$ on $V_F \times W_F$. The coefficients $c[k]$ of the optimal pre-corrector $y_k$ satisfy the equation:

$$||\eta - y_k \circ \phi||_W = \sigma_{\text{pre span } v[k]}(\phi, \eta).$$

We obtain the unconstructive form of determination of the pre-corrector of the system $\phi$ by the linear combination of elements of system $v[k]$ for the given required system $\eta$. The unconstructiveness of this form arises from the fact that there is no well known mathematical method of solving this difficult problem. After due consideration, we can notice that this approach can be rearranged to the constructive form of determination of the pre-corrector of the system $\phi$. From the left side of the equation (*) we have $||\phi^* \circ \eta - y_k \circ \phi||_W = ||\phi^* \circ \eta - y_k \circ \phi||_W$. The system $\phi^* \circ \phi = :\text{id}$ is some approximation of the identity map $\text{id}$. If this approximation is good then we can assume that $\text{id} \equiv \text{id}$. So, the coefficients $c[k]$ of the optimal pre-corrector $y_k$ are approximately the same as for the optimal model $y_k$ of the system $\phi^* \circ \eta$ which satisfy the equation: $||\phi^* \circ \eta - y_k ||_W = \sigma_{\text{mod span } v[k]}(\phi^* \circ \eta)$. Obtained proceeding can be relatively easily computed.

This result is especially interesting e.g. from the electrical circuit diagrams or metrology perspective.

6. Generalized Theorem on the LMS Approximation

One of the main problems which came out here is the linear dependence of the sets $v[k], \omega[k]$ or especially $\omega[k]$ what can arise in general case. Let us denote for any set $v[k]$ by $G(v[k]) := \{(v_i, u_i)\}_{i=1}^{k}$ the Gramian matrix and for any matrix $A$ by $A^*$ let us denote the pseudo-inverse matrix of Moore-Penrose (F. R. Gantmacher [5], p. 33 and J. Stoer, R. Bulirsch [11], p. 220). Now, the remedy for this problem is the following theorem.

**GENERALIZED THEOREM ON THE LMS APPROXIMATION.**

If $U_F$ is the unitary space with the norm $|| \cdot ||$ determined by the inner product $(,)$ as $|| x || = \sqrt{(x, x)}$, $v[k] := (u_k)_k^{i}$ is some set of elements from $U_F$, such that $\dim \text{span } v[k] \leq k$, and $c[k] := G^* (v[k]) (v[k], u) := G^* (v[k]) (v_i, u)_i$, then there exists exactly one LMS approximation $v_k := c[k]^{i} v[i]$ of element $u \in U_F$ in the linear space $\text{span } v[k]$ and the LMS error is $||u - v_k|| = \sqrt{||u||^2 - ||v_k||^2}$. 
Proof. The existence of the exactly one element \(v_k\) results from the fact that for each matrix exactly one pseudoinverse matrix exists (F. R. Gantmacher [5], p. 32). If \(\dim \text{span} v_{li} = 0\) then of course \(v_k = 0\). So, let us consider the case when \(\dim \text{span} v_{li} > 0\). From well known features of the LMS approximation, it will be enough to show that \((u - v_k) \perp \text{span} v_{li}\), which means that \((u - v_k, v_{li}) = 0 \in F^k\). To do this, let us notice that

\[
(u - v_k, v_{li}) = (u - (G^+(v_{li})(v_{li}, u))^T v_{li}, v_{li}) =
\]

\[
= (u, v_{li}) - ((G^+(v_{li})(v_{li}, u))^T v_{li}, v_{li}) =
\]

\[
= (u, v_{li}) - ((G^+(v_{li})(v_{li}, u))^T v_{li}, v_{li})^T =
\]

\[
= (u, v_{li}) - G^+(v_{li}) G(v_{li})(u, v_{li}) =
\]

\[
= (u, v_{li}) - G(v_{li}) G(v_{li})(u, v_{li}) =
\]

\[
= (u, v_{li}) - G(v_{li}) G(v_{li})(u, v_{li}).
\]

As we know, the Gramian matrix \(G(v_{li})\) is invertible if and only if the system \(v_{li}\) is linearly independent (V. A. Ilyin, E. G. Poznayak [6], p. 216), but the theorem assumptions do not make this certain, so we have \(\dim \text{span} v_{li} \leq k\). Then, we can choose such elements of the set \(v_{li}\) which form the base \(\omega_{li} = (\omega_{li})^T\) of the linear space \(\text{span} v_{li}\). Let \(A \in \mathbf{M}_{F \times k}\) be such matrix that \(v_{li} = A \omega_{li}\). Let us denote \(G = \omega_{li}^T A^+ \omega_{li}\), then from fundamental features of the pseudoinverse matrices (J. Stoer, R. Bulirsch [11], p. 220) we get

\[
G(v_{li}) = A G(A \omega_{li}) A^+ G^{-1}(A \omega_{li}) A^+ =
\]

\[
= AG(A \omega_{li}) A^+ G^{-1}(A \omega_{li}) A^+.
\]

Because the rank of the matrix \(A\) denoted by \(\text{rank} A = l\), then let \(B \in \mathbf{M}_{F \times k}\) and \(C \in \mathbf{M}_{F \times l}\) be decomposition matrices of the matrix \(A\) such that \(\text{rank} B = \text{rank} C\) and \(A = BC\). It can be proved (F. R. Gantmacher [5], p. 33) that \(A^+ = C(C^+)^{-1}(B^+)^{-1}B^+\) and from here, assuming \(B = A\) and \(C = I_{[l, l]}\), we get

\[
A^+ A = I_{[l, l]} (I_{[l, l]} I_{[l, l]})^{-1} (A^+ A)^{-1} A^+ A =
\]

\[
= (A^+ A)^{-1} A^+ A = I_{[l, l]},
\]

which means that the product

\[
G(v_{li}) G(v_{li}) = A A^+ A G(A \omega_{li}) A^+ =
\]

\[
= A G(A \omega_{li}) A^+ = G(v_{li}).
\]

Presented relations and the pseudo-inverse matrices features (J. Stoer, R. Bulirsch [11], p. 220) make the following conditions be fulfilled:

\[
G(v_{li}) G(v_{li}) = A A^+ A G(A \omega_{li}) A^+ =
\]

\[
= A G(A \omega_{li}) A^+ = G(v_{li}).
\]

\[
G(v_{li}) G(v_{li}) = (A^+)^* G^{-1}(A \omega_{li}) A^+ A^+ =
\]

\[
= (A^+)^* G^{-1}(A \omega_{li}) A^+ =
\]

\[
= (AA^*) A^+ = A A^+ = (G(v_{li}))^T.
\]

Finally, the LMS error formula we can notice the Generalized Pythagoras Theorem what ends the proof. 

This is a very strong and novel theorem on the LMS approximation.

7. Obtained Results

Some results obtained by presented theory for the linear time-invariant systems (R. Wojciechowski [12]) are shown in fig. 1, for the static system (A. Albrecht [11]) are shown in fig. 2 and for the nonlinear time-invariant system (G. Ciesielski [4]) are shown in fig. 3.

---

Fig. 1. The step responses of the first order inertial system (a) and oscillatory system (b) before ■ and after ■ the optimal past- and pre-correction with the use of 8 Laguerre Functions (R. Wojciechowski [12]).
Fig. 2. The error functions of the pressure measurements system with the use of the DRUCK Sensor PDCR 901 and some classic past-correction (a) and after the optimal past-correction (b) with the use of the Cubic B-Splines tensor product for the breakpoints \( p_{\text{appr}} = [100:500/20:600] \) kPa and \( t_{\text{appr}} = [-20:70/7:50] ^\circ C \) (A. Albrecht [1]).

Fig. 3. The step responses of the Danaide (orifice gauging tank) volumetric flow sensor \( \circ \) together with its optimal model \( \bullet \) (a) and its step responses after optimal past-correction \( \bullet \) for required system \( \circ \) (b) with the use of 27 Wiener Operators (G. Ciesielski [4]).

References


Differences between Theoretical and Measured Spectrum in Systems Employing a Spread-Spectrum Clock for EMI Reduction Purposes

Fabio Pareschi†§, Gianluca Setti†§, Riccardo Rovatti‡§ and Giovanni Frattini¶

† ENDIF – Engineering Department, University of Ferrara
Via Saragat, 1 – 44100 Ferrara, Italy
‡ DEIS – Department of Electronics, Computer Sciences and Systems, University of Bologna
Viale Risorgimento, 2 – 40136 Bologna
§ Research center “ARCES” – University of Bologna
Via Toffano, 2 – 40125 Bologna
¶ National Semiconductor, Italy
Email: {fabio.pareschi,gianluca.setti}@unife.it, rovatti@deis.unibo.it, giovanni.frattini@nsc.com

Abstract—Many digital systems employ a spread spectrum clock technique for an inherent reduction of the Electro-Magnetic Interference. Spread Spectrum clocking consists in a proper modulation of the system clock, thus reshaping the power density spectrum of all synchronized digital signal. The aim of this paper is to show that, depending on the specific setting, very large differences between the theoretically computed power density spectrum and the measured one arise. This is an important issue to take into consideration when a Spread Spectrum system is optimized for EMI reduction.

1. Introduction

The reduction of the Electro-Magnetic Interference (EMI) in electronic system is an inherent problem in all modern digital equipment. In fact digital signals, due to their sharp edges and their synchronization with a periodic clock waveform, are preeminent sources of interference, since they give arise to a large amount of interfering power in a narrow-band frequency range.

This point of view is coherent with many regulations [1, 2] which link compliance, or Electro-Magnetic Compatibility (EMC), with the ability to constrain the interfering power density spectrum (PDS) within a prescribed mask. The reason why EMI are related to the shape of the interfering signal PDS, in particular to its peak value, is that in the coupling process between EMI sources and third-party nearby circuits (EMI victims), these latter can be usually modeled as a number of narrow-bank filters, i.e. a victim is sensitive only in a few frequency ranges. The worst case scenario is when the PDS of an interfering signal is composed by few components with a high power level (as for a synchronous digital system) and the largest of them is exactly located in one of the victim sensitivity frequency ranges. In this case all its power is transferred to the victim, potentially yielding to its complete failure.

It is worth noticing that in addition to common solutions to increase the EMC which are based on a-posteriori methodologies (such as the adoption of filters, shielded cables and filtered connectors) which aim at reducing the coupling between EMI sources and EMI victims, some design-time solutions, usually known as spread spectrum clocking techniques, can be adopted. Referring to [3], spread spectrum clocking is defined as “a technique to reduce the emission from all signals synchronized with a clock” and, roughly speaking, consists of introducing a controlled jitter in the reference clock, thus avoiding a perfect periodicity of all the synchronized signals. This gives rise to additional components in the PDS, while, at the same time, it lowers the power of the already present ones, with a positive effect in EMI reduction.

In this paper we consider the differences between the theoretical PDS of a spread spectrum system and the spectrum measured according to EMC regulations. This is particularly import since, despite the fact that most common way to design a spread spectrum system is to reshape the theoretical PDS [4, 5], regulations require that measurements are taken in a prescribed setting with an EMI receiver, (which is an analog Spectrum Analyzer): we can show that the theoretical spectrum and the measured one match only for particular cases.

The organization of the paper can be summarized as follows. The aim of the Section 2 is to provide a brief theoretical background. Here, we first introduce the working principle of the analog spectrum analyzer. Then, we consider a Spread Spectrum clock system based on the Frequency Modulation (FM) of the clock with a sinusoidal waveform as driving signal. Despite the fact that this approach is not commonly used in real systems, where a clock FM with a triangular waveform [6, 7], a more complex and patented periodic waveform [3] or a Pulse-Amplitude Modulated (PAM) waveform [8, 9] is used as driving signal, the sinusoidal case is interesting since its theoretical PDS can be expressed in a simple, closed form. In Section 3 this PDS is compared with the measured spectrum, and despite being the latter very complex to compute, we will be able to easily explain why and under which circumstances differences exist between the two. Finally in Section 4 we draw the conclusion.

2. Mathematical Background

2.1. Working Principle of a Spectrum Analyzer

An analog Spectrum Analyzer is based on a superheterodyne receiver as schematized in Figure 1(a). The input sig-
nal is mixed with a pure tone generated by a local oscillator to shift it to a lower frequency range and then filtered by narrow band-pass filter (whose bandwidth is called Resolution BandWidth, RBW). The resulting output signal is then demodulated by an envelope detector, low-pass filtered (the VBW) and its power level is measured by a peak detector. Note that this must be distinguished from the digital spectrum analyzer, or Digital Signal Analyzer, which is based on the real-time Fourier transform of the sampled input signal.

By changing the local oscillator frequency, it is possible to tune a different frequency band on the RBW filter, so that to measure the spectrum of the input signal, it is enough to tune the system to all the frequency of interest.

The EMI regulations allows the use of both the quasi-peak detector and the positive peak detector. For the sake of simplicity, we consider here only the latter, which estimate the power of the tuned signal as the power of a pure sinusoidal tone whose amplitude is equal to the maximum amplitude detected while the system is tuned to a given frequency.

Note that the same measurement results can be achieved by a system like the one in Figure 1(b), where the signal is tuned by properly setting the central frequency of the RBW filter. The VBW filter is not considered since under the common assumption of VBW > RBW its effect can be neglected. This will be more clear at the end of Section 3. In the following, we will always refer to this simplified diagram.

A detailed survey on this instrument can be found in [10], where the RBW filter is assumed as a four-pole system, with nearly-Gaussian transfer function

$$|H(f; f_0, \text{RBW})|^2 = \frac{1}{\left(1 + \left(\frac{f - f_0}{\nu_0 \text{RBW}}\right)^2\right)^4}$$  (1)

where $f_0$ is the center frequency and $\nu_0 = \frac{1}{2\sqrt{2/\text{RBW}^2 - 1}}$ and the RBW is defined as the $-3$ dB filter bandwidth.

### 2.2. Sinusoidal FM

Let us consider here an FM clock signal which, referring to the first harmonic only, can be written as

$$s(t) = A \cos \left(2\pi f_c t + 2\pi Df \int_{-\infty}^{t} \xi(\tau)d\tau\right)$$

where $-1 < \xi(t) < 1$ is the normalized driving signal, and $Df$ the frequency deviation. It is worth stressing that considering only the fundamental tone of a timing signal is a standard practice for EMI measurement purposes, since it is the harmonic giving rise to highest peaks in the spectrum, and which is therefore responsible for generating the most severe EMI components.

When using a sinusoidal waveform as driving signal, i.e. $\xi(t) = \cos (2\pi f_m t)$, we get $s(t) = A \cos (2\pi f_c t + m \sin (2\pi f_m t))$, with $m = Df / f_m$ is known as modulation index. The PDS $S(f)$ of $s(t)$ is a discrete spectrum with components at all frequencies $f_c \pm k f_m, \forall k \in \mathbb{N}$, and can be written using the first kind Bessel Function [11]

$$S(f) = \frac{A^2}{2} \sum_{k=-\infty}^{\infty} J_1^2(m) \delta \left(f - f_c - \frac{k}{m} Df\right)$$  (2)

### 3. Comparison between Theoretical Spectrum and Measured One

The theoretical spectrum of a sinusoidal FM measured by means of a band-pass filter is given by the convolution of (2) with the filter transfer function, which is given by (1), so that we get

$$S^{\text{th}}(f) = \frac{A^2}{2} \sum_{k=-\infty}^{\infty} J_1^2(m) \left|H\left(f; f_c + \frac{k}{m} Df, \text{RBW}\right)\right|^2$$  (3)

The spectra achieved for $f_c = 1$ MHz, $Df = 50$ KHz, RBW = 3 KHz and different values of $m$ are shown in Figures 2(a), 2(b) and 2(c). In all Figures the 0 dBc reference level is the power level of the first harmonic of the unmodulated clock, i.e. is $A^2 / 2$. Figures 2(d), 2(e) and 2(f) show, for the same cases, the spectrum measured with an HP8563E Spectrum Analyzer. Similarly, the 0 dBm reference level is the power level of the first harmonic of unmodulated clock.

By comparing the two sets of Figures, some remarks can be made.

- In the case when $m$ is low (Figures 2(a) and 2(d), where $m = 3$) the match between the spectrum expected according to (3) and the measured one is almost perfect.
- When $m$ assumes intermediate values (as in Figures 2(b) and 2(e), where $m = 25$) despite a good matching between the shape of the expected and the measured spectrum, the measured one has a much higher level then the expected one. In this example, the difference is approximately 5 dB.

![Diagram](image-url)
Figure 2: Comparison between the power spectrum expected from Equation (3) and the one measured from an HP8563E Spectrum Analyzer. (a) Expected spectrum for $m = 3$; (b) expected spectrum for $m = 25$; (c) expected spectrum for $m = 200$; (d) measured spectrum for $m = 3$; (e) measured spectrum for $m = 25$; and (f) measured spectrum for $m = 200$. In all cases, it is $f_c = 1$ MHz, $Df = 50$ KHz, RBW = 3 KHz.

- For very large values of $m$ (Figures 2(c) and 2(f), where $m = 200$) the difference between the expected and the measured spectrum is increasing (about 10 dB in this case). Furthermore, the shapes of the two spectra do not match anymore, as in the measured one a saturation effect seems to arise. Note also that the peak level of the measured spectrum is almost equal to the unmodulated signal power.

Even if, for the sake of simplicity, we have considered the sinusoidal case, comments similar to those above can be made for any modulating signal, including all those which are used in practical application (triangular [6, 7], patented [3] and PAM [8, 9]). These observations have a serious impact on the design of a spread spectrum clocking system for EMI reduction.

In fact, to the best of our knowledge, the way which is almost always adopted to design a spread spectrum system is to optimize its performance according to the theoretical PDS, i.e. according to the spectrum one could expect from Equation (3). In this case, as it is clear from the above example, the highest EMI reduction is given when $m$ has intermediate or large values, with more than 10 dB reduction in the peak level in the power spectrum with respect to the unmodulated case. Like in the case of the example, most of the actual spread spectrum systems are designed [3, 6, 8] to operate with reasonably very large values of $m$, as the performance of a spread spectrum system is optimized for $m \rightarrow \infty$.

What we want to show with the above example is that, as $m$ increases, the differences between the expected and the measured spectrum also increase. More important, when $m$ assumes very large values (as in the case of Figure 2(f)), the EMI reduction with respect to the unmodulated case may become negligible, i.e. cases exist where a spread spectrum system would give no advantages in terms of EMI reduction.

Despite the fact that a formal mathematical explanation of this effect would be extremely complex, we can give a simple, intuitive one. It is known that the positive peak detector may overestimate the power of the filtered signal. It is common to say that it gives results higher than the quasi peak detector, which gives results higher than the average detector. Note however that neither the quasi peak nor the average detector are usually present in a standard spectrum analyzer. The overestimation of the positive peak detector depends on the features of the analyzed signal. In the considered case, it is clear that, the higher is $m$, the higher the overestimation.

Let us consider a slow sinusoidal modulation, i.e. the case where $m$ is very high or, equivalently, $f_{m}$ very low. Let us also refer to the block diagram of Figure 1(b). We can model the system as slowly changing its instantaneous frequency from $f_c - Df$ to $f_c + Df$. When the time required for sweeping from $f_0 - RBW/2$ to $f_0 + RBW/2$ (i.e. the time to completely cross the RBW filter bandwidth) is long enough, the Spectrum Analyzer sees the signal as virtually “unmodulated” and the amplitude memorized by the peak detector is almost equal to the amplitude memorized in the
From a mathematical point of view, we can consider the simple case in which \( f_0 = f_c \). Let us also consider that the RBW filter is an ideal rectangular band-pass filter. In the theoretical spectrum, only components with frequency \( f_c - \text{RBW}/2 \leq f \leq f_c + \text{RBW}/2 \) will contribute to the power at \( f_c \).

\[
S^{(th)}(f_c) = \frac{A^2}{2} \sum_k J_k^2(m), \quad |k| \leq \frac{1}{2} m \frac{\text{RBW}}{Df}
\]  

(4)

When using the model of Figure 1(b), it is easy to understand that the signal at the input of the peak detector can be computed as the absolute value of the complex envelope \( \hat{s}(t) \) of the signal \( s(t) \) being filtered by the base-band equivalent of the RBW filter, which is a low-pass filter with bandwidth \( \text{RBW}/2 \). Note that it is now clear that the condition for neglecting the VBW filter is \( \text{VBW} > \text{RBW}/2 \), which is usually satisfied.

Since the input signal can be written as

\[
\hat{s}(t) = \Re \left( A e^{j2\pi f_m t} + j m \sin(2\pi f_m t) \right)
\]

its complex envelope is [11]

\[
\hat{s}(t) = A e^{j2\pi f_m t} = \sum_{k=-\infty}^{\infty} J_k(m) e^{j2\pi f_m t}
\]

Under the assumption of a rectangular low-pass filter with bandwidth \( \text{RBW}/2 \), the filtered complex envelope \( \hat{s}_H(t) \) is obtained by considering the only few components

\[
\hat{s}_H(t) = A \sum_k J_k(m) e^{j2\pi f_m t}, \quad |k| \leq \frac{1}{2} m \frac{\text{RBW}}{Df}
\]

The signal power is estimate by the positive peak detector as \( \frac{1}{T} (\text{max}|\hat{s}_H(t)|)^2 \). Note that independently on the number of considered components, \( |\hat{s}_H(t)| \) is periodic with a maximum for \( t = \frac{T}{2} \pm \frac{m}{2} \); the power measured in \( f_c \) by the positive peak detector is

\[
S^{(\text{meas})}(f_c) = \frac{A^2}{2} \sum_k J_k^2(m), \quad |k| \leq \frac{1}{2} m \frac{\text{RBW}}{Df}
\]  

(5)

By comparing Equation (4) with Equation (5) we can determine when the measured spectrum differs from the theoretical PDS. Despite the fact that the two above expression are very difficult to handle with, it is possible to see that one always have \( S^{(\text{th})}(f_c) \leq S^{(\text{meas})}(f_c) \). Furthermore, the only condition for which we can ensure a matching between the two spectra is \( \frac{1}{2} m \frac{\text{RBW}}{Df} < 1 \). In this case, both the sums of Equation (4) and Equation (5) result in

\[
S^{(\text{th})}(f_c) = S^{(\text{meas})}(f_c) = \frac{A^2 J_1^2(m)}{2}
\]

In conclusion, based on what computed for the power spectrum at frequency \( f_c \), we can say that in a fast modulation, more precisely when \( m < \frac{2 Df}{\text{RBW}} \) (which means \( f_m > \text{RBW}/2 \)) we can be sure that the measured spectrum matches the theoretical PDS. On the contrary, when \( m > \frac{2 Df}{\text{RBW}} \) we can expect a power level of the measured spectrum higher than the one computed through the theoretical PDS.

4. Conclusion

In this paper we have compared the theoretical PDS of a sinusoidal Frequency Modulation with the spectrum achieved by a measurement with a Spectrum Analyzer. By means of a very simple modeling of this instrument, we have concluded that the two spectra match only when the modulation index \( m < \frac{2 Df}{\text{RBW}} \). Otherwise, the measured power level is higher than the expected one from the theoretical PDS. This has important consequences in spread spectrum systems based on the frequency modulation of the clock, since the EMI reduction measured according to EMC regulations may be much smaller than the expected one when \( m \) is too large.

References


Rigorous parameter estimation for noisy mixed-effects models

Alexander Danis † Andrew Hooker‡ and Warwick Tucker‡

†Dept. of Mathematics, Uppsala University
Box 480, 751 06 Uppsala, Sweden
‡Department of Pharmaceutical Biosciences, Uppsala University
Box 591, 751 24 Uppsala, Sweden
Email: alexander.danis@math.uu.se, andrew.hooker@farmbio.uu.se, warwick.tucker@math.uu.se

Abstract—We describe how constraint propagation techniques can be used to reliably reconstruct model parameters from noisy data. The main algorithm combines a branch and bound procedure with a data inflation step; it is robust and insensitive to noise. The set–valued results are transformed into point clouds, after which statistical properties can be retrieved. We apply the presented method to a mixed-effects model.

1. Introduction

Finitely parameterized mathematical models are used to describe, explain, summarize, and predict the behavior of physical, biological, and economical systems. A parameter estimation problem is a problem of finding a set of parameter values that makes the model function fit the experimental data. Incomplete approaches to this problem search for one solution in parameter space; complete approaches search for all. In this article, we describe a complete approach—a general-purpose solution strategy based on set–valued computations and global search algorithms that operate reliably under noise. We begin by describing the basic components of the solver which involves set–valued computations, directed acyclic graphs, constraint propagation, and data inflation. We end the article with an example demonstrating the usefulness of our approach.

2. Interval analysis

The foundation of most computer-aided proofs dealing with continuous problems is the ability to compute with set-valued functions. This not only allows for all rounding errors to be taken into account, but—more importantly—all discretization errors too. Here, we will briefly describe the fundamentals of interval analysis. For a concise reference on this topic, see e.g. [Moo66, Neu90].

Let \( \mathbb{R} \) denote the set of closed intervals. For any element \( a \in \mathbb{R} \), we adopt the notation \( a = [a, b] \), where \( a, b \in \mathbb{R} \). If \( \star \) is one of the operators \(+, -, \times, \div\), we define the arithmetic on elements of \( \mathbb{R} \) by

\[
a \star b = [a \star b; a \in a, b \in b],
\]

except that \( a \div b \) is undefined if \( 0 \in b \). Working exclusively with closed intervals, the resulting interval can be expressed in terms of the endpoints of the operands. Note that the identities (1) reduce to ordinary real arithmetic when the intervals are thin, i.e., when \( a = \overline{a} \) and \( b = \overline{b} \).

A key feature of interval arithmetic is that it is inclusion monotonic, i.e., if \( a \subseteq x \), and \( b \subseteq y \), then

\[
a \star b \subseteq x \star y,
\]

where we demand that \( 0 \notin y \) for division.

One of the main reasons for passing to interval arithmetic is that this approach provides a simple way of enclosing the range of a function \( f \), defined by \( R(f; D) := \{f(x) : x \in D\} \). Except for the most trivial cases, classical mathematics provides few tools to accurately bound the range of a function. To achieve this latter goal, we extend the real functions to interval functions which take and return intervals rather than real numbers. Based on (1) we extend rational functions to their interval versions by simply substituting all occurrences of the real variable \( x \) with the interval variable \( x \) (and the real arithmetic operators with their interval counterparts). This produces a rational interval function \( F(x)\), called the natural interval extension of \( f \). As long as no singularities are encountered, we have the inclusion

\[
R(f; x) \subseteq F(x),
\]

by property (2). In fact, this type of range enclosure can be achieved for any reasonable function. A higher-dimensional function \( f : \mathbb{R}^n \rightarrow \mathbb{R} \) can be extended to an interval function \( F : \mathbb{R}^n \rightarrow \mathbb{R} \) in a similar manner.

There exist several open source programming packages for interval analysis, as well as commercial products.

We will now illustrate the use of interval techniques, with a special emphasis on parameter estimation problems.

Example 2.1 Consider the model \( y = f(x; p) = xe^{-ps} \), together with the (exact) data point \((x, y) = (2, 1)\), and search region \( p = [0, 1] \). A straight-forward interval evaluation of the model function yields:

\[
f(x; p) = f(2; [0, 1]) = 2e^{-(0.1\times 2)} = [2e^{-2}, 2].
\]

This constrains (at \( x = 2 \)) the value of the model function \((y = 1)\) to belong to the interval \([2e^{-2}, 2] \approx [0.27, 2] \), which it does. Had we chosen \( p = [1, 2] \) as our search
space, we would obtain an inconsistency: $1 \notin f(2, [1, 2]) = [2e^{-4}, 2e^{-2}] \approx [0.03, 0.27]$. This would allow us to discard the entire set $p$.

This example illustrates how a divide and conquer approach can be devised. Starting from a large search space $p$, we adaptively bisect $p$ into smaller subsets, many of which we can discard via inconsistency checks.

### 3. Noise and data inflation

Given a model function $y = f(x; p)$ together with a finite set of noisy data $(x_1, \hat{y}_1), \ldots, (x_N, \hat{y}_N)$, we will attempt to find parameters that make the model consistent with the data, i.e., we want to find the set

$$S = \{ p \in p: f(x_i; p) = \hat{y}_i \text{ for all } i = 1, \ldots, N \}. \quad (5)$$

In general, this set will be empty, indicating the presence of noise in the data. In order to improve matters, we inflate each data value into an interval. This can be done in several ways; assigning a width roughly proportional to the value $\hat{y}_i$ is a good heuristic choice in many situations:

$$\hat{y}_i \mapsto y_i = \hat{y}_i(1 + \alpha[-1, +1]) + \beta[-1, +1]. \quad (6)$$

Here $\alpha$ is a scaling factor, and $\beta$ is a threshold factor, necessary for situations when $\hat{y}_i$ is very small.

The new requirement for consistency is now formulated in terms of more robust inclusion conditions:

$$S = \{ p \in p: f(x_i; p) \in y_i \text{ for all } i = 1, \ldots, N \}. \quad (7)$$

With no data inflation, this reduces to (5). Gradually increasing $\alpha$ (and/or $\beta$) will eventually produce a non-empty set of consistent parameters $S$.

Given a partition $P = \{ p \}_i$ of the search space $p = p_1 \cup \cdots \cup p_k$, the consistent parameters can be enclosed via $S \subseteq S \subseteq S$, where

$$S = \{ p \in P: f(x_i; p) \subseteq y_i \text{ for all } i = 1, \ldots, N \},$$

$$S = \{ p \in P: f(x_i; p) \cap y_i \neq \emptyset \text{ for all } i = 1, \ldots, N \}.$$

The next example displays how interval-valued data can be contracted, using constraints from both the model function and the search space.

**Example 3.1** Repeating the calculations from Example 2.1, with $p = [0, 1]$, but with the data $(x, y) = (2, [1, 3])$, we can contract the data range according to

$$y \mapsto y \cap f(x; p) = [1, 3] \cap [2e^{-2}, 2] = [1, 2].$$

### 4. Constraint propagation for pedestrians

In this section, we will outline the main ingredients of our parameter estimation procedure. As mentioned in Section 1, our method is global. As such, it attempts to find all parameters consistent with the data. Of course, when the data is noisy, there are no consistent parameters, in general. This forces us to inflate the data – a process that compensates for the loss of information caused by the noise. Once sufficiently inflated, the set of consistent parameters is non-empty and bounded. We shrink the bounding set (and the inflated data) by constraint propagation techniques. To be fully effective, these techniques require that the model function be represented in a special form.

#### 4.1. The DAG representation

We use a directed acyclic graph (DAG) representation of the model function to automate constraint propagations. This representation captures the natural way of decomposing a (possibly complicated) function into its basic building blocks. The graph nodes represent variables, constants or simple functions, while the edges represent dependencies between them.

**Example 4.1** Returning to the model function of Example 2.1, $f(x; p) = xe^{-px}$, it can be decomposed into the following code list:

- $n_1 = x$
- $n_2 = p$
- $n_3 = n_1 \times n_2$
- $n_4 = -n_3$
- $n_5 = e^{n_4}$
- $n_6 = n_1 \times n_5$.

This list is equivalent to the DAG illustrated in Figure 1.

The DAG representation is used to obtain numeric and symbolic information about various mathematical objects, such as derivatives, slopes, mean value forms, linear or convex relaxations (over- and under-estimators), and convexity information. Representations of equations and inequalities can be included for the purpose of accelerating the constraint propagations. Information on these matters can be found in [GW08, SN05].
4.2. Constraint propagation on DAGs

To each elementary mathematical operation one associates two operations, forward and a backward operations. The forward operator evaluates the range based on the range of its arguments and intersect it with the current range. The backward operator evaluates the ranges of its predecessors and intersect it with their current ranges.

Example 4.2 Once more, we will use the model function of Example 2.1, \( f(x; p) = x e^{-px} \). As we saw in Example 4.1, it can be decomposed into a code list as well as a DAG. We will now show how we can use these objects to propagate constraints from data to the parameter. Moving backwards in the code list of Example 4.1, starting from \((x, y) = (n_1, n_6)\), and ending in \( p = n_2 \), we obtain a new code list:

\[
\begin{align*}
n_3 &= n_6 \div n_1 \\
n_4 &= \log n_3 \\
n_5 &= -n_4 \\
n_2 &= n_3 \div n_1.
\end{align*}
\]

This list is equivalent to the DAG illustrated in Figure 2.

Thus, viewing a function in terms of its code list or DAG allows us to compute its formal inverses, without knowing the formulae for these. All we need is the code list for \( f \). Traversing the list backward produces the desired information. This is extremely useful for parameter estimation problems, as we illustrate in the following example.

Example 4.3 Given the model function \( y = f(x; p) = x e^{-px} \), together with the data \((x, y) = (2, 1)\), we can generate, and evaluate the code list of Example 4.2:

\[
\begin{align*}
n_3 &= n_6 \div n_1 = 1 \div 2 \\
n_4 &= \log n_3 = \log \frac{1}{2} \\
n_5 &= -n_4 = -\log 2 \\
n_2 &= n_3 \div n_1 = \frac{1}{2} \log 2 \approx 0.34657359.
\end{align*}
\]

The conclusion is that the only parameter that corresponds to the data \((x, y) = (2, 1)\) is \( p = \frac{1}{2} \log 2 \). Of course, for this simple example, we can find the explicit inverse: \( p = \frac{1}{2} \log \frac{1}{2} \), which gives the sought result. The point is, however, that we never use this formula; we only use the code list of \( f \).

Continuing Example 3.1, where we have interval-valued data \((x, y)\), and a parameter domain \( p \) to examine, we can combine the forward and backward sweeps to contract both \( y \) and \( p \).

Example 4.4 Again, we work on the model function \( y = f(x; p) = x e^{-px} \), but now with the data \((x, y) = (2, [1, 3])\), together with the parameter domain \( p = [0, 1] \). The forward sweep, performed in Example 3.1, contracts the interval data to \( y = [1, 2] \). Performing a backward sweep, as in Example 4.3, contracts the interval parameter to \( p = [0, \frac{1}{2} \log 2] \):

\[
\begin{align*}
n_3 &= n_6 \div n_1 = [1, 2] \div [0, 1] = [\frac{1}{2}, 1] \\
n_4 &= \log n_3 = \log [\frac{1}{2}, 1] = [\log 2, 0] \\
n_5 &= -n_4 = [0, \log 2] \\
n_2 &= n_3 \div n_1 = [0, \log 2] \approx [0, 0.34657359].
\end{align*}
\]

Note that, in one forward/backward sweep, we managed to exclude over 65% of the parameter domain, at the same time reducing the data uncertainty by 50%.

In most cases, the described constraint propagation techniques do not result in a complete contraction to the optimal state. Rather, a stage is reached where no further contraction can be obtained, even though there are inconsistencies present. In order to proceed, some type of partitioning must be employed. The partitioning can be performed at any node of the DAG. Once a node has been selected for partitioning, its domain is split, resulting in two new DAGs, differing only in the domain of the split node. Each DAG is updated through forward/backward sweeps, possibly generating more contraction.

4.3. Data gridding

In order to extract traditional statistics from the set-valued results, we discretize the set \( \mathbf{S} \) into a collection of points. Recall the the outcome of our parameter estimation is a collection of boxes \( p_1, \ldots, p_m \), whose union \( \mathbf{S} \) may or may not be a connected set. We form the hull of this collection by taking the smallest box \( \mathbf{p} \) that contains \( \mathbf{S} \). Next, we introduce \( m \) equally spaced nodes along each side of \( \mathbf{p} \). This defines a grid of size \( m^d \) where \( d \) is the dimension of \( \mathbf{p} \). From this grid, we discard all nodes that are not sufficiently close to the set \( \mathbf{S} \). This leaves us with a set of points amenable to statistical tests.

5. Methods and examples

We demonstrate our method on a mixed–effects model. A mixed–effects model is a model that includes a mixture
of fixed and random factors. In the current setting, we will consider a model function of the form \( f(t; \tilde{p}) \), where \( t \) denotes time, and \( \tilde{p} = (p_1, p_2, p_3) \) is a three-dimensional parameter vector. A population parameter is a parameter that is shared among every member of the population. An individual parameter is a parameter that is unique to each individual. In what follows, we let \( p_1 \) be an individual parameter, whereas \( p_2 \) and \( p_3 \) act as population parameters. Thus \( p_1 \) corresponds to a random factor (sampled from some underlying distribution), whereas \( p_2 \) and \( p_3 \) correspond to fixed factors.

5.1. Methods

Starting from a given parameter vector \( \tilde{p} = (p_1, p_2, p_3) \), we perturb the individual parameter according to

\[
p_i^* = p_i + \eta_i \quad \text{where} \quad \eta_i \sim N(0, \sigma^2) \quad (i = 1, \ldots, N_p).
\]

Here \( N(a, b) \) denotes the normal distribution with mean \( a \) and (positive) variance \( b \). This produces \( N_p \) different parameter vectors \( \tilde{p}^1, \ldots, \tilde{p}^N_p \) (each corresponding to a different subject), where \( \tilde{p}^j = (p_1^j, p_2^j, p_3^j) \). For each of the \( N_p \) subjects, we generate exact data \( y_{ij}^* = f(t_i^j; \tilde{p}^j) \) for \( j = 1, \ldots, N_d \). Next, the exact data is perturbed, with maximal intensity \( \epsilon > 0 \), according to

\[
\hat{y}_{ij} = y_{ij}(1 + \theta_{ij}) \quad \text{where} \quad \theta_{ij} \sim U(-\epsilon, \epsilon).
\]

Here \( U(a, b) \) denotes the uniform distribution on the interval \([a, b]\). This produces the data set \((t_j, \hat{y}_{ij})\), which, together with the model function, is all information we have.

In order to find the typical performance of our method, we repeat the entire estimation process \( N_t = 200 \) times, and report the average results.

5.2. Example

The following model is used to study the growth of orange tree trunks, see [LB90, DS98].

**Example 5.1** Consider the following function

\[
f(t; \tilde{p}) = \frac{p_1}{1 + p_2 e^{p_3 t}}.
\]

For this specific example, we will use \( N_p \in \{10, 25, 50\} \) subjects, sampled at \( N_d = 10 \) data sites, evenly spaced within \([100, 1600] \). The parameter values corresponding to the mean population are \( \tilde{p} = (191.84, 8.153, -0.0029) \); the perturbation parameters are \( \sigma = 20 \) and \( \epsilon \in \{0.1, 0.2, 0.3\} \). We perform the constraint propagation over the parameter region \( p = (\{10, 300\}, [0, 9], [-1, 0]) \). The results of this procedure are illustrated in Figure 3 and in Table 1.

6. Conclusion

We have described a new approach to solve parameter estimation problems for noisy data. It is a deterministic global search method based on relaxation methods (via data inflation) and the use of set-valued constraint propagation.
Evolution of Density of States for Delay Blood Cell Production Model

Pawel J. Mitkowski† and Maciej J. Ogorzalek‡
†PhD student Faculty of EAIIE, AGH
‡Department for Information Technology,
Faculty of Physics, Astronomy and Applied Computer Science,
Jagiellonian University, Krakow, Poland
Email: pawel.mitkowski@agh.edu.pl, Maciej.Ogorzalek@uj.edu.pl

Abstract—In the paper we analyse delay model for the blood cell production (The Lasota equation). The main goal is to study the evolution of the density of states. Frobenius-Perron operator describing the density transformation is presented. Numerical calculations are constructed to analyse the approximation of evolving density functions. Initial density functions are determined using inverse cumulative distribution functions. For selected parameters of the equation simulations shows that chosen different initial densities converge to the same invariant density, what indicates the possibility of existence of an invariant measure.

1. Introduction

We are considering the following delay model for blood cells production:

\[
\frac{dN(t)}{dt} = -\alpha \cdot N(t) + (\rho \cdot N(t - \tau))^s \cdot e^{-\gamma N(t - \tau)}. \tag{1}
\]

Equation (1) was formulated by A. Lasota in [5]. Its biological interpretations are related with earlier model for red blood cells system called the Lasota-Wazewska equation, which is age structured equation with delay feedback [23]. \(N(t)\) represents amount of red blood cells (erythrocytes) in blood circulation, \(\alpha, \rho, \gamma\) are constants that have biological meaning (for details see [23] or [22]). \(\tau\) is a delay time interpreting as a time of maturation of erythrocytes and \(s\) is a power in nonlinearity describing the production rate of blood cells. Some informations about the power dependence of production rate of blood cells can be found e.g. in [22], [13]. Because of the non-monotone character of this nonlinearity equation (1) exhibits complicated dynamics. The influence of such nonlinearities on dynamics were studied by many authors e.g. [21], [6], [10], [19], [3], [11]. The nonlinearities with non-monotone character were used in some other delay models in biology and medicine e.g. in Nicholson’s blowflies equation [16], [5], [4] or Mackey-Glass delay model for white blood cell production [14].

In [5] A. Lasota was analysing chaotic behaviour of biological systems, using approach of ergodic theory. The goal was to investigate existence of continuous invariant and ergodic measures in theoretical models of biological systems. From the Birkhoff individual ergodic theorem it follows, that almost all trajectories are complicated if such measure exists see [7]. With reference to eq. (1) the following conjecture were formulated [5, p. 248]:

Let \(C_\mu\) be the space of continuous functions \(\nu : [-\tau, 0] \to \mathbb{R}\) with the supremum norm topology. For some positive values of parameters \(\rho, \tau, s\) and \(\alpha\) there exists a continuous measure which is ergodic and invariant with respect to equation (1). Searching of an invariant measures is a very difficult problem, requiring advance mathematical knowledge especially in the measure theory, ergodic theory and the stochastic approach to dynamical systems [8], [17], [20]. One of the methods consist in analysis of the convergence of the Frobenius-Perron operator for given transformation. In the section 2.1 we will briefly describe Frobenius-Perron operator. Other methods have also been reported (see e.g. [9], [18], [2], [1]).

For some parameters numerical simulations of eq. (1) indicates that system exhibits ergodic properties and suggest that continuous invariant measure could exists on some subspaces of the space \(C_\mu\) (see [15]). Here we want to study preliminary evolution of density of distribution of initial states of the system (1). If there exists invariant and ergodic measure numerical simulations should indicate that the initial density of any distribution converges to some invariant limiting density. The goal will be to prepare numerical simulations for eq. (1) in order to be able to set distribution of initial states and later to observe the evolution of initial density.

2. Evolution of Densities

2.1. The Frobenius-Perron operator

Let \((X, \mathcal{A}, \mu)\) be a measure space. Let \(S : X \to X\) be a measurable and nonsingular transformation. The evolution of the density function \(f(x)\) for the initial states under the action of \(S\) is given by the Frobenius-Perron operator \(P\) (see [8], [20]) corresponding to the transformation \(S\) and
defined by the equation
\[ \int_A Pf(x) \mu(dx) = \int_{S^{-1}(A)} f(x) \mu(dx), \quad \text{for } A \in \mathcal{A}. \]  

2.2. Numerical simulations

The space \( C_h \) is an infinite dimensional space, thus we can only investigate the evolution of densities on some subspaces of \( C_h \).

We will analyse the set of trajectories of eq. (1) for constant initial functions, with values distributed with some initial density. We determine the initial density using inverse cumulative distribution functions. Let us take the ensemble \( X_1, X_2, \ldots, X_n \) with density \( f \), that is with the cumulative distribution function \( F \). Now we take \( Y_1, Y_2, \ldots, Y_n \) uniformly distributed on \((0,1)\) and we have

\[ X_i = F^{-1}(Y_i) \]  

(3)

Let us then determine some distributions for the ensemble of values of constant initial functions. For example the normal distribution with mean \( \mu = 0 \) and standard deviation \( \sigma = 2 \) and the exponential distribution with mean \( \mu = 2 \). The cumulative distribution functions of the normal distribution is

\[ F(x) = \frac{1}{2}(1 + \text{erf}(\frac{x - \mu}{\sigma \sqrt{2}})), \]  

(4)

(see Fig. 1 (a)), where erf is the so-called "error function". For the exponential distribution we have

\[ F(x) = \begin{cases} 1 - e^{-\frac{x}{\tau}}, & x \geq 0 \\ 0, & x < 0 \end{cases} \]  

(5)

(see Fig. 1 (b)). Applying formula (3) we can obtain ensemble with given distribution. The result can be observed by displaying the histograms of the locations of the values of constant initial functions (Fig. 2 - normal distribution, of the set of 197 values, \( \mu = 0 \) and \( \sigma = 2 \), Fig. 3 - exponential distribution, of the set of 189 values, \( \mu = 2 \)).

Now we can approximate the density function by dividing the number of values in each bin of the histograms by \( \epsilon \cdot N \), where \( \epsilon \) is the width of the bin and \( N \) is the total number of the values in all bins. Fig. 4 (a) shows approximation of initial density for normal distribution and Fig. 5 (a) for exponential distribution. To study the evolution of this initial densities we calculate the numerical solution of eq. (1) for all initial functions. For obtained ensembles of solutions we construct the histograms of the locations of values for the increasing time of simulation. The histograms are normalized like before to get the approximation of the evolving density functions. The evolution of initial density of normal distribution is presented in Fig. 4 (a)-(g) and the exponential distribution in Fig. 5 (a)-(g). We can see that both initial densities converge to some invariant densities, which additionally seem to be identical. This results indicate possibility of existence of invariant measure, for eq. (1), because the system independently on initial density converge always to the same invariant density. Numerical solutions of eq. (1) were obtained here for \( \alpha = 0.8, \rho = 0.46, \gamma = 1, s = 8 \) and delay \( \tau = 10 \). Calculations were done using MATLAB’s solver dde23.

3. Final Remarks

Discussed measure related with the eq. (1) can be invariant and ergodic (see [15]) what is connected with the chaotic behaviour. In Fig. 6 we have the example of irregular (chaotic) trajectory obtained for eq. (1) with the values

![Figure 1](image1.png)

(a) Cumulative distribution functions of the normal distribution with \( \mu = 0, \sigma = 2 \). (b) Exponential distribution with mean \( \mu = 2 \).

![Figure 2](image2.png)

Figure 2: Histogram for the normal distribution of an ensemble of 197 values of constant initial functions.

![Figure 3](image3.png)

Figure 3: Histogram for the exponential distribution of an ensemble of 189 values of constant initial functions.
Figure 4: Evolution of initial density of normal distribution ($\mu = 0$, $\sigma = 2$). 197 constant initial functions

Figure 5: Evolution of initial density of exponential distribution ($\mu = 2$). 189 constant initial functions
Figure 6: Chaotic trajectory of eq. (1)

of parameters the same as for the simulations of evolution of densities from Fig. 4 and Fig. 5.

Approach to chaotic dynamics, concerning evolution of densities and existence of invariant and ergodic measures brings interesting interpretations of behaviour of biological systems. It shows that chaos in biological systems can be related with their properties and not only the with the difficulties in measuring of very complicated biological parameters [5].

References


Generalized modeling of heterogeneous nonlinear networks

Thilo Gross, Dirk Stiefs, Lars Rudolf, and Martin Zumsande

Max-Planck Institute for the Physics of Complex Systems, Nöthnitzer Straße 38, 01187 Dresden, Germany.

Abstract—Generalized modeling (GM) can be used to explore the dynamics of large, complex networks of nonlinear dynamical elements. Importantly, the interaction terms in the corresponding dynamical systems do not need to be fixed but can be expressed as general (i.e., unspecified) functions, describing a large class of different conventional models. A normalization procedure is then used to derive the Jacobian matrix governing the dynamics close to all steady states in the whole class of models. By GM the Jacobian matrices can be expressed as functions of parameters that have a well-defined interpretation in the context of the respective application. By an ecological example, we show how GM can be used to obtain meaningful information on the dynamics on networks comprising 50 variables and thousands of parameters.

1. Introduction

Many systems in nature, ranging from the biochemical reactions inside cells to societies and ecosystems, can be described as networks of interacting factors [1, 2]. Today the structure of many of these networks is known or purported, offering an opportunity to gain detailed insights in their functioning and failure. However, the functioning of a complex regulatory network is often linked more closely to the network’s dynamics than to the structure [3]. Extracting the dynamics supported by a network with given structure is therefore a central goal for theory [4–9].

Here we focus on the approach of generalized modeling (GM) by which certain features of the dynamics of a system can be extracted efficiently and robustly. GM was first applied in [10] and was subsequently proposed as a general method for the analysis of nonlinear systems in [11]. Subsequently, it was applied to a wide range of different topics including cell signaling [12–14], metabolism [15–18], ecology [11, 12, 19–27], laser physics [11], epidemiology [28] and history [11, 12]. In systems of ordinary differential equations [11] and partial differential equations [25], GM can determine the stability of steady states, detect the local bifurcations in which the stability is lost, identify parameter regions where complex dynamics are likely [21], and also has potential applications in model reduction [27].

2. Modeling approaches: a motivation

Investigation of the dynamics of regulatory networks faces three major obstacles: The networks of interest are typically large and heterogeneous, with network nodes corresponding, for instance, to metabolites with very different chemical properties. Furthermore, the mathematical functions describing processes in the network are often strongly nonlinear and cause dynamics on many different time scales. Finally, there is typically a high degree of uncertainty, which is reflected in most models by a (potentially large) number of unknown parameters.

For the analysis of biological networks, methods from network science and spectral graph theory have recently received much attention. These methods consider the network as an abstract graph and use statistical properties such as the number of connections, the occurrence of certain network motifs, and the length and path of shortest connections between pairs of nodes [1]. Although providing certain general insights into the dynamics of the network, this approach cannot easily make use of specific biological insights, such as the heterogeneity of nodes.

For incorporating specific knowledge and formulating detailed predictions and hypotheses, networks are typically modeled as a dynamical systems, where each network node corresponds to a dynamical variable. The time evolution of these variables is then governed by a system of ordinary differential equations (ODEs). The task of analyzing the dynamics of a complex network is thus mapped to the analysis of a large nonlinear dynamical system.

By far the most common approach to model analysis is simulation. However, for the investigation of the long term behavior this is fundamentally inefficient because detailed information on the transient dynamics, which is later discarded, is obtained at a high computational cost.

Because of the high degree of uncertainty involved in most models, one would ideally wish for an analytical method that can reveal insights without requiring the researcher to fix the parameters to specific values. Indeed, applying the tools of dynamical systems theory [29] can reveal the bifurcation points, i.e., the critical points in parameter space, where qualitative changes in the long-term dynamics occur. This conventional approach to model analysis starts typically with com-
putation the steady states of the model, where all vari-
ables remain constant in time. After a small perturba-
tion from the steady state, a system may either return
asymptotically or depart entirely to approach a differ-
et attractor. In the former (latter) case the steady
state is called asymptotically stable (unstable). The
stability of a given steady state is analyzed by com-
puting the Jacobian matrix, which constitutes a lo-
cal linearization of the dynamical system. The steady
state is asymptotically stable if all eigenvalues of the
corresponding Jacobian have negative real parts. If
changes in parameters cause one or more eigenvalues
to acquire positive real parts the stability of the steady
state is lost in a local bifurcation. Investigating the Ja-
cobian matrices corresponding to the steady states of
the model can therefore analytically reveal the critical
parameter values at which the system departs from
stationary behavior.

Beyond the basic steps described above further tools
dynamical systems theory may reveal boundaries of
more complex dynamics, such as bifurcations of limit
cycles and tori. However, even the basic steps outlined
above can present a considerable challenge. In particu-
lar the first step, the computation of steady states can
be prohibitively difficult if the system contains more
than three or four dynamical equations. Even nume-
crical extensions of the analytical procedure frequently
fail in the analysis of heterogeneous dynamical net-
works with more than, say 20, nodes.

The mathematical difficulties, of the conventional
approach, which mainly arise from the computation
of steady states, are circumvented in random matrix
models [30]. The central idea of this approach is that
the Jacobian of a sufficiently complex dynamical sys-
tem can be modeled as a random matrix. Because
of their simplicity random matrix models can often be
investigated analytically even for large systems. More-
over, a random matrix model does not require the re-
searcher to restrict the underlying processes to a spe-
cific functional forms. On the one hand, formulating a
random matrix model thus requires less assumptions
and can thereby provide more robust insights. On the
other hand, the abstract nature of random matrix
models makes answering specific questions often
very difficult.

3. Generalized Modeling

GM offers an intermediate way between conven-
tional and random matrix models. Specifically, GM
comes close to the generality and efficiency of random
matrix models, while offering interpretability compa-
rable to conventional models. Below we illustrate the
approach of GM by discussing the key ingredients of a
general food web model that has recently been inves-
tigated [11, 24].

As a first step consider just a single biological pop-
ulation \(X\), changing in time due to biological repro-
duction \(S\) and mortality \(M\), leading to the dynamical
system

\[
\frac{d}{dt}X = S(X) - M(X).
\]

Because our focus is on systems where little informa-
tion is available, we avoid restricting \(S\) and \(M\) to spe-
cific functional forms. The aim of our analysis is to
determine the conditions under which a steady state
in the system is stable. Although we cannot compute
steady states at the desired level of generality, we can
formally denote a steady state under consideration as
\(X^*\). We further denote the rates of the two processes
in \(X^*\) as \(M^* = S^*\). We then normalize the system
by introducing \(x = X/X^*\), \(m(x) = M(X)/M^*\), and
\(s(x) = S(X)/S^*\). Using the normalized variables and
functions the model can be written as

\[
\frac{d}{dt}x = \alpha(s(x) - m(x)),
\]

where \(\alpha = S^*/X^*\). In the normalized variables the
steady state under consideration is at \(x^* = 1\), we can
therefore write the corresponding Jacobian as

\[
J = \alpha(s_x - m_x),
\]

where \(s_x = \partial s(x)/\partial x\big|_1\) and \(m_x = \partial m(x)/\partial x\big|_1\) are
coefficients from the linearization.

So far we have succeeded in writing the Jacobian
corresponding to an arbitrary steady state of the sys-
tem as a function of the constants \(s_x\), \(m_x\), and \(\alpha\). The
central insight of GM is that these constants can be
treated as unknown parameters and have in general
a well-defined meaning in the context of the model:
The parameter \(\alpha\) denotes the per-capita growth and
mortality rate in the steady state. It is therefore
simply the inverse of the life expectancy of an in-
dividual in the population. The parameters \(s_x\) and
\(m_x\) are so-called elasticities, logarithmic derivatives
of the original functions in the steady state. For in-
stance \(s_x = \partial \log S/\partial \log X\big|_1\). This implies that if \(S\) is
any power-law \(AX^p\), then the corresponding param-
eter is \(s_x = p\). In contrast to conventional param-
eters, e.g. half-saturation constants, elasticities can
be directly measured in data observed in the steady
state and do not require reference to an artificial state
(e.g. the half-saturation point).

In the one-dimensional example considered here one
can directly read off the single eigenvalue, \(\lambda = \alpha(s_x - m_x)\), of the Jacobian. We can therefore conclude that
in every system of the form of Eq. 1 every given steady
state is stable if the elasticity of the mortality in the
steady state exceeds the elasticity of the reproduction.

Essentially the same procedure applied above can
also be used to study much larger sytems. In the re-
mainder of this paper we discuss three complications
that may arise in such larger systems. The most harm-
less of these is encountered when equations of motion
contain more than two terms. Consider for instance the
example
\[
\frac{d}{dt} X = S(X) - M(X) - F(X, W),
\] (4)
which is analogous to Eq. 1, except that we have in-
cluded an additional loss term describing predation
by a predator \( W \). Typically, \( W \) will follow its own
equation of motion, which we ignore here. Applying
essentially the same procedure as above, we find the
normalized system
\[
\frac{d}{dt} \bar{X} = \alpha(s(x) - \beta m(x) - \bar{f}(x, w))
\] (5)
where \( \alpha = S^*/X^* = (M^* + F^*)/X^* \), \( \beta = M^*/(M^* + F^*) \), and \( \bar{f} = F^*/(M^* + F^*) \). Here, we had to
introduce the new parameter \( \beta \) and its complement
\( \bar{\beta} = 1 - \beta \) weighting the different types of losses. In
words, \( \beta \) is the probability that an individual will
eventually be eaten and \( \bar{\beta} \) the probability that it will live
until its natural death.

In general, the normalization of a dynamical equa-
tion containing \( N \) terms will require introducing \( N - 1 \)
parameters. It is generally advantageous to introduce
one parameter (\( \alpha \)) describing the per-capita turnover
rate and \( N - 2 \) parameters (\( \beta \)) weighting the contribu-
tions to the total gain and loss rates, respectively.

A second more subtle complication is for instance
encountered in a model where \( X \) is a predator feeding
on two prey populations \( Y \) and \( Z \) such that
\[
\frac{d}{dt} X = S(X, Y, Z) - M(X)
\] (6)
When we carry out the normalization procedure, we
end up with a Jacobian containing the elasticities
\( s_y = \partial s/\partial y \) and \( s_z = \partial s/\partial z \) describing the sensitivity
of the predation rate to the size of the prey populations.
However, ecological knowledge may tell us that the two
elasticities are not unrelated. For instance if one of the
two species were very abundant then the sensitivity of
the predation to the size of the rarer prey population
is greatly reduced because the predator is likely to be
saturated from a encounters with the abundant one.

Insights as the one described above can be inte-
grated into a previously derived GM by an iterative
refinement procedure. Suppose for instance that be-
yond what is stated in Eq. (6) we know predation to
depend only on the sum of the sizes of the two
prey populations, i.e. \( S(X, Y, Z) = S(X, T) \), where
\( T = Y + Z \). The algebraic equation for the total
amount of prey, \( T \), can be normalized like the dif-
ferential equation, yielding \( t = \gamma y + \bar{\gamma} z \), where the
parameters \( \gamma = Y^*/(Y^* + Z^*) \), \( \bar{\gamma} = 1 - \gamma \) measure the
relative contributions of the two prey species. Using
this new relation we can now write
\[
g_y = \frac{\partial s}{\partial y} - \frac{\partial s}{\partial y} \frac{\partial s}{\partial t} = \gamma g_t
\] (7)
and analogously \( g_z = \bar{\gamma} g_t \), where \( g_t \) is now the elastic-
ity of the predation rate with respect to \( T \). By sub-
stituting the equations for \( g_y \) and \( g_z \) we can rewrite a
previously derived Jacobian in terms of the new pa-
rameters incorporating the additional insight on the
dependence of \( g_y \) and \( g_z \). Although this refinement
does not generally reduce the number of parameters
in a GM it often facilitates the interpretation of re-
sults.

The final and perhaps most obvious complication
encountered in larger models is having more dynamical
variables and hence more equations of motion and
larger Jacobians. In this case the normalization pro-
sEDURE is applied to all of the equations of motion.
For large systems containing tens or even hundreds of
equations, the manual work can be reduced by using
the matrix formalism proposed in [17]. When deal-
ning with GMs of large systems the main challenge is
therefore to extract information from large Jacobians.
Using the method described in [10] analytical compu-
tation of the local bifurcations is feasible for systems
of up to 10 dynamical variables. Furthermore, a nu-
merical procedure that is applicable to larger system
is described in [12].

In large systems identifying the decisive parameters
having a strong impact on stability can be a challeng-
ing task. In GMs this is often accomplished by a Monte
Carlo sampling of the parameter space: We create a
large ensemble of parameter sets, where each param-
eter in each set is randomly chosen. In the second step
we compute the stability of the steady states corre-
sponding to the random parameter sets by substitut-
ing one set at a time into the Jacobian and numerically
computing the leading eigenvalue of the matrix that is
obtained. To each parameter set we assign a stability
value \( q_i \), which is 1=stable if the leading eigenvalue is
negative and 0=unstable otherwise. We can then esti-
mate the impact of a given parameter, say \( m_x \), on sta-
bility by computing the correlation between the values
of \( m_x \) and \( q_i \) in the ensemble. More detailed insights
can be gained for instance by plotting a histogram of
the fraction of stable states found, over one of the pa-
rameters [17, 24].

The analysis described above profits greatly from
the efficiency of GMs. Because the computation of
the leading eigenvalue of a matrix is much faster than,
for instance, the simulation of the corresponding dy-
namical system very large ensembles can be studied.
For instance in [24] approx. \( 10^{11} \) sample parameter
sets were used for analyzing a 50-dimensional systems
containing several thousand parameters.
References


Hierarchical and Modular organization of Corticocortical Networks supports Functional Integration and Segregation in the Mammalian Brain

Gorka Zamora-López1,2,6,*, Changsong Zhou3,4 and Jürgen Kurths5,6

1 Interdisciplinary Center for Dynamics of Complex Dynamical Systems, University of Potsdam, Germany
2 Bernstein Center for Computational Neuroscience, Berlin, Germany
3 Department of Physics, Center for Nonlinear Studies, Hong Kong Baptist University, Hong Kong, China
4 The Beijing-Hong Kong-Singapore Joint Centre for Nonlinear and Complex Systems, Hong Kong Baptist University, Hong Kong, China
5 Potsdam Institute for Climate Impact Research, Potsdam, Germany
6 Department of Physics, Humboldt University, Berlin, Germany
*Email: gorka@agnld.uni-potsdam.de

Abstract—There is increasing evidence that the architecture of corticocortical networks support the capacity of the brain to simultaneously segregate and integrate information. Here, we confront the concept of integration, which is rarely addressed in the literature of information theory, to cross-validate the functional implications assigned to such networks. We propose a novel framework to quantify the segregative and integrative properties of cortical networks by defining a minimal set of conditions their nodes need to obey from a functional perspective. Application of these conditions shows that a particular set of cortical areas are highly responsible for the integration of multisensory information. This set coincides with the areas predicted from the purely topological analysis to perform such a function. Our findings are in agreement with modern models which propose that high-level brain functions emerge from interactive and overlapping networks of neurones which trascend any of the traditional subdivisions of the cortex by structural (cytoarchitecture) and functional criteria.

1. Introduction

Sensory neurones encode environmental information into electrical signals which propagate in a “bottom-up” manner through different processing stages of the nervous system [1, 6]. Information of the same modality (e.g. visual, auditory, somatosensory, etc.) traverses the body together, typically separated from the processing paths of other modalities. This permits that particular regions of the cortex specialise in detecting features of the sensory stimuli, e.g. orientation, velocity and colour of the visual input; or frequency and pitch of the auditory stimuli. However, in order to generate a coherent perception of the reality, the brain needs to combine (integrate) this multisensory information at some place [8] and during some time [2, 3, 11]. For that, the paths of information need to converge.

There is increasing evidence that the functional capacity of the nervous system to balance between segregation (specialisation) and integration might be facilitated by its structural organisation. Analysis of the connectivity between regions of the cerebral cortex in macaque monkeys and cats has revealed their modular organisation [10, 9, 4, 5]. Two areas are more likely connected if both are specialised in the processing of the same modal information (e.g. visual or auditory information). Additionally, the network contains several interconnected hubs, conforming a modular organisation with centralised hierarchy [13] that might be an optimal natural solution to keep different information separated, but permitting at the same time a controled integration of all the information.

In this paper we challenge the functional properties of this modular and hierarchical structure by means of dynamical and information theoretical measures. We propose minimal conditions that lead to integration of multisensory information and we test them in the corticocortical network of the cat. In order to objectively detect the set of hubs that optimise integration, we perform a statistical analysis of these properties in a wider ensemble of possible hub combinations. We find that only simultaneous lesion of particular hubs leads to a dynamical segregation of the sensory modules (visual, auditory, somatosensory-motor and frontolimbic), and only the same hubs form a dynamical cluster after simultaneous excitation of primary sensory areas, a clear sign of their integrative capacities.

2. Data

After an extensive collation of literature reporting anatomical tract-tracing experiments, Scannell and Young [10, 9] published a dataset containing the corticocortical and cortico-thalamic projections between regions of one brain hemisphere in cats. The connections were weighted according to the axonal density of the projections. Connections originally reported as weak or sparse were classified with 1 and, the connections originally reported as strong or dense with 3. The connections reported as intermediate strength, as well as those connections for
which no strength information was available, were classified with 2. Here we make use of a version of the network consisting of $N = 53$ cortical areas interconnected by $L = 826$ directed corticocortical projections.

3. Functional Capacity of Integration

While information theory has largely dealt with describing and quantifying channel capacity, coding and decoding of signals, etc., it has not faced the problem of integration. Many natural and artificial systems, such as the nervous system, need to deal with information arising from different sources. In this sense, we aim for a definition of integration which characterises the capacity of a system to receive and process information of different character and to combine it generating new useful information. Certainly, this definition involves crucial theoretical problems, e.g. what the character of information is, or what are the rules under which information is combined. Nevertheless, within a networked system, the nodes with a capacity to integrate information should obey certain measurable conditions. We propose the following:

1) Accessibility to information: A node can perform an integrative function only if it has general access to the information contained within the system.

2) Sharing of information: Two or more nodes can perform integrative function in a collaborative manner only if they are sufficiently connected with each other.

As a corollary of these two conditions, we should include the following, third condition:

3) Segregation after selective damage: If a node has an integrative function, its removal should lead to a decrease of the integrative capabilities of the whole system.

In order to test these conditions on the central hubs of the corticocortical network of the cat, we perform two numerical experiments: (i) after excitation of primary sensory areas, we quantify the dynamical interdependence between the hubs, see Figure 1a, and (ii) we measure the decrease in integration capacity of the network after the hubs have been removed, Figure 1b. Aiming for a statistical description that objectively highlights the more relevant hubs for integration, we relax the definition of hub and perform both experiments for each of the 524,097 possible sets of cortical hubs, of sizes $N_S = 1$ to $N_S = 19$, formed by the following set of areas:

$S_{hubs} = \{20a, 7, AES, EPP, 6l, 6m, 5Am, 5Al, 5Bm, 5Bl, SSSA, SSAo, PFL, LA, Ig, CGA, CGP, 35, 36\}$.

The steady-state of a linear system whose $N$ subsystems $x = (x_1, x_2, \ldots, x_N)$ are driven by a Gaussian noise $\xi = (\xi_1, \xi_2, \ldots, \xi_N)$, is described by $x_t = g \sum_j A^j_t x_j + \xi_t$, where $g$ is the coupling strength and $\hat{A}^j_t$ is the normalised transpose of the adjacency matrix. Otherwise the dynamics of $x_i$ would be characterised by its own outputs, not by the inputs it receives. The adjacency matrix is normalised by its largest eigenvalue such that the coupling strength $g$

![Figure 1: Schematic representation of the numerical experiments. (A) The network is stimulated by increasing the noise level of the primary sensory areas. The consequent integration $I(S)$ between a group of hubs is measured. (B) After simultaneous removal of a set of hubs, the integration capacity $I(X - S)$ of the remaining network is measured.](image)

is equivalent for networks of different size and topology. Written in matrix form:

$$x = g\hat{A}^j x + \xi.$$

The entropy of such a multivariate Gaussian system can be analytically calculated as $H(X) = \frac{1}{2} \log |\text{COV}(X)|$, where $| \cdot |$ stands for the determinant $[7, 12]$. The entropy of an individual Gaussian process is $H(x_i) = \frac{1}{2} \log(2\pi e v_i)$, where $v_i$ is the variance of $x_i$, say, the $i^{th}$ diagonal element of the $\text{COV}(X)$ matrix. The covariance matrix can be analytically computed by solving the system such that $x = \frac{1}{1-\alpha} \diamond \xi$, and averaging over the states produced by successive values of $\xi$ one finds: $\text{COV}(X) = \langle x \cdot x^\top \rangle = \langle (Q \cdot \xi) \cdot (Q^\top \cdot \xi^\top) \rangle = Q \cdot Q^\top$.

Following Tononi and Sporns [12] we define the integration of the system $X$ as the extension of the mutual information for $X$ composed of more than two subsystems:

$$I(X) = \sum_{i=1}^{N} H(x_i) - H(X)$$

where $H(x_i)$ is the entropy of one subsystem and $H(X) = H(x_1, x_2, \ldots, x_N)$ is the joint entropy of the system considered as a whole. $I(X) = 0$ only if all $x_i \in X$ are statistically independent of each other, and positive otherwise. $I(X)$ measures the internal level of statistical dependence among all the subsystems $x_i \in X$. Replacing $H(X)$ and $H(x_i)$ of the linear system into Equation (2) and applying basic algebra,
we reduce the integration of such a multivariate Gaussian system as:

\[ I(X) = \frac{1}{2} \log \left| \prod_{i=1}^{N} v_i \right| \text{COV}(X) \].

(3)

4. Results

In the following, we perform the two numerical experiments considering, unless otherwise stated, coupling strength \( g = 0.5 \) for all the links and a noise level of \( \xi_j = 1.0 \) added to all the areas. These parameters lead to similar covariance matrices as those obtained from more realistic models [14, 15, 16].

4.1. Dynamical integration after sensory stimulation

We simulate external stimulation by an increase in the noise level of primary sensory areas: primary visual cortex (area 17), primary auditory cortex (area AI) and primary somatosensory cortex (areas 1, 2 and 3b). According to [9] the cortical areas 1, 2 and 3b are subregions of the primary somatosensory area, named by some authors as SI. We simultaneously excite all the primary sensory areas \{17, AI, 1, 2 and 3b\} by assigning them a larger noise level \( \xi_j = 10.0 \) and we measure the local integration among the areas in a subsets \( S \) of hubs out of \( S_{\text{hubs}} \). Because of the excited condition, the integration of the subsets is denoted as \( I'(S) \). The integration \( I(S) \) of a subset of nodes \( S \) is computed as in Equation 3 by extracting \( \text{COV}(S) \) as a sub-matrix of \( \text{COV}(X) \) formed by the nodes in the set \( S \), and by considering the variance \( v_i \) of the nodes in the set.

The results depicted in Figure 2A show that \( I'(S) \) can largely differ. For example, among all the subsets of size \( N_x = 10 \), the integration of some of them is very small, \( I'(S) \sim 0.1 \), while the integration of others becomes much larger, \( I'(S) \sim 0.5 \). These differences permit us to identify those cortical hubs which, grouped together, become more dynamically dependent among them as a consequence of the multisensory stimulation. Considering only those subsets whose \( I'(S) \) lies within the largest 10% (red crosses in Figure 2A) a co-participation matrix \( C \) is constructed such that \( C_{ij} \) is the number of times (given in frequency) that two cortical hubs participate together in one of the maximal sets, Figure 2B. It is observed that areas \{7, AES; EPP; 6m; 1a, 1g, CGp, 35, 36\} participate together in over 50% of the occasions with those areas in the core. The remaining areas, \{3Am, 5AI, 5Bm, 5BI, 5SSAI, 5SSAo and PFCL\}, can be discarded as members of the dynamical core.

4.2. Dynamical segregation after multiple lesions

For all the possible subsets \( S \) composed of hubs in \( S_{\text{hubs}} \), we perform a lesion to the network by simultaneously removing the nodes \( x_i \in S \) and characterise the consequent functional segregation of the network as the change in statistical dependence between the four modules (V, A, SM and FL). Lesion of areas critical for the integration capacities of the system should lead to a dynamical segregation of the modules, i.e. a decrease in their statistical dependence.

As defined in Equation (3), \( I(X) \) represents the limit case in which the statistical dependence among all the elements \( x_i \) in the system \( X \) is quantified. To cover different scales of organisation we propose to characterise the statistical dependence between groups of elements. Imagine a partition \( \mathcal{P} = \{S_1, S_2, ..., S_n\} \) into \( n \) groups (modules) of the elements \( x_i \) such that \( X = S_1 \cup S_2 \cup \ldots \cup S_n \). Then, we define the modular integration of the partition \( \mathcal{P} \) as:

\[ I_{\mathcal{P}}(X) = \sum_{j=1}^{n} H(S_j) - H(X). \]

(4)

Note that when \( n = N \), then \( I_{\mathcal{P}}(X) = I(X) \).

Considering the partition \( \mathcal{P}_4 = \{V, A, SM, FL\} \) and the corticocortical network of the cat, then \( I_{\mathcal{P}_4}(G_{\text{cov}}) = 0.292 \). The modular integration of each lesioned network \( G_{\mathcal{P}_4} \) is computed for the partition \( \mathcal{P}_4 \). Notice that a) the nodes are also removed from the partition and b) every \( G_{\mathcal{P}_4} \) is adequately normalised by its largest eigenvalue such that the measured observables are comparable across realisations. The results in Figure 2C permit us again to discriminate between subsets of hubs whose simultaneous removal lead to a large segregation of the network, while removal of other subsets has barely no effect.
Selecting only those subsets whose lesion leads to a larger segregation of the modules, i.e. \( I_p(G_2) \) lies among 10% of the minimal modular integration for each size \( N \) (red dots in Figure 2A), a co-participation matrix \( C \) is constructed, Figure 2D. The entries \( C_{ij} \) are the number of times (given in frequency) that two areas participate together in one of the minimal subsets. A core of cortical areas is found which participate together in over 70% of these cases: \{ AES; EPp; Ia, Ig, CGp, 35, 36 \}. Somatosensory-motor areas \( 6m, 5Al \) and \( 5Bl \) join them in over 50% of the cases.

5. Conclusions

In summary, we have analysed the modular and hierarchical organisation of the corticocortical network of the cat and its relationship to the intrinsic necessities of the brain to simultaneously segregate and integrate multisensory information. By means of dynamical and information theoretical measures, we have corroborated its capacity to integrate multisensory information, i.e. after simultaneous excitation of visual, auditory and somatosensory primary areas, a particular set of hubs becomes dynamically dependent forming a cluster. Additionally, the simultaneous lesion of these hubs leads to a largest decrease in the integrative capacities of the network. Both numerical experiments indicate that visual areas 7 and AES, auditory area EPp and frontolimbic areas Ia, Ig, CGp, 35 and 36 are the most likely candidates to form the top hierarchical module. The participation of somatosensory-motor areas is less clear, although area \( 6m \) is the strongest candidate of them. Visual area \( 20a \) and somatosensory-motor areas \( 5Al \) and \( 5Bl \) are also potential candidates. This set largely coincides with the top hierarchical level found by the graph analysis [13], corroborating the integrative function assigned to the hubs by intuitive interpretation of their topological characteristics.

6. Acknowledgments

We thank the organisers for inviting us to the conference. G. Z.-L. and J. K. are supported by the Deutsche Forschungsgemeinschaft (grants EN471/2-1, KL955/6-1, and KL955/14-1) and by the BioSim network of excellence contract No. LSHB-CT-2004-005137 & No. 65533., C.S. Z. is supported by the Hong Kong Baptist University.

References


Recurrence network approach to a phase space of a time-delay system

D. V. Senthilkumar†‡, N. Marwan‡ and J. Kurths‡,*

†Centre for Dynamics of Complex Systems, University of Potsdam, 14469 Potsdam, Germany
‡Potsdam Institute for Climate Impact Research, 14473 Potsdam, Germany
* Institute for Physics, Humboldt University, 12489 Berlin, Germany
Email: senthil@pik-potsdam.de, marwan@pik-potsdam.de, kurths@pik-potsdam.de

Abstract—An interesting potential approach for non-linear time series analysis by exploiting the analogy between the recurrence matrix, representing the recurrences in phase space, and the adjacency matrix of a complex network to characterize and analyze the dynamical transitions in the phase space of complex systems is being emerging. In this work, we present our preliminary results by applying this method to a high dimensional phase space of a time-delay system.

1. Introduction

Among modern data analysis techniques recurrence analysis has its unique advantages and is being used as a potential tool for time series analysis in almost all branches of science and technology [1]. The analogy between the recurrence matrix and the adjacency matrix has recently provoked a flurry of investigations in employing complex network measures to recurrences in phase space to analyse and characterize dynamical transitions in phase space in terms of network topology (cf.[2, 3, 4, 5, 6, 7]).

Recently, the network measures, namely link density (ρ), average path length (L) and clustering coefficient (C), along with the recurrence quantification analysis (RQA), namely maximal diagonal line length (Lmax) and laminarity (LAM) are estimated to capture the dynamical transitions in the well-known logistic map [2], as shown in Fig. 1, in terms of these measures. The dotted lines in these figures correspond to the four different dynamical regimes (i) period-3 window at a = 3.830, (ii) band merging at a = 3.679, (iii) cross points of supertrack functions at a = 3.791 and (iv) outer crisis at a = 4, which has been analysed in detail [2].

As a natural extension of these studies to high dimensional phase space, we have applied the above measures to a time-delay system, essentially an infinite-dimensional system, and discuss our preliminary results and the difficulties involved in it. In particular we consider a piecewise linear time-delay system and apply these measures to analyse the dynamical transitions in such a high-dimensional phase space.

The plan of the paper is as follows. In Sec. 2, we introduce the model system and discuss briefly about its dynamical properties. We will point out the measures that we have employed in Sec. 3. We will present our preliminary results and discuss the practical difficulty involved in analyzing such a high dimensional phase space in Sec. 4. Finally, in Sec. 5, we present our summary and conclusion.

2. Scalar piecewise linear time-delay system

We consider the following scalar first order delay differential equation represented as

\[ \dot{x}(t) = -ax(t) + bf(x(t-\tau)) + c, \quad (1) \]

where a, b and c are parameters, \( \tau \) is the time-delay and \( f \) is an odd piecewise linear function defined as

\[ f(x) = \begin{cases} 
0, & x \leq -4/3 \\
-1.5x - 2, & -4/3 < x \leq -0.8 \\
x, & -0.8 < x \leq 0.8 \\
-1.5x + 2, & 0.8 < x \leq 4/3 \\
0, & x > 4/3.
\end{cases} \quad (2) \]
We have investigated the above system in detail including
linear stability analysis, bifurcation analysis and transient effects [8]. For the choice of the parameters $a = 1.0, b = 1.2, c = 0.001$ and $\tau = 25.0$

linear of a system in phase space is given, with $X_t \in \mathbb{R}^3$. The RP efficiently visualises recurrences and can be formally expressed by the matrix

$$R_{i,j} = \Theta(\epsilon - ||X_i - X_j||), \quad i, j = 1, \ldots, N,$$

where $N$ is the number of measured points $X_t$, $\epsilon$ is a predefined threshold, $\Theta$ is the Heaviside function and $||.,||$ is the Euclidean norm. For $\epsilon$-recurrent states, that is for states which are in an $\epsilon$-neighbourhood, we have the following notion:

$$X_i \approx X_j \iff R_{i,j} \equiv 1.$$  

The graphical representation of the matrix $R_{i,j}$ is called recurrence plot (RP). The RP is obtained by plotting the recurrence matrix, Eq. (3), using different colors for its binary entries, for example by marking a black dot at the coordinates $(i,j)$, if $R_{i,j} \equiv 1$, and a white dot, if $R_{i,j} \equiv 0$. More details about the RPs and RQA along with their applications can be found in [1]. Now, we will directly introduce the measures which we have used for our analysis:

1. Maximal diagonal length ($L_{\text{max}}$) defined as

$$L_{\text{max}} = \max_x (|R_{i,j}|),$$  

where, $l$ is length of diagonal lines and $N_l$ is their total number.

2. Laminarity defined as

$$LAM = \frac{\sum_{v=1}^{N_l} v p(v)}{\sum_{v=1}^{N_l} p(v)},$$  

where $p(v)$ is the distribution of the vertical lines of at least length $v$.

3. Link density given as

$$\rho = \frac{1}{N(N-1)} \sum_{i,j=1}^{N} A_{i,j},$$  

corresponding to the global recurrence rate and $A_{i,j} = R_{i,j} - \delta_{i,j}$, where $\delta_{i,j}$ is the Kronecker delta.

4. Clustering coefficient, $C = \sum_i C_i/N$, where the local clustering coefficient $C_i$ is defined as

$$C_i = \frac{\sum_{v=1}^{N} A_{i,v}A_{i,v}/k_v}{k_v(k_v-1)},$$  

where $k_v = \sum_{i=1}^{N} A_{i,v}$ is the degree centrality giving the number of neighbors of node $v$.

5. The average length of shortest path between all pairs of nodes is given by the average path length

$$L = \frac{1}{N(N-1)} \sum_{i,j=1}^{N} d_{i,j},$$  

where $d_{i,j}$ is the shortest path connecting the nodes $i$ and $j$. 

---

Figure 2: (a)The hyperchaotic attractor of the piecewise linear time-delay system, (1), for the choice of the parameters $a = 1.0, b = 1.2, c = 0.001$ and $\tau = 25.0$

Figure 3: The first ten maximal Lyapunov exponents $\lambda_{\text{max}}$ of the scalar time-delay system, (1), for the parameter values $a = 1.0, b_1 = 1.2c = 0.001$, $\tau \in (2, 29)$
More details and discussions on these measures and their application to understand the dynamical transitions in the phase space of complex systems can be found in Refs. [2, 3, 4].

Figure 4: (a-c) Periodic attractor, its time series and its corresponding RP, respectively, for the value of \( c = -0.1 \) and (d-f) hyperchaotic attractor, its time series and its corresponding RP, respectively, for the value of \( c = -0.06 \).

4. Application to the time-delay system

To investigate the structural changes in the above mentioned measures corresponding to the dynamical transitions in the phase space of the piecewise linear time-delay system, we consider the same parameter values \((a = 0.16, b = 0.2, \tau = 25.0)\) and the bifurcation diagram as in Fig 4 of Ref. [8]. In our simulations, we have left sufficiently large transients and analysed time series of length \( N = 100,000 \). We have fixed the integration time step as \( \Delta t = 0.01 \), sampling interval as \( \Delta t_s = 100 \) and the threshold value for \( \epsilon = 0.13\sigma \), where \( \sigma \) is the standard deviation. The periodic attractor projected in the phase space \((x(t), x(t + \tau))\) with \( \hat{\tau} = 10 \) and its corresponding time series for the value of \( c = -0.1 \) are shown in Figs. 4a and 4b, respectively. Similarly, the chaotic attractor and its corresponding time series for \( c = -0.06 \) are shown in Figs. 4d and 4e, respectively. The RPs of the periodic (Fig. 4a) and the chaotic (Fig. 4d) attractors are shown in Figs. 4e and 4f, respectively.

The bifurcation diagram in the range of the control parameter \( c \in (-0.1, -0.05) \) is shown in Fig. 5a. We have calculated the values of the measures, mentioned in Sec. 3, corresponding to the attractors in the phase space \((x(t), x(t + \hat{\tau}))\) with \( \hat{\tau} = 10 \). Equivalently one may also consider other phase variables \( x(t + M\Delta t) \), where \( M = t / \Delta t = \frac{2500}{0.01} = 2500 \) for the chosen values of the delay time \( \tau \) and the integration time step \( \Delta t \). However, more attention have to be paid for choosing the threshold value of \( \epsilon \) to avoid other recurrences within the \( \epsilon \) neighbourhood due to the tangential motion, namely, the sojourn points. This plays a vital role in determining the resulting structures in the above measures.

The maximal diagonal line length \( (L_{max}) \) is depicted in Fig. 5b as a function of \( c \) corresponding to the bifurcation diagram (Fig. 5a). As expected \( L_{max} \) is very large (nearly equal to the length of the considered time series after the sampling) in the periodic regimes. However, \( L_{max} \) does not acquire low values in the chaotic regime, when compared to the logistic map (Fig. 1b) as expected due to the highly disconnected diagonal lines in the RP for a chaotic attractor in general. Nevertheless, the scenario is different in the case of chaotic/hyperchaotic attractors of time-delay systems as such systems will have trajectories with large periods as confirmed by the long diagonal lines in the RP of the chaotic attractor (Fig. 4f). It is also to be accounted that small amplitude to the large value of \( L_{max} \) is also contributed by the sojourn points.

The laminarity \((LAM)\), illustrated in Fig. 5c, in the periodic regime should be almost around zero (as in Fig. 1c of Logistic map) as there should not be any clusters of recurrence points in the periodic regime. However the large values of \( LAM \), throughout the range of \( c \) and in the periodic regime in particular, is due to the recurrence of a large number of tangential motions within the \( \epsilon \) neighbourhood. Due

Figure 5: (a) Bifurcation diagram of the scalar piecewise linear time-delay system for the parameter values \( a = 0.16, b_1 = 0.2, \tau = 25 \) and \( c \in (-0.1, -0.05) \). Selected RQA measures: (b) maximal diagonal line length \( L_{max} \) and (c) laminarity \( LAM \), as well as complex network measures: (d) link density \( \rho \), (e) average path length \( L \) and (f) clustering coefficient \( C \).
to high dimensional phase space of the time-delay systems, 
the projection of the trajectory in certain phase space may 
remain static for long time, while it is evolving in some 
other phase space. This may be avoided by considering 
more number of phase space/variables in embedding, while 
constructing the recurrence matrix and choosing appropriate 
phase space.

The link density ($\rho$), shown in Fig. 5d, quantifies the av-
erage phase space density. It attained large values in the 
periodic regime and low values in the chaotic regime as ex-
pected corresponding to large recurrences in the periodic 
regime (Fig. 4a) and comparatively low recurrences in the 
chaotic regime (Fig. 4d). However, as mentioned earlier 
the sojourn points will have small contribution to the am-
plitude of $\rho$. The average shortest path length ($L$) and the 
clustering coefficient ($C$) are plotted in Figs. 5e and 5f, 
respectively. In the periodic regimes, the different periods 
correspond to different disconnected components, as they 
never occur at the same point in phase space, with each 
components being a fully connected network as each pe-
riodic trajectory/behavior represents the same state in the 
phase space. Hence $L$ should acquire the value unity and $C$ 
take the largest possible value ($C = 1$) in the peri-
odic regime as in Figs. 1e and 1f, respectively, for the case 
of the Logistic map. The average shortest path length takes 
the small value and the clustering coefficient takes largest 
value in the periodic regime as can be seen in Figs. 5e and 5f, respectively. However, $L$ is slightly above and $C$ is less 
than unity in the periodic regime because of the presence 
of sojourn points. In chaotic regime, $L$ acquires larger values 
and $C$ takes lower values than in the periodic regime as 
expected.

5. Summary and Conclusion

We have presented our preliminary results by extending 
the concept of recurrence network to a high dimensional 
system, namely a scalar piecewise linear time-delay sys-

tem. In particular, we have estimated some of the recur-
rence quantification and network measures based on the 
recurrences in the phase space of the time-delay system. 
We have found that these measures characterize and quan-
tify dynamical transitions in high-dimensional space state 
similar to that observed in the well-known logistic map. 
However, applying the recurrence network concept to re-
trieve exactly the dynamical and the statistical properties 
involved in the high-dimensional space state is a difficult 
task. One has to carefully analyse the phase spaces of 
infinite-dimensional systems such as time-delay systems 
to choose more appropriate phase spaces, in which most 
of the dynamical transitions occur, to be considered in the 
embeddings to construct the recurrence matrix to avoid any 
artifact.

Acknowledgments

DVS has been supported by Alexander von Humboldt 
Foundation. JK has been supported by his Humboldt-CSIR 
research award and EU project No. 240763 PHOCUS(FP7-ICT-2009-C).

References

[1] N. Marwan, M. C. Romano, M. Thiel and J. Kurths, 
“Recurrence plots for the analysis of complex sys-

[2] N. Marwan, J. F. Donges, Y. Zou, R. V. Donner and 
J. Kurths, “Complex network approach for recur-
rence analysis of time series,” Phys. Lett. A, vol.373, 

J. Kurths, “Ambiguities in recurrence-based complex 
network representations of time series,” Phys. Rev. E, 

J. Kurths, “Recurrence networks-a novel paradigm for 
nonlinear time series analysis,” New J. Phys., vol.12, 

ena and motifs of networks induced from time series,” 
2008.

14), 2009.

based on phase space reconstruction,” Chaos, vol.19, 
pp.033137(1-12), 2009.

and chaos in time delayed piecewise linear dynamical 
systems,” Int. J. Bifurcation and Chaos, vol.15, 

[9] J. D. Farmer, “Chaotic attractors of an infinite-
dimensional dynamical system,” Physica D, vol.4, 
Recurrence-based evolving networks for time series analysis of complex systems

Reik V. Donner1,2, Jonathan F. Donges1,3, Yong Zou1, Norbert Marwan1, and Jürgen Kurths1,3

1Research Domain IV Transdisciplinary Concepts and Methods, Potsdam Institute of Climate Impact Research
P.O. Box 60 12 03, 14412 Potsdam, Germany
2Institute for Transport and Economics, Dresden University of Technology
Würzburger Str 35, 01187 Dresden, Germany
3Department of Physics, Humboldt University of Berlin
Newtonstrae 15, 12489 Berlin, Germany
Email: reik.donner,donges,yong.zou,marwan,juergen.kurths@pik-potsdam.de

Abstract—This paper presents a novel approach for analyzing the structural properties of time series from real-world complex systems by means of evolving complex networks. Starting from the concept of recurrences in phase space, the recurrence matrices corresponding to different parts of a time series are re-interpreted as the adjacency matrices of complex networks, which link different observations if the associated temporal evolution is sufficiently similar. We provide some illustrative examples demonstrating that the local properties of the resulting recurrence networks allow identifying dynamically invariant objects in the phase space of complex systems. Moreover, changes in the global network properties of evolving recurrence networks allow identifying time intervals containing hidden dynamical transitions, which is exemplified for some financial time series.

1. Introduction

During the last years, classical graph theory has been systematically extended and applied for studying real-world networks in various scientific disciplines. In particular, the corresponding results have triggered substantial progress in our understanding of the interplay between structure and dynamics of complex networks, i.e., systems that are composed of a number of mutually interacting units [1, 2, 3, 4].

In 2006, Zhang and Small [5, 6, 7] suggested studying the topological features of pseudo-periodic time series in terms of complex networks. For this purpose, individual cycles (defined by distinct minima or maxima of the studied time series) have been considered as vertices of a network, the connectivity of which has been established by different proximity measures. A similar approach applicable also to time series without obvious oscillatory components has been suggested by Yang and Yang [8], considering embedded time series and the resulting phase space vectors as vertices, whose mutual (Pearson) correlation determines the network connectivity. Both approaches are based on the mutual proximity of different parts of a time series in a certain abstract space and utilize thresholds to this proximity for determining a network pattern. A general alternative to such proximity networks has been recently suggested by Lacasa et al. [9] in terms of so-called visibility graphs.

In this work, we apply an alternative threshold-based concept, which exploits recurrences in phase space. In this formalism, a state (phase space vector) \( X(t) \) is said to be recurrent if there is \( t_j \neq t_i \) such that \( d(X(t_i) - X(t_j)) < \epsilon \) for some distance measure \( d(\cdot, \cdot) \) in phase space. Under general conditions, the structure of recurrences in phase space can be simply encoded in terms of the recurrence matrix [10, 11]

\[
R_{i,j} = \Theta(\epsilon - d(X(t_i) - X(t_j)))
\]  

(1)

where \( \Theta(\cdot) \) is the Heaviside function. Following the above considerations, this matrix can be re-interpreted as the adjacency matrix of an unweighted complex network associated with the given time series (more specifically, the adjacency matrix is given by \( A_{i,j} = R_{i,j} - \delta_{i,j} \)), which is called (\( \sim \)) recurrence network [12, 13, 14, 15]. Note that such recurrence networks and closely related concepts have been independently suggested by a variety of authors (see [14, 15] for details).

The advantage of considering the concept of recurrences instead of defining distances in terms of correlations [8] is that it allows generating networks based on individual observations without any embedding or consideration of groups of observations. Recent results have revealed that some of the fundamental dynamical invariants of complex systems are conserved in the recurrence matrices obtained without embedding [16] (even more, embedding is found to sometimes induce spurious correlations [17]). This preservation property allows the full reconstruction of a time series from its recurrence matrix (modulo some rescaling of its probability distribution function) [18, 19].

We have to underline that the properties of a recurrence network do not depend on the temporal order of vertices, i.e., the corresponding complex network measures are invariant under permutations of the observations in the underlying time series. In this sense, recurrence networks en-
code purely geometric information on a time series (similar to fractal dimensions and related concepts), which is distinctively different from most established methods of time series analysis that rely on temporal correlations between observations (including such that are commonly used for the quantitative analysis of recurrence matrices [11]). Consequently, recurrence networks capture different properties of attractors or, more generally, time series than other methods of time series analysis. Recently, the detection of dynamical transitions in time series [12, 15] and invariant objects in phase space [13, 14] have turned out to be two very promising fields of application of these networks. In this work, we review the basic corresponding results and provide additional examples for both types of application.

2. Detecting invariant objects in phase space

Dynamically invariant objects in the phase space of complex systems, such as invariant manifolds or unstable periodic orbits, can be detected by considering the local vertex properties of recurrence networks. Basic examples for such measures are degree, closeness, and betweenness centrality. Specifically, the degree centrality $k_v$ measures the number of direct neighbors of a vertex $v$ with respect to a given spatial threshold distance $\varepsilon$, i.e., it is proportional to the local phase space density. Closeness centrality $c_v$ is related to the inverse mean network distance of a vertex with respect to all other vertices, implying that high values of closeness appear in the central parts of the attractor, whereas the outer parts are characterized by small values. Betweenness centrality $b_v$ particularly highlights phase space regions with a low state density, which separate regions of higher density. Hence, it characterizes the local attractor fragmentation. Note that although betweenness and degree are not fully independent, they measure clearly distinct aspects of the phase space density [13, 14, 15].

For a fixed $\varepsilon$, all three centrality measures depend on the system size $N$. In contrast, the local recurrence rate $RR_v = k_v/(N-1)$ (i.e., the density of connections in the vicinity of a vertex $v$) is a non-extensive property (i.e., does not depend on $N$). Another non-extensive vertex property is the local clustering coefficient $C_v$, which measures the presence of closed triangles in the network and, hence, characterizes localized higher-order spatial correlations between observations. Since recurrence networks are spatial networks, structures resolved by spatial variations of $C_v$ correspond to a heterogeneous spatial filling of points. Specifically, high values of $C_v$ often coincide with dynamically invariant objects, such as unstable periodic orbits or, more generally, invariant manifolds [13, 14, 15].

In order to illustrate the above general statements, we consider the behavior of the mentioned vertex properties for realizations of the logistic map at different values of the control parameter $a$ (Fig. 1). One observes that the degree centrality is indeed directly proportional to the invariant density of the chaotic attractor, which explains the sharp increase of $k_v$ at the attractor boundaries and supertracks. $b_v$ shows low values (i.e., few shortest paths) close to the attractor boundaries, whereas the high values of $k_v$ along the supertracks coincide with low betweenness values. The latter observation can be understood as an effect of the increasing redundancy of vertices for shortest paths in high-density regions of phase space. We further note that the supertracks are also resolved by $C_v$, which is consistent with its interpretation as an indicator for dynamically invariant structures. Note that these considerations apply not only to maps, but also to continuous dynamical systems such as the Rössler, Lorenz [13, 14, 15], or Duffing systems (see Fig. 2).

3. Detecting dynamical transitions in time series

One of the main applications of the quantitative analysis of recurrence matrices is the identification of dynamical transitions in time series. Specifically, if corresponding quantitative measures are calculated for individual, mutually overlapping parts of the time series [11], it is possible to use their variation with time for identifying changes in the dynamics of the underlying system. In a similar way, one may argue that sufficiently strong changes in the geometric properties of the associated attractor in phase space can be detected by measures obtained from recurrence networks. Recently, it has been demonstrated that a corresponding approach works indeed very well for detecting bifurcations in one-dimensional maps as well as real-world paleoclimate time series [12, 15]. Specifically, the average path length $L$ has turned out to react sensitively to qualitative changes in the systems dynamics, whereas the global
clustering coefficient $C$ is a good indicator for the presence of regular (e.g., periodic) dynamics. In a similar way, it has been shown that network measures allow a better discrimination between periodic and chaotic dynamics in the parameter space of continuous-time dynamical systems than traditional recurrence quantification analysis [20].

In this work, we suggest another potential application for the quantitative analysis of recurrence networks. In economic time series, the behavior of the underlying system is typically influenced by both exogenous and endogenous shocks. Such shocks might be thought of not only causing abrupt changes in the mean and variance of the data set, but also in the qualitative appearance of its distribution. Considering the individual data as states in phase space, this implies that the geometric properties of the system are altered as well, which can be quantitatively characterized by measures computed from the associated recurrence networks.

Fig. 3 shows the temporal variations of global clustering coefficient and average path length of the recurrence networks obtained for the daily exchange rates between US Dollar and Euro over the last about 10 years (source: http://www.ecb.int/stats/exchange/eurofxref/html/index.en.html). In order to test the statistical significance, we additionally computed recurrence networks from surrogate data containing 10,000 random samples (containing the same number of data as the considered sliding windows) taken from the original data set [12, 15]. Our results reveal some interesting general features. For example, the pronounced minimum of the global clustering coefficient $C$ in 2004 (panel (C)) appears to coincide with a period of increased volatility in the data (panel (A)). In a similar way, we find a significant minimum of $C$ (panel (C)) in late 2001, possibly related with an increased market dynamics after the September 11 assaults. Multiple dynamical transitions, which appear to be related with change points in long-term trends, can be found in the years 2000-2001 and around early 2004, late 2005 and late 2006 when considering the variability of the average path length $L$ (panel (B)). While the detection of this type of transition is fairly trivial, we point out that our method also allows identifying more subtle transitions, e.g., by considering the exchange rates subjected to some appropriate detrending before further analysis. Note, however, that in case of the highly volatile multi-scale dynamics of financial time series, the features resolved after detrending crucially differ from those obtained from the original data. A similar behavior can be expected when additional embedding is used.

4. Summary

We have demonstrated that evolving complex networks constructed from the recurrence properties of dynamical
systems allow studying non-trivial properties of these systems in terms of network-theoretic measures. Local vertex properties of recurrence networks (in particular, the local clustering coefficient) can be used for identifying dynamically invariant objects such as unstable periodic orbits or, more generally, invariant manifolds, whereas their global network properties (especially average path length and global clustering coefficient) detect time intervals corresponding to dynamical transitions and highly regular or volatile dynamics. Our findings open a wide field for novel applications of complex networks in data analysis, which make recurrence networks a promising candidate for the analysis of a variety of different transdisciplinary problems.

Acknowledgments

This work has been financially supported by the German Research Foundation (SFB 555 project C4 and DFG project no. He 2789/8-2), the Max Planck Society, the Federal Ministry for Education and Research (BMBF) via the Potsdam Research Cluster for Georisk Analysis, Environmental Change and Sustainability (PROGRESS), and the Leibniz Society (project ECONS).

References

Clustering CNN devices for smart networks

Lambert Spaanenburg† and Suleyman Malki‡

† Department of Electronic and Information Technology, Lund University
P.O. Box 118, 22100 Lund, Sweden
‡Department of Electronic and Information Technology, Lund University
P.O. Box 118, 22100 Lund, Sweden
Email: Ispaanenburg@ieee.org, Suleyman@eit.lth.se

Abstract— Cellular Neural Networks promise an optimal implementation of non-linear systems. It is hard to keep that promise on platforms with limited capacity. Literature shows that a tiled processor array can be configured to an algorithm-specific processor to facilitate this in a network node. The paper discusses whether such can be scaled up towards smart networks.

1. Introduction

The beauty of Cellular Neural Networks (CNN) lies in the elegant coupling of the mathematical formulation to the silicon implementation. Not every algorithm works nicely on silicon. In fact, the performance gap between function and realization is rapidly growing. This ‘more than Moore’ gap has become a curse for areas where high speed has to be compromised to low power dissipation. Pervasive computing will bring networks of hundreds of ambient sensors, each as powerful as last year’s supercomputer. Clearly this dream can only be fulfilled if the computational power goes down by factors.

The promise of cellular networks is founded on the lack of global control. Where there is no need to schedule / monitor the overall operation, the realization may show up only short wires between synchronous gates and therefore a potentially higher clock rate. But it has not been trivial for CNNs to convert this potential to the real benefit within a large system and/or network.

For historical reasons the older CNN realization is the analog focal-plane image processor [1]. The light diodes capture the image in parallel, but getting the image out by line takes considerable time. At least, it is necessary to separate image extraction from transport by intermediate saving of the pixel values. Notable improvements in performance are further reached by pre-processing the pixels locally to decrease the communication load [2].

The first streaming CNN architectures are based on a pixel-flow mechanism, where a pixel-processing pipeline performs the iterations in time, finally writing the results of a single pixel operation back into memory [3]. The alternative is a tiled architecture, where the streaming is reduced to a pixel line FIFO [4]. Each FIFO element corresponds to a cell in a matrix of simple processing elements with an on-chip network for internal broadcasting. The network applies subsequent instructions while iterating to a stable result before returning to memory.

Recently a CNN processor is developed that can work in both modes. A reconfigurable CNN network is supplemented by program- and data stores, which can be loaded from the attached client processor. After loading, the network can independently perform pixel processing and feature recognition on the current image memory while communicating results with the client processor where higher-order image processing will take place.

Figure 1: Architecture of streaming CNN system.

The paper is composed as follows. First different scaling aspects of the proposed processor are discussed. Then we look at the modeling of boundary conditions by virtual cells and how these impact network scalability.

2. Scaling aspects of the single CNN node

The CNN equation is in its mathematical notation parametrizable, as the constructs are defined over linear ranges. The nature of the 'more than Moore' gap is essentially this lack of reflection on the potential of the target technology. For a network of devices it is mandatory that all platforms where a function can be executed, this function is executed with either the same or a pre-known performance. In this paper we will largely look at numerical aspects as these set the usability of the results.

Number values are not restricted on the algorithmic level, but will become limited when transferred to the implementation level. Algorithms do not take implementation details such as the effect of finite number representation on a hardware platform into account. The usual design approach starts by building an accurate and precise model, initially cast into software using double-length floating-point numbers for all the values. Realizing this model in a specific technology implies the need to limit the values in accuracy and/or precision.
Fixed-point number representation is usually not enough when the CNN handles dynamic behaviour. It is reported that 36-bits numbers are required to handle 2\textsuperscript{nd}-order differential equations. But this is not necessarily a maximum and one may wonder whether it is enough. Therefore the reasonable extension is the block floating-point format. It splits the requirement for precision and accuracy into two parts: the width of the mantissa sets the precision while the scale factor turns this into accuracy. In case of CNNs, full floating-point representation is not necessary as all the scale factors within the block of nodal template values are identical. Sofar a 5-bit scaling for an 8-bit mantissa has sufficed in experimentation.

3. Networking the nodes into a CNN

As a concept, the Cellular Neural Network promises a 1-cell-per-core solution. But even when the core can be made extremely small, the size of the network will be limited. For a focal-place processor, this is no issue as the network is implicitly at the size of the problem. But in general the problem has to be cut into parts, and in turn the parts have to be suitably mapped onto the platform. As a typical algorithmic approach, the solution is also dependent on the initialization. In the spatial implementation, such initial values are provided over the boundary of the structure. To properly model this in line with the nodal implementation of a cell requires a specialized node to deliver the boundary value. This overhead can be substantial.

![Figure 2: Boundary nodes have an incomplete communication cycle. Squares represent regular nodes while the dotted lines show which part of the packet path is missing. The node of interest is shaded.](attachment:figure2.png)

<table>
<thead>
<tr>
<th>Value</th>
<th>Fixed-point notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>u and y value</td>
<td>&lt;1:7&gt;</td>
</tr>
<tr>
<td>a and b coefficients</td>
<td>&lt;4:4&gt;</td>
</tr>
<tr>
<td>Bias</td>
<td>&lt;5:3&gt;</td>
</tr>
<tr>
<td>Multiplication results</td>
<td>&lt;5:11&gt;</td>
</tr>
<tr>
<td>State x</td>
<td>&lt;10:11&gt;</td>
</tr>
</tbody>
</table>

Table 1 Typical data representation of a digital DT-CNN. The notation \(<k:l>\) means that the number consists of \(k\)-bits integer part and \(l\)-bits fractional part.
Traditionally, the effect of boundary conditions is modelled by adding virtual nodes on the edges of the network. These virtual nodes either supply the boundary nodes with a predefined value (fixed boundary condition) or mirror the value of the boundary node itself (Zero-flux boundary condition). The problem here is further complicated by the asymmetry of the pre-scheduled communication pattern: boundary nodes experience different needs depending on their position in the network. Figure 2 (left) illustrates the disturbed communication cycle for top boundary nodes. The situation is even worse for the corner nodes (Figure 2 right). Actually, not only boundary nodes are affected by the incompleteness of broadcasting but even close-to-boundary nodes as well (Figure 3 left).

Employing the traditional approach of adding virtual nodes is not as simple as it may seem. It is unable to solve the problem entirely and adds on the network size. In any prescheduled communication scheme, virtual nodes should follow the sequence of sending (and eventually forwarding) of values that is accommodated by all regular nodes in the network. This works fine for close-to-boundary nodes (Figure 3 left), but the communication path is still incomplete for boundary nodes. It is clear from Figure 3 (right) that top boundary nodes will not receive any data in steps (4), (5) and (6). In other words, the partially asymmetric transfer cycle necessitates the existence of two (!) layers of virtual nodes to achieve completion. This situation exists for boundary nodes located on the bottom and on the right side of the network as well, but the left edge nodes need only one layer (Figure 2).

Hence, for an R(ow) x C(column) CNN, the number of virtual nodes is equal to \(4C + 3R + 12\). Each virtual node needs a router to send and forward packets, a local register and a simplified controller, which will require around 25 slices per node! This seriously degrades area utilization and there is a pressing need to replace the virtual nodes by a simpler mechanism that still completes the communication cycle.

We aim here for a total removal of the need for virtual nodes. This is possible by slightly refining the communication pattern of boundary nodes. Let’s consider top and bottom boundary nodes (Figure 2). The actions have to be performed in addition to the regular functionality of the node, mainly when a zero-flux boundary condition is used. For fixed boundary condition most of the sending/forwarding is redundant as all boundary nodes will need to store a single fixed value only that can be used instead of the received value.

Implementing the actions introduces the need for boundary nodes to, sometimes, send or receive two packets simultaneously, which requires a remarkable redesign of the nodal controller and the router in addition to the need of an extra register that keeps one value (W-value). Once again, different boundary nodes will require different refinements. This is of course better than the virtual nodes approach, but still increases the area considerably. A better solution makes use of the existing routing mechanism to forward boundary conditions. We call it ‘swing broadcasting’ as each boundary node will send its own value to one neighbouring boundary node and then to the other boundary node in the opposite direction. Due to the use of duplex lines between the nodes, the inter-nodal connections have to be idle for one time step in between (Figure 4). In this case, all boundary nodes will have the value of their neighbouring boundary nodes available locally. This requires two additional buffering elements to store the values, but the effect on area utilization is kept at a minimum. Overall, 3 time steps are introduced for each newly calculated y-value.

![Figure 3: Broadcasting scheme of close-to-boundary nodes (left) is incomplete but the situation is salvaged by adding a single layer of virtual nodes (middle). For boundary nodes more than one layer of virtual nodes is needed (right).](image)

![Figure 4: Swing broadcasting allows distributing of boundary conditions in three steps clock-wise (a) and anti-clock wise (c). For proper functionality on the duplex lines a separating idle step is introduced (b).](image)

4. Partition and Merge

But boundary nodes terminate the field of computation. Whether this is good or bad depends on the partition style. In an hierarchical partition each node represents a number of cells. By going down the hierarchy the problem size gets limited and each node represents less cells until eventually we have a 1-to-1 correspondence on the problem sized to the available platform. In a distributive partition we cut the problem thereby creating boundaries. This essentially disturbs the behavior. So simply slicing is only affordable when bouncing is not required by the interaction of the cells. Most of the image-processing applications are of this type.

Shared memory allows for inter-cell reaction without much change to the boundary concept. If new data is written to a boundary cell, this will automatically be
copied into shared memory. Reading a value from a boundary cell will be re-routed from shared memory, if previously invalidated. Such mechanisms are easily incorporated into the cache coherence protocol of a multi-core system. The system efficiency relies on the designed-in elimination of memory access conflicts. It is illustrative to judge the severity of this problem by looking at the relative size of the boundary. Given n-by-n functional cells we have 4(n+1) boundary cells. The difference \( n^2 - 4n + 4 \) gets quadratic larger with increasing network size and is already substantial for moderate values. For instance, with \( n = 20 \) we have 5 times more functional than boundary cells.

A shared memory has its main advantages in combination with general-purpose, pipelined cores. Such software-based solutions are known to be much slower than algorithm-specific integrated circuits. Even when both architectures are implemented on an FPGA, the speed difference of a factor 10 is easily incurred. But the use of standard, high-volume products produced in the latest technology will compensate for this, at least partly.

A modern FPGA is more like a logic-enhanced memory. When a network is partitioned, it creates boundary cells shared 2-by-2 on each cut. Clearly the overall system should work as if no cuts were made, or in other words no boundary cells are inserted. Further it is desirable that the boundary cells do not correspond to package pins.

This seems similar to the Boundary Test approach. In BT we have ports that we want to set, test & scan separate from the chip interior without introducing (many) additional pins. We already have the boundary chain but we do not want the ports to be each physical. In general, the ways to do this are based on multiplexing: (a) in space through a rotating buffer, or (b) in time through encoding.

5. Discussion

The previous methods have enabled the design of a configurable single node CNN architecture. But a single node is not enough to create an intelligent system. Next to interaction, where nodes perform actions based on directions from other nodes, we need reaction whereby also the ongoing actions are changed. The simplest example of such reactivity is where CNN vision nodes adapt the image parts to look at and the granularity to work with.

A typical smart vision sensor is comprised of three layers. On the lowest level we find the pure pixel operations. This is the need for a high data-handling rate usually leads to a dedicated pixel processor. The feature processing is performed to retrieve information. Finally, this is further condensed to knowledge by reasoning on the extracted features. For example, on the WiCa platform pixel-processing is performed by the IC3D stream-processing chip, while all higher levels are handled by a 80xx processor [11].

The principle benefit of a Cellular Neural Network is that as a computing paradigm it handles both the pixel processing and the feature extraction. Usually it provides also some basic reasoning. As the potential for parallel processing is extended to cover at least information gathering, it is in principle much faster. For the moment, a disadvantage remains that no theory on co-operating CNNs is available.

The hierarchical approach has been used in the development of an algorithm-specific integrated processor that runs simple integer software together with CNN programs [12]. The integer processor is hardly burdened with CNN-related activities but largely functions as network server and knowledge exchanger. This supports the creation of vision through multiple intelligent vision sensors in a low capacity communication network, in extension of arrays of bare cameras with a high-speed connection to a single server.

References


Hierarchical Feature Extraction for Dynamic Feature and Signature Tracking

Vilmos Szabo† and Csaba Rekeczky‡

†Dept. of Information Technology, Pazmany Peter Catholic University
Prater 50/A, Budapest, Hungary
‡Eutecus Inc.
Berkeley, California, USA
Email: szavi@digitus.itk.ppke.hu, rcsaba@eutecus.com

Abstract— The goal of this paper is to introduce an improved tracking framework, which exploits dynamic feature and signature selection techniques for data association models. It performs robust multiple object tracking in a noisy, cluttered environment with closely spaced targets. This method extends the back-end processing capabilities of tracking systems by creating a two-level hierarchy between the parallelly extracted features. These features are dynamically selected based on a spatio-temporal consistency weight function, which maximizes the robustness of data association, and reduces the overall complexity of the algorithm.

1. Introduction

Multiple object or target tracking is an important task in computer vision applications. However, it can become a challenging problem, especially if the object is in a dynamically changing environment. A number of computer vision applications could be characterized by two complex stages of processing. The first stage is the topographic image acquisition, which may include pre-processing, image segmentation, and post-processing. The second stage is a non-topographic sensing which includes feature-signature extraction, data assignment, and state-prediction. High resolution spatio-temporal detection can be accomplished using topographic or cellular processing hardware, such as the Cellular Neural Network (CNN) [1]. The multiple object tracking back-end is usually accomplished using serial Digital Signal Processors (DSP). Therefore, the numerical complexity of the tracking algorithm is crucial in order to meet the systems real-time demand. This paper focuses on object tracking using dynamic data association and its spatio-temporal signature analysis. Application areas may include traffic monitoring, vehicle navigation, automated surveillance and biological applications.

2. Dynamic Multiple Target Tracking Framework

Multiple target tracking can be defined as estimating the trajectory of objects in the image plane as they move around in the scene [2]. Generally, an object segmentation algorithm runs on each frame of the video flow in order to detect objects. This can be done on a CNN-like massively parallel topographic hardware to achieve high spatio-temporal resolution video flow processing. The detected objects are then assigned to consistent labels, called tracks [3]. The temporal analysis of tracks can be used to identify and select features that best represent each object. The final goal of target tracking is to determine the position of an object or a bounding box on each frame of the video sequence. Our algorithm follows a bottom-up approach:

1. Pre-processing of input video flow
2. Parallel image segmentation algorithm
3. Post-processing of segmented video frame
4. Image labeling
5. Object shape and appearance representation
6. Parallel image feature extraction
7. Image feature normalization and selection
8. Assignment of object to tracks based on dynamic feature selection
9. Feature signature analysis

Steps 1–3 can be implemented on CNN-type hardware. Pre-processing of each video frame is an important step to eliminate unwanted noise, and to condition the signal for further analysis. Throughout the evaluation, Gaussian filtering was employed that can be approximated on the CNNs resistive grid. The time or scale parameter depends on the amount of noise in the scene. The range of pixel intensity values was converted to \( I \in \{-1; 1\}^N \) where \( N \) and \( M \) are the width and height of the image. In case of color processing, each chromatic channel is processed separately (see subsection 2.3 Hierarchical Feature Extraction).

For post-processing, basic mathematical binary morphological [4] operators were used. The aim was to connect fragmented objects with the closing operation, and to clear individual pixels created by the “non-perfect” segmentation algorithm.

Steps 4–9 are typical serial DSP-like processing. Each of the connected components is labeled on the binary image. A number of features are extracted from the connected components for the dynamic tracking. The results of feature signature analysis (8) provide a feedback to the
dynamic feature analysis (6) in order to calculate the track consistency metric (see subsection 2.4 for details).

2.1. Motivation

The motivation for employing dynamic feature selection for multiple object tracking emanates from the need to reduce the complexity of data association steps of the overall algorithm. Let \( x \) and \( y \) be \( d \)-dimensional vectors, where each component corresponds to a feature value. The two most widely used distance metrics are the \( L_1 \) city block (eq. 1) and \( L_2 \) Euclidean (eq. 2) metrics.

\[
L_1 : d_1(x, y) = ||x - y||_1 \quad (1)
\]

\[
L_2 : d_2(x, y) = ||x - y||_2 = \sqrt{\sum_{n=1}^{d} (x(n) - y(n))^2} \quad (2)
\]

The best feature will provide the maximum interclass distance between objects. Increasing the feature space dimensionality will increase the discriminative power. However, noisy channels can decrease the robustness of the system. Therefore, the algorithm should try to select as few salient features as possible for data association. This decreases the number of features that need to be extracted. There are existing methods for dimensional reduction, such as Principal Components Analysis (PCA) [5]. These methods usually require a training set or block processing for dimension reduction. The feature selection method explained in this paper is a recursive one; it has a relatively low computational complexity and is able to successfully select a set of salient features in a changing environment.

2.2. Simulation Videos

The algorithm was evaluated on three computer generated video flows. The first video Scene 1 (Shapes) contains five dynamically changing objects. See Figure 1 for a demonstration on three objects. Each object is able to change its location, visibility, orientation, color, shape, noise, and inner-structure according to the following list:

- Location: [0–1]
- Visibility: [0–1]
- Orientation: [0–360°]
- Color: [red, green, cyan, blue]
- Shape: [circle, triangle, square, pentagon]
- Noise: [on, off]
- Inner structure: [dots, lines, concentric circles]

The second video flow is called Scene 2 (Bipeds). This scene contains walking humans with crossing and overlapping paths; they are in partial and full occlusion, entering and exiting the scene. The third video flow is called Scene 3 (Cars). The first two scenes contain non-rigid objects, while the third scene contains only rigid objects. Figure 2 shows actual frames from all three video flows. The noisy version of the simulation videos had SNR\(_{dB} = 15$.

2.3. Hierarchical Feature Extraction

The input image is highly redundant. The transformation, to reduce the dimensionality of input data while keeping relevant information content, is called feature extraction. The result of the segmentation algorithm is a binary mask, where every pixel corresponds to a background or foreground pixel. A set of features were extracted over the foreground that can identify and describe each object in a given frame. These features are grouped into six statistically independent main feature groups. The two-level feature extraction is summarized in Table 1. In the current framework, each main feature group can be weighted separately.

The position feature group contains features associated with the location and its derivatives of each connected foreground pixels on the mask image. The second group, describes the size of each object. The shape feature group contain features that represent the shape of an object such as eccentricity (ratio of length and width of an object), extent (ratio of the area of an object to the area of the bounding box) and solidity (ratio of convex area of an object to the area of the object).
Table 1: Summary of the two level hierarchical feature extraction used for the dynamic tracking framework.

<table>
<thead>
<tr>
<th>Feature Group</th>
<th>Subgroup</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Position</td>
<td>Location</td>
</tr>
<tr>
<td></td>
<td>Speed</td>
</tr>
<tr>
<td></td>
<td>Acceleration</td>
</tr>
<tr>
<td>2. Scale</td>
<td>Area</td>
</tr>
<tr>
<td></td>
<td>Major Axis Length</td>
</tr>
<tr>
<td></td>
<td>Minor Axis Length</td>
</tr>
<tr>
<td></td>
<td>Bounding Box</td>
</tr>
<tr>
<td>3. Shape</td>
<td>Eccentricity</td>
</tr>
<tr>
<td></td>
<td>Solidity</td>
</tr>
<tr>
<td></td>
<td>Extent (Opacity)</td>
</tr>
<tr>
<td>4. Structure</td>
<td>Euler Number</td>
</tr>
<tr>
<td>5. Texture</td>
<td>Variance</td>
</tr>
<tr>
<td>6. Color</td>
<td>Average Y Luminance Component</td>
</tr>
<tr>
<td></td>
<td>Average Cb Color Component</td>
</tr>
<tr>
<td></td>
<td>Average Cr Color Component</td>
</tr>
</tbody>
</table>

Table 1: Summary of the two level hierarchical feature extraction used for the dynamic tracking framework.

object to the area of the object). The texture group contains the variance feature which is extracted from the grayscale image. In order to extract the color information the image is converted to YCbCr color space, but various other color spaces can be used such as Hue, RGB, LUV, Yuv or Lab. Finally, each object is represented by a 17 dimensional feature vector (excluding speed and acceleration features because they are derived from the position). The values in the feature vector are normalized between 0 and 1 in order to make comparable measurements among each frame of the video sequence.

The choice of the feature set should be based on the requirements of a specific application area.

2.4. Dynamic Spatio-Temporal Feature Selection

In a real-time application, the number of features should be minimal to increase the speed of the system, but all relevant information must be kept. This can be done by creating a hierarchy among the features based on their confidence or robustness. The noisy feature channels should be filtered out. The tracking system consists of feature selection, data assignment, state space estimation, prediction and error correction.

2.4.1. Feature Selection

The feature selection is done by analyzing the spatial and temporal property of each feature channel. The “good” features are selected based on a spatio–temporal consistency metric. Let \( x_k^i \) and be the feature state space vector at frame \( k \) for the \( i^{th} \) object. Let \( Q_k^j(n) \) quality matrix (eq. 3) be the minimum of pair wise \( l_1 \) (eq. 1) distance of the current state space vector \( n^{th} \) component between the \( i \) and \( j^{th} \) objects.

\[
Q_k^j(n) = \min \{ d_k(x_k^i(n), x_k^j(n)) \mid (i > j) \}
\]  

The second term of the consistency metric is the inverse of the residual gradient magnitude of the previous state space estimation. Features that are well separated from each other and do not change much in time are preferred. The final consistency metric (eq. 4) is defined by a linear \( \mu \) parameter homotopy of the first part and the second parts. (The variable \( m \) donates the number of features.)

\[
C_k = (1 - \mu)Q_k + \mu \frac{1}{m \sum_{j=1}^{m+1} |x_k^j - x_k^m|}
\]  

C_k vector contains the quality measurement for each feature at a given time. Different feature selection strategies can be considered. A fix number of best features can be selected, or features can be selected above a given threshold level, resulting in a varying number of features for each frame. Section 3 - Performance Evaluation gives detailed comparison of the different feature selection strategies.

2.4.2. Data Assignment

The assignment of measurements to consistent tracks is accomplished using a combinatorial optimization algorithm called the Hungarian [6] method. Only selected features selected contribute to the calculation of the distance matrix. The assignment algorithm is used to match the current and predicted states together with minimal cost. The time complexity of the assignment algorithm is low order polynomial. More complex data association models can be applied, such as described in [7].

2.4.3. State Space Estimation

Interacting Multiple Model (IMM) was used as state estimation framework. Recursive steady-state Kalman filters [8] are used for the state prediction and correction phases. The prediction filters are also known as alpha, alpha-beta, alpha-beta-gamma filters. Particle filters may be used to improve the accuracy of the results [9].

3. Performance Evaluation

The evaluation of the tracking algorithm was performed on computer simulated videos. The black and white reference masks do not contain information about tracking individual objects. Therefore an object map is synthesized, where a unique color is to each object for track representation (see Figure 3). For each color value, the center of mass coordinates are extracted, which gave the reference tracks for the evaluation. Note that for Scene 1 (Shapes) the object ID mask is multiplied with the corresponding binary mask before evaluation.
The mean square error (MSE) is calculated between the reference track and measured tracks. The MSE can be calculated according to the following equation:

\[ MSE = \frac{1}{n} \sum_{i=1}^{n} d_i(Ref_i, Meas_i) + d_i(Ref_j, Meas_j) \]  

A total of three feature selection strategies were evaluated. The first is the Best-Feature selection strategy, where only one feature is selected. The second strategy is when all the features (All-Feature) are used by the algorithm. The third is the K-Dynamic selection, where a feature is selected above a given threshold level. This threshold level was set such that the quality of tracking approximately achieved the All-Feature selection strategy. Table 2 summarizes the MSE comparison measurements for the three feature selection strategies.

Table 2: Summary of MSE measurement between the reference and measured trajectories for the different feature selection strategies.

<table>
<thead>
<tr>
<th>Scene</th>
<th>Best-Feature</th>
<th>K-Dynamic</th>
<th>All-Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scene 1</td>
<td>5.4E-3</td>
<td>8.8114E-4</td>
<td>6.459E-4</td>
</tr>
<tr>
<td>Scene 2</td>
<td>2.9E-3</td>
<td>2.6E-3</td>
<td>2.5E-3</td>
</tr>
<tr>
<td>Scene 3</td>
<td>1.94E-4</td>
<td>1.7366E-4</td>
<td>1.698E-4</td>
</tr>
</tbody>
</table>

The K-Dynamic selection used an average of 2.76, 3.01, 3.41 features on average for Scenes 1-3 respectively.

Figure 4 shows the final object detection and tracking result. The images also include the bounding box and trajectory of each tracked objects.

Figure 4: Object detection and tracking results on the three simulation videos. Scene 1 (Shapes) frame: 215, Scene 2 (Bipeds) frame: 250, Scene 3 (Cars) frame: 200

4. Conclusion

The tracking framework uses a dynamic feature and signature selection method for multiple target tracking. This algorithm can be used to track objects in a changing environment after topographic CNN-like segmentation and hierarchical feature extraction. The algorithm arranges the parallelly extracted features into a hierarchy, based on their consistency measurement. The overall complexity is reduced by using only the relevant features for tracking the objects in the scene, which reduces the computational time demand. Performance evaluation on synthesized videos confirmed that instead of using the 17 dimensional feature vector dynamically selecting the best 3-4 features can result in as accurate tracking. Future work will include the development of more complex selection strategies and testing the algorithms on real video sequences.

References

Mapping of High Performance Data-flow Graphs into Programmable Logic Devices

Csaba Nemes†, Zoltán Nagy‡, Miklós Ruszinkó§, András Kiss† and Péter Szolgay†‡

†Faculty of Information Technology, Péter Pázmány Catholic University
Práter u 50/a, Budapest, Hungary
‡Cellular Sensory and Wave Computing Laboratory Computer and Automation Institute,
Hungarian Academy of Sciences, Lágyamányosi u 11, Budapest, Hungary
§Applied Mathematics Research Laboratory Computer and Automation Institute,
Hungarian Academy of Sciences, Lágyamányosi u 11, Budapest, Hungary
Email: nemcs@digitus.itk.ppke.hu, nagyz@sztaki.hu, ruszinko@sztaki.hu, kissa@digitus.itk.ppke.hu, szolgay@sztaki.hu

Abstract—In high-performance processors computation time and communication delay are comparable. Only those design methodologies can be successful which take care of the precedence of locality. In this article new design methodology is introduced which partitions the execution units and assigns a locally distributed control unit to each partition. Execution and control are relatively fast inside the partition and this results in a speed gain contrast to the global control unit where the fan out of the wiring can cause a slow operation. An optimization problem is described and an algorithm is developed which targets to find the optimal partitioning where fast local control units can be used with relatively small area increase. The optimal solution of the partitioning problem is NP complete [1] but a reasonable algorithm can be constructed for practical engineering applications. We have successfully designed a greedy algorithm and tested on few test cases.

1. Introduction

Having a computational problem defined on 2D or 3D array (NxM, NxMxL) and the operation on every element is described as a mathematical expression, acyclic data flow graph or UMF diagram [2]. The problem to be solved is how to map a computational problem on a virtual array to a given physical FPGA where area/processor (logic slices, DSP slices), on-chip memory (BRAM) and off-chip memory bandwidth are limited. Depending on the complexity of the operator a small amount of physical execution units can be implemented $n << NxM$ (in 2D case) or $NxMxL$ (in 3D case). The operator can be decomposed into small basic blocks which use either the logic resources (such as adders) or the dedicated resources (embedded multipliers) of the FPGA. The result of this process is a Physical Cellular Machine optimized for the given application. The optimization can be focused on speed, area, accuracy etc. Main components are the on-chip memory and the specialized execution unit.

Using current high speed DDR2/3 SDRAM and SRAM memories data read and write operations can be carried out in consecutive bursts. Additionally the available memory bandwidth might be fluctuating, therefore the execution unit should be halted during the computation if no data available.

The simplest and most area efficient solution of this problem is to use one global control unit to monitor the state of the I/O buffers and enable the operation of the entire system by using a global enable signal. The global enable signal has very high fan-out and is hard to route even if global wires are available on FPGA. As wire delay dominates over gate (LUT) delay on the current state-of-the-art FPGAs this solution results in very low operating frequency.

One possible solution of this problem is to create a data driven pipeline where a basic processing unit is halted automatically when no input data is available or the results cannot be processed by the next unit. Therefore local control unit can be added to every operator (adder, multiplier). In this case the control units are the simplest, but area requirements are significantly increased by the large number of FIFOs.

Alternative solution is to share the control unit among several basic processing units thus the FIFO buffers inside the groups can be eliminated significantly reducing area requirements. Determining the parameters of the groups carefully significant loss in operating frequency can be avoided.

2. Resources on an FPGA

The main configurable element of the new Xilinx Virtex family[3] is the Advanced Silicon Modular Block (ASMBL). The architecture is column based where each ASMBL column has specific capabilities, such as logic, memory, I/O, DSP, hard IP and mixed signal. By using different mix of the ASMBL columns domain specific devices can be manufactured. Currently four families are available optimized for different application areas: logic intensive (LX), logic intensive with serial transceiver (LXT), high performance DPS with serial transceiver (SXT), and em-
bedded processing (FXT). Due to the smaller transistor dimensions the total net delay is mainly determined by the wire delay, hence the CLBs of the Virtex-5 architecture are completely redesigned. In the new architecture traditional 4-input LUTs are replaced by 6-input LUTs. Each CLB is divided into two slices and every slice contains 4 6-input LUTs, 4 registers, and carry logic.

In the new FPGAs the simple multipliers are replaced by complex DSP blocks called XtremeDSP (DSP48E) slices. The heart of the DSP48E is a 25bit by 18bit 2's complements signed multiplier. It also contains a 48bit ALU unit with optional registered accumulation feedback. Additionally, hardwired 17 bit shift capability simplifies the construction of large multipliers, while optional pipeline registers enable even 550MHz operation. The currently available largest Virtex-5 device contains 1056 DSP48E slices, while the largest member of the recently introduced Virtex-6 family contains 2016 DSP48E slices.

3. SystemC and Mathematical representation

Using a high-level description language the computationally intensive algorithm can be efficiently described by using the data-flow model. This model can be transformed to an abstract mathematical graph representation, where operations and connections are represented by nodes and arcs of the graph respectively.

In case of distributed control units data driven pipelines are created where basic processing units are halted automatically when no input data is available or the results cannot be processed by the next unit. Synchronization of the processing elements is done by using FIFO buffers.

If we assign one control unit to every processing unit and attach a FIFO to each of its output we obtain a restricted case of the Kahn process networks [4] where independent processes are communicating over bounded FIFO channels. Each processing unit can be treated as a process because it reads its input from a FIFO attached to previous processor or the memory interface and writes the updated result into a FIFO. The enable signal and the FIFO control signals are local to the given processor; their states depend on the state of the connected FIFOs.

However it is more practical to partition the processing units into groups where each group has one control unit. In this case FIFOs inside the locally controlled groups can be omitted. The disadvantage of this solution is that each control unit should handle more FIFOs, which results in more complex control logic and smaller operating frequency as shown in Figure 1.

There is a design trade of between the speed of the control unit and the area requirements of the circuit. Larger partitions have more inputs and outputs which results in a slower control unit, on the other hand small partitions increase the area requirements of the circuit because of the overhead generated by the larger quantity of FIFOs. We can formulate an optimization problem where we would like to enlarge the size of the locally controlled groups as long as the operating frequency of our control blocks does not limit the operation of the entire circuit.

Mathematical formulation of the partitioning problem:

1. Optimization: Make partitions from the nodes of a directed acyclic hyper graph where the number of cut edges are minimal. (In this case a hyperlink can be cut many times.)

2. Constraint: Any given partition shall have less than or equal connection to the other partitions than a given upper threshold.

Finding the optimal solution is an NP-complete problem, however our goal is to find a reasonable solution in polynomial time. The result of the algorithm will be an optimized graph, where the control is local and data driven. According to our experiments one control unit can handle 10 input/output FIFOs without decreasing the expected 450MHz operating frequency of the entire data path significantly (see Figure 1).

In our implementation the input of the algorithm is an initial solution of a computationally intensive task implemented in C++ via the SystemC library [5]. The only restriction for the implementation is that the classes of the modules have to inherit our interfaces as well. This is an elegant way to extend the SystemC model with some extra information without the modification of the library itself. While the modeling features of the SystemC library are still available, the extra information is used to build up the graph representation via the Lemon Graph Library [6].

4. The proposed partitioning algorithm

Hereby we propose a heuristic greedy algorithm for the previously described partitioning problem. While the pseudo code of the algorithm is shown in Algorithm 1. the key steps are also summarized below:
Algorithm 1

function runPartitioning(G,T):
1: Assign a number to each node, which indicates the number of cycles required for the data to reach the given node through the pipeline.
2: Sort the nodes based on the assigned numbers.
3: repeat
4: Create a new partition P.
5: Move the lowest-numbered node, which is unpartitioned to P.
6: growSubgraph(P,T)
7: until there is any unpartitioned node

function growSubgraph(P,T):
1: for all N nodes which gets input from one of the nodes of P do
2: P1 := P
3: Move N to partition P1.
4: Move all ancestor nodes of N which is unpartitioned to P1.
5: end for
6: if the number of the incoming and outgoing arcs of the subgraph \( \leq T \) then
7: P := P1
8: growSubgraph(P,T)
9: break
10: end if

1. The input of the algorithm is a directed acyclic hypergraph.
2. Based on the delays of the arithmetic units different levels can be assigned to the nodes. These levels indicate the number of cycles required for the data to reach the given node through the pipeline.
3. Based on the levels the nodes are numbered. The order of two nodes on the same level are arbitrary but nodes on lower level always have lower numbers than the ones on higher levels.
4. The algorithm starts from the lowest-numbered node and a subgraph is grown from this node. After the subgraph cannot be grown further the first partition is created from the nodes of the subgraph.
5. In every upcoming iteration the lowest-numbered node id selected which has not been partitioned yet. If there are no more unpartitioned nodes the algorithm ends.
6. The subgraph is also created iteratively. In the first iteration the subgraph contains only one node. In every upcoming iteration the algorithm tries to deepen the subgraph by selecting one node “below” the subgraph and includes all the ancestors of the selected node which have not been partitioned yet.

7. Extension of the subgraph is only successful if the number of incoming and outgoing arcs of the subgraph does not exceed the upper threshold.
8. If no node can be selected below the subgraph which is suitable to a successful extension the subgraph cannot be extended and the algorithm is continued from step 4.

5. Examples

5.1. Simple SystemC code and its graph representation

Simple SystemC code fragment and an equivalent graph representation is shown in Figure 2 and Figure 3 respectively. Sum and Prod modules are base modules which are already implemented in SystemC. Instances of these modules will be the nodes of the graph. Signal1 and signal2 are wires corresponding to the interconnections.

5.2. Base template operation of the CNN state equation

Our first test case was the frequently used 3x3 template operation of the CNN state equation [7]. It can be regarded as a convolution with a 3x3 kernel where 9 state values should be multiplied by 9 template parameters. The final result is computed by summing the partial results and the old state value. However this example is relatively simple and has a limited size, it is ideal for the demonstration of our algorithm. The result of the algorithm is shown in Figure 4. The first partition (partition #1) is grown from the top-left node of the graph (node 1). The algorithm extends the partition by stepping one level down and recursively selecting all the ancestors of the newly founded node in every iteration, therefore nodes 2 and 12 are added to the partition in the first iteration. Partitions are extended as long as the number of input/output arcs are still smaller than 10.
which was determined earlier as an upper threshold. In this example the first two extensions are successful. The third extension failed because the extended partition would have 17 input/output arcs. The second partition (partition#2) is grown from the next non-partitioned node (node 5) which has the lowest number.

Area requirements and operating frequency of the optimized control unit are shown on Table 1. The optimized control unit requires less additional area than fully distributed control unit but operates on higher frequency than global control unit.

### 6. Conclusions

A design methodology is described to implement customized high-performance data-flow architectures using distributed control unit. A data-flow graph of a mathematical expression constructed from a high-level language description is given. An optimization problem has been described to efficiently partition the execution units between control structures without significantly increasing the area of the circuit or decreasing the operating frequency. A heuristic algorithm has been proposed to give an affordable solution. The number and the size of the FIFOs are optimized in the design process to reach high speed with small increase in implementation area. The operation of the algorithm was presented by optimizing a simple CNN state equation example.

### Acknowledgments

This paper was supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences.

### References


An Improved Emulated Digital CNN Architecture for High Performance FPGAs

L´aszl´o F¨uredi†, Zolt´an Nagy‡, Andr´as Kiss† and Péter Szolgay†‡

†Faculty of Information Technology, P´eter P´azm´any Catholic University
Pr´ater street 50/a, Budapest, Hungary
‡Cellular Sensory and Wave Computing Laboratory Computer and Automation Institute,
Hungarian Academy of Sciences
L´agym´anyosi street 11, Budapest, Hungary
e-mail: furla@digitus.itk.ppke.hu, nagyz@sztaki.hu, kissa@digitus.itk.ppke.hu, szolgay@sztaki.hu

Abstract—Cellular Neural Network (CNN) is a prototype Single Instruction Multiple Data (SIMD) like architecture, where the basic operation of this architecture is the weighted sum calculation. The emulated digital CNN-UM architecture was implemented and tested on different kind of array computers, eg. Cell Broadband Engine (Cell BE), Field-Programmable Gate Arrays (FPGAs), for utilizing the high performance of the digital microprocessors. The arithmetic unit of the original Falcon architecture was mainly optimized for the special features of the Xilinx Virtex-II architecture. Implementing the same architecture on the new Digital Signal Processor (DSP) optimized FPGAs will be inefficient. In order to achieve the highest possible performance the dedicated elements of the new FPGA should be fully utilized. Therefore an improved arithmetic unit should be designed. According to the requirements of the new arithmetic unit the input data structure and the data-flow of the processor should be redesigned. Additionally the interconnection of the Falcon processing elements are optimized to utilize the specialized interconnect resources on the FPGA. Compared to the original Falcon processor with the modified implementation on the new FPGA families the clock frequency can be improved by 20 percent. Additionally the area requirement of the arithmetic unit is significantly reduced by utilizing the special features of the DSP blocks.

1. Introduction

In high performance processors the operation delay and the wiring delay is comparable. This effect is explicable with the scaling down of the technology. The increase of clock frequency the signal does not have enough time to reach the destination in one cycle. The adjacent computational elements can communicate faster because in short range the wiring delay is not significant. The effective architecture design’s prime aspect is the locality precedence. This precedence is studied in an emulated digital CNN-UM implementation. A multi-layer CNN array can be used to solve the state equation of complex dynamical systems [1][2]. The CASTLE and Falcon emulated digital CNN chips were designed to reach this goal [3][4], where the accuracy, template size, cell array size and the number of layers can be configured. This paper describes synthesis and implementation methods used for the modified Falcon processor array on Virtex-5 and Virtex-6 FPGAs. The Falcon architecture is designed to solve the full signal range model of the CNN cell [5][6].

\[
\dot{x}_{i,j}(t) = \sum_{k=0}^{2n} \sum_{l=0}^{2n} A_{k,l} \cdot x_{i+k-n,j+l-n}(t) + \sum_{k=0}^{2n} \sum_{l=0}^{2n} B_{k,l} \cdot u_{i+k-n,j+l-n}(t) + I_{i,j}
\]

where x, u and z are the state, input and the bias values of the CNN cell, n is the neighbourhood size, A is the feedback, B is the feed forward template. The templates are \((2n+1)\times(2n+1)\) sized matrices. The state equation of the CNN array is solved on the Falcon architecture by forward Euler discretization. The \(t\) time step value can be inserted into the templates A and B, these modified templates are denoted by \(\hat{A}\) and \(\hat{B}\). Usually the input values do not change for several time steps so the state equation (1) can be partitioned into two parts, the feedback (2) and the feedforward part (3).

\[
x_{i,j}(m+1) = \sum_{k=0}^{2n} \sum_{l=0}^{2n} \hat{A}_{k,l} \cdot x_{i+k-n,j+l-n}(m) + g_{i,j}
\]

\[
g_{i,j} = \sum_{k=0}^{2n} \sum_{l=0}^{2n} \hat{B}_{k,l} \cdot u_{i+k-n,j+l-n} + h \cdot I_{i,j}
\]

The problem to be solved how to map the computational problem defined in (2) and (3) on a virtual array to a given physical FPGA where area/processor (logic slices, DSP slices), on-chip memory (Block Random Access Memory (BRAM)) and off-chip memory bandwidth are limited. Depending on the complexity of the operator a small amount of physical execution units can be implemented \(n << N\times M\) (in 2D case) or \(N\times M\times L\) (in 3D case). The operator can be decomposed into small basic blocks which
use either the logic resources (such as adders) or the dedicated resources (embedded multipliers) of the FPGA. The result of this process is a Virtual Cellular Machine optimized for the given application. The optimization can be focused on area, accuracy, speed, dissipated power etc. Main components are on-chip memory and the specialized execution unit.

2. The resources on an FPGA

The main configurable elements of the new Xilinx Virtex family is the Advanced Silicon Modular Block (ASMBL)[7]. The architecture is column based where each ASMBL column has specific capabilities, such as logic, memory, Input/Output, DSP, hard IP and mixed signal. By using different mix of the ASMBL columns domain specific devices can be manufactured. In the new architecture traditional 4-input Look-up Tables (LUTs) are replaced by 6-input LUTs. Each configurable logic block (CLB) is divided into two slices and every slice contains 4 6-input LUTs, 4 registers, and carry logic. In the new FPGAs the simple multipliers are replaced by complex DSP blocks called XtremeDSP (DSP48E) slices, it supports over 40 dynamically controlled operating modes including: multiplier, multiplier-accumulator, multiplier-adder/subtractor, three input adder, barrel shifter, wide bus multiplexers, wide counters, and comparators. The heart of the DSP48E is a 25bit by 18bit 2’s complements signed multiplier with full precision 43-bit result. It also contains a 48bit Arithmetic Logic Unit (ALU) with optional registered accumulation feedback and support for SIMD operations. Additionally, hard wired 17 bit shift capability simplifies the construction of large multipliers, while optional pipeline registers enable even 550MHz operation. The number of DSP48Es is 1056 in a Virtex-5 SX240T and 2016 in a Virtex-6 SX475T FPGA. The other key configurable elements are the interconnect wires. In the contribution we especially focus on minimization of wire delays.

3. Architectural improvements

The new FPGA families have much more resources than the Virtex-II FPGA which was used for the first implementation of the Falcon processor. For solving the discretized version of the CNN state equation a large number of multiplication is needed which can easily and efficiently implemented by using the dedicated elements (multipliers or DSPs) of the FPGAs. The aviable dedicated resources of the different FPGAs can be seen on Figure 1. Scaling up the original Falcon architecture on the new FPGAs in terms of the multipliers shows that on new FPGAs there are not enough configurable logic resources. If 32 original Falcon processor cores are implemented on the Virtex-II 3000 FPGA 94 percent of the configurable logic blocks and all of the multipliers can be utilized. Examining the aviable resources on the Virtex-5 SX50T, Virtex-5 SX240T and Virtex-6 SX475T devices 228, 182 and 140 percent of aviable logic block is required to implement the orignal Falcon processor when all multipliers are utilized as shown in Figure 2. The question is how to arrange the computation to use all multipliers while not overusing configurable logic blocks to implement the Falcon architecture on new FPGAs. The new modified type of architecture is shown on Figure 3, where the mixer and arithmetic units were changed. With these changes which will be described in the next sections all of the built-in DSP48E slices can used and the configurable logic block requirement of the processor core is also reduced.

3.1. Modified Mixer Unit

The structure of the mixer unit is shown in Figure 4. This unit contains one block of shift registers to store a window around the currently processed cell and two additional
that the DSP slices allow. Depending on the speed grade of the FPGA the operating frequency of the arithmetic unit can reach 550MHz on the Virtex-5 and 650MHz on the Virtex-6 FPGAs. Only one external element a register is required to store the feedforward value of the computation and it comes from the memory unit. The modified arithmetic unit is shown in Figure 5, and it can be used in pipelined mode.

4. Performance

Performance of the modified Falcon processor is compared to the speed of the software simulation. In the software simulation Intel Core 2 Duo E8400 and IBM CellBE processors are used. To simulate CNN array functions of the Intel Performance Primitives - Image Processing Library (IPP IPL) was used to help to optimize image- and vector-processing tasks. Performance of the software simulation depends on the size of the cell array. If the size is larger than 688×688 the performance drops to a lower level, due to the memory bottleneck and L2-cache memory occupancy. Even in a single Falcon processor configuration 38 percent performance improvement can be achieved compared to Intel Core 2 Duo processor. The easy scalability of the array makes it possible to connect several modified Falcon processor cores on one FPGA and get even more performance. Using the previously described architecture and utilizing all the 224 modified Falcon processors on the Virtex-6 FPGAs 145.6 billion cell iteration per second computing performance can be achieved. Our Virtex-6 FPGA based solution is 364 times faster compared to a high performance microprocessor, using all of the modified Falcon processors during the computation. Compared to a high performance Intel Core 2 Duo microprocessor, for a 1024 pixel ×1024 pixel picture instead of 38 template execution per second, 13885 template execution per second can be used with the Virtex-6 Falcon implementation.

5. Conclusions

An improved emulated digital CNN-UM architecture implementation was successful on our prototyping boards,
using the Virtex-5 SX50T and Virtex-5 SX240T FPGA from Xilinx Inc. and implementation in simulation using the Virtex-6 SX475T FPGA. Our solution was optimized to the special requirements of the Virtex-5 and Virtex-6 FPGAs. The main parameters of the architecture is described and compared to the parameters of the software simulation of the CNN full signal range modell running on high performance processors such as Intel Core2Duo and IBM Cell.

6. Acknowledgements

This paper was supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences.

References


| Table 1: Comparison of different implementations |
| Implementations | Intel Core 2 Duo | Cell Processor 8 SPEs | XC5VSX50T | XC5VSX240T | XC6VSX475T |
| Implementation type | Software (Intel IPP)[8] | Software (Cell SDK) | FPGA | FPGA | FPGA |
| Technology (nm) | 45 | 65 | 65 | 65 | 40 |
| Clock Frequency (MHz) | 3000 | 3200 | 550 | 550 | 650 |
| Number of Processing Elements | 2 Cores | 8 SPE | 32 FPE | 117 FPE | 224 FPE |
| Million cell iteration/s | 400 | 3627 | 17600 | 64350 | 145600 |
| Speedup | 1 | 9 | 44 | 160 | 364 |
| Power Dissipation (W) | 65 | 85 | 16 | 59 | 102 |
| Area (mm²) | 107 | 2×253 | N/A | N/A | N/A |
Fractal Basins and Boundaries in 2D Maps inspired in Discrete Population Models

Danièle Fournier-Prunaret†, Ricardo López-Ruiz‡

†LATTIS-INSA, LAAS-CNRS, Université de Toulouse
Toulouse, France
‡DIIS-BIFI, Universidad de Zaragoza
Zaragoza, Spain
Email: danielle.fournier@insa-toulouse.fr, rilopez@unizar.es

Abstract—Two-dimensional maps can model interactions between populations. Despite their simplicity, these dynamical systems can show some complex situations, as multistability or fractal boundaries between basins that lead to remarkable pictures. Some of them are shown and explained here for three different 2D discrete models.

Keywords: fractal, basin, two-dimensional map, fuzzy boundary.

1. Introduction

Two-dimensional maps can be used to model interactions between two different species. Such applications can be considered in Ecology, Biology or Economy [1, 2]. Generally, real systems consist in a large number of interacting species but the understanding of the behaviour of such systems in the low dimensional case can be of great help as a first step attempt. In this work, we consider two-dimensional (2D) models based on logistic multiplicative coupling [3] where complex behaviours occur such as multistability phenomena, fractal basins of attractors and fractal boundaries between basins [4, 5]. These phenomena lead to some remarkable graphical representations in the phase space plane. In Section 2, we recall three of these considered 2D models. In Section 3, we emphasize these models, which permit to obtain fractal basins. Section 4 is devoted to multistability phenomena and fuzzy or fractal boundaries between basins.

2. The models

The first considered model is the noninvertible 2D map $T_1$ defined by:

$$\begin{align*}
x_{k+1} &= \lambda(3x_k + 1)y_k(1 - y_k) \\
y_{k+1} &= \lambda(3y_k + 1)x_k(1 - x_k)
\end{align*}$$

(1)

where $\lambda$ is a real control parameter, $x$ and $y$ are real state variables. Previous studies of (1) have been done in [5]. This model is the symmetrical case of a 2D model.

![Image of fractal basins and boundaries](image.png)

Figure 1: Fractal basins of two order 2 cycles for the map $T_1$ (1), one basin is in yellow, the other in red.

![Image of enlarged fractal basins](image.png)

Figure 2: Enlargement of Figure 1, Z1 area.
The second model is the noninvertible 2D map \( T_2 \) defined by:

\[
\begin{align*}
    x_{k+1} &= \lambda(3y_k + 1)x_k(1 - x_k) \\
    y_{k+1} &= \lambda(3x_k + 1)y_k(1 - y_k)
\end{align*}
\]  

(2)

where \( \lambda \) is a real control parameter, \( x \) and \( y \) are real state variables. The map (2) is also inspired in the symbiosis case [6] by including a time asymmetric feedback. Previous studies of (2) have been presented in [7].

The third model is the noninvertible 2D map \( T_3 \) defined by:

\[
\begin{align*}
    x_{k+1} &= \lambda(-3y_k + 4)x_k(1 - x_k) \\
    y_{k+1} &= \lambda(-3x_k + 4)y_k(1 - y_k)
\end{align*}
\]  

(3)

where \( \lambda \) is real, \( x \) and \( y \) are real state variables. The map (3) corresponds to a competitive interaction between two species [8]. In each model, \( \lambda \) measures the strength of the coupling.

Figure 3: Enlargement of Figure 1, Z2 area.

Figure 4: Enlargement of Figure 3, Z21 area.

Figure 5: Basin of a fixed point for the map \( T_2 \) (2).

Figure 6: Enlargement of Figure 5. Fractal structure is observed.
3. Fractal basins

Figures 1-4 show the basin of two coexisting attractors and successive enlargements in the case of the map $T_1$, each attractor being an order 2 cycle; one basin is in yellow, the other one in red. Basins are symmetrical, non-connected and fractal. Successive enlargements of the 2-dimensional phase space on the diagonal ($Z_1$ area) and on $Z_2$ area show auto-similarity properties. The shape of each enlarged piece of basin is similar to the precedent piece with an alternation between the successive locations of symmetrical yellow and red basins. Such a shape can be explained by using the critical manifolds of (1) [4, 5].

Figures 5-6 show the fractal basin of a fixed point for the map $T_2$. This basin is non-connected. Figure 6 shows clearly the auto-similarity property. The appearance of such a fractal basin can also be explained by using the critical manifolds [4, 7].

Figures 7-8 show a fractal basin for the map $T_3$. The attractor is chaotic and it is not represented on the Figures [8]. As in the case of the map (1), the successive auto-similar pieces of the basin are located on the diagonal. The fractalization of the basins can be understood by means of the critical manifolds [4, 8], as in the case of the maps (1) and (2).

\[ \lambda = 1.2 \]

Figure 7: Fractal basin for the map $T_3$ (3).

4. Multistability and Fractal boundaries

Multistability among several attractors is very frequent in these models. As it can be seen in the Figures 9-12, the basins can take complex and fractal forms. Figure 9 shows the shape of the basins of three different attractors. The boundary among those basins is fractal, due to the accumulation and the entanglement close to the boundary of the square $[0, 1]^2$. Figure 10 represents the basin of two different attractors, which are order 3 invariant closed curve (ICC), for the map $T_2$. Then we obtain six different basins in $T_2^3$ for each piece of the two co-existing order 3 ICC in $T_2$. Some of the basins can be riddled, that is, successive zooms of a basin zone cannot differentiate the boundaries between the basins of the different attractors. Such basins are shown on Figures 11-12. There are two chaotic attractors, which are order 52 chaotic rings. They are called weakly chaotic because of the greater Lyapunov exponent, which is slightly positive [7].

\[ \lambda = 1.2 \]

Figure 8: Enlargement of Figure 7.

References


Figure 9: Basins of an order 29 cycle and two order 3 invariant closed curves (ICC) for the map $T_2$.

Figure 10: Basin for $T_3^2$ of 2 order 3 ICC.


Figure 11: Riddled basin of two order 52 weakly chaotic rings (WCR) for the map $T_3$.

Figure 12: Enlargement of Figure 11, BZ area.

Accurate Formulas Locating Unstable Periodic Points in Chaos

Tetsushi Ueta and Kei Nagao

†Center for Advanced Information Technology, Tokushima University
2-1, Minami-Josanjima, 770-8506 Japan
Email: tetsushi@is.tokushima-u.ac.jp

Abstract—We have studied fractal nature of patterns given by the directional coloring for chaotic attractors. With this method, unstable periodic points are visualized as concentrated points or crossing points in the state space. A fractal pattern around the specific unstable fixed point are identified as a formula derived from relationship among neighbor unstable periodic points. In this paper, we try to model an accurate formula giving locations of unstable periodic points. Higher periodic points are predicted by the obtained formula. Some numerical examples are shown.

1. Introduction

Given two-dimensional discrete dynamical system, it is a burden toil to pick up unstable periodic points (abbr. UPPs) embedded in a chaotic attractor. We have studied fractal nature of patterns given by the directional coloring for chaotic attractors. With this method, UPPs are visualized as concentrated points or crossing points in the state space. A fractal pattern around the specific UPP are identified as a formula derived from relationship among neighbor UPPs. In this paper, we try to model an accurate formula giving locations of UPPs. Higher periodic points are predicted by the obtained formula. Some numerical examples are shown.

2. Invariant Patterns

Let us consider the following two-dimensional irreversible discrete system:

\[ x_{k+1} = f(x_k) \]  

where, \( x = (x, y) \), and

\[ f(x_k) = \left( \begin{array}{c} y_k + ax_k \\ x_k^2 + b \end{array} \right). \]

It exhibits a chaotic attractor at \( a = 0.4, b = -1.24 \), and theoretically many UPPs are embedded within it. Figure 1 show an 8-periodic and a 40 periodic orbits in the chaotic attractor. In general, locating UPPs are not so easy task for a given chaotic attractor and parameter values even if the system equation is explicitly obtained.

The directional coloring method\[1\] is a visualization scheme of chaos attractors that each mesh point in the state space is colored by the direction between the current point and its \( n \)-mapped point. Figure 2 shows the directional coloring result for Eq. (1) with \( a = 0.4, b = -1.24 \). Among (a) to (c), it is confirmed that a portion circled by a dashed line rotates and shrinks about the fixed point located in the center as \( n \) increases. In the past works\[2][3\], by using image processing technology, it is clarified that color concentrated or crossing points can be candidates UPPs. Figure 3(a) shows a magnified picture around the fixed point \( x^* \) with evaluation of each 37-mapped points by directional coloring. The color-crossing points are specified by the arrows. points in which every color is terminated is specified by arrows, and they are candidates of UPPs. In fact, by using error collection method, accurate locations of UPPs are obtained from these candidates\[1\]. Notice that the point labeled by \( c_{37} \) is the nearest 37-periodic unstable point for the fixed for \( x^* \). Figure 3 (b) and (c) show the locations of \( c_{38} \) and \( c_{39} \) for \( n = 38 \) and \( n = 39 \), respectively. As \( n \) increases, patterns are rotated, shrunk, and the new UPP \( c_n \) is generated.

The rough location of \( c_n \) is computed by an edge detection method\[3\], and a further error correction is done by Newton’s method. We set an error tolerance for \( c_n \) is less than \( 10^{-15} \).

3. Fractal in patterns

Let us define \( \epsilon_n = \|c_n - x^*\|_2 \). This \( \epsilon \) shrink exponentially as \( n \) increases. In Ref. [3], we derived a formula of the
Figure 2: Patterns obtained by the directional coloring for Eq. (1).

Figure 3: Pattern rotation and new $c_n$ generation. Arrows indicate other candidate of UPPs.

Figure 4: The definition of $\epsilon_n$ and $\theta_k$.

In Fig. 5(a) shows actual value of $|\epsilon_n|$ and its fitted line. The red line is approximated as $\log_{10} |\epsilon_n| \approx ax + b$. A good agreement between them is confirmed.

Let us define $\theta_n$ as an angle determined by $\epsilon_{k+1}$ and $\epsilon_k$. Plotted points in Fig. 5(b) indicate actual values of $\theta_n$. The points alternative jump up and down, thus it makes like a quasi periodic wave form (beat). We attach two sinusoidal curve to fit the envelope of the wave. The green and red lines are modeled as:

$$\theta_n = (-1)^n A \sin(\omega n + \phi) + b$$

(4)

where, $A = 0.303$, $\omega = 0.306$, $\phi = -0.588\pi$, $b = 1.42$. Over $n > 25$, plotted points and lines have a good agreement. From Eq. (4), the averaged angular is constant $b \approx 0.45\pi$; the dashed line in Fig. 5(b). These results specifies enough evidence of an existence of a logarithmic spiral, thus fractal nature is confirmed in the given chaotic attractor.

Many color-crossing points observed in Fig. 3 are simultaneously generated by increment of $n$. Since relative assignment of these points can be located from $c_k$ in the invariant pattern, it is possible to locate and enumerate other UPPs by using an appropriate image processing.

Figure 5: Distributions of the radius $\epsilon_n$ and angle $\theta_n$ by Eq.(3) and (4). Plotted points indicate numerical values of $\epsilon$ or $\theta_n$.

4. Locating of UPPs

As an application of above approximation, in this paper, we try to derive a formula that locates the UPPs embedded in a chaotic attractor. Some modifications are applied for Eqs. (3) and (4) by using more precise fitting scheme, and we have the following equations:

$$\tilde{c}_n = x^* + \sigma_n \left( \cos S\theta_n \right)$$

$$\sin S\theta_n = \psi + \sum_{i=0}^{n-1} \theta_i$$

(5)

where, $\psi$ is the absolute angle between $x^*$ and $c_0$. Then $\sigma_n$ and $\theta_n$ are given as follows:

$$\sigma_n = 10^{ar+b(-1)^nA \sin(\omega n + \eta_1)}$$

$$\theta_n = (-1)^n B \sin(\omega_2 n + \eta_2) + d$$

(6)

These functions can specify accurate locations of UPPs $\tilde{c}_n$ by only substitution of $n$. 
4.1. Example 1

Suppose we fix the parameter values for Eq. (1) as \(a = 0.4, b = -1.24\), then we obtain the following parameters by using fitting method: \(a = -0.1156, b = 0.4, A = 0.07, \omega_1 = 0.312, \eta_1 = 0.16\pi, B = 0.31, \omega_2 = 0.306, \eta_2 = -0.572\pi, d = 1.418\). Figure 6 shows matching of \(\tilde{c}_k\) and \(c_k\). The error between them is invisible.

It is noteworthy that the frequency components \(\omega_1\) and \(\omega_2\) are independent for the multiplier (eigenvalues) of the fixed point \(x^*\).

![Figure 6: Relationship between \(n\) and \(\epsilon_n\) for Example 1. Plotted points indicate numerical values of \(\epsilon\).](image)

Both \(\epsilon_k\) and \(\theta_k\) are affected by sinusoidal functions of \(k\), but they are not synchronized each other. We once tried an amplitude modulated wave for \(\epsilon_k\) and \(\theta_k\) models, but it failed. Two alternative (anti-phase) sinusoidal function are essential.

Now we obtain an analytic formula that gives UPPs in the chaotic attractor. If we want to know the location of \(c_k\), the formula resulting the accurate location with specifying only \(k\) is very useful.

Figure 7 shows locations of UPPs. In this simulation, we use \(\theta_0\) instead of \(\psi\). Each vertex of the red line show the location of \(c_k\) whose accuracy is guaranteed by Newton’s method. While vertices on the black line are locations \(\tilde{c}_k\). Please note that these lines do not show the part of the solution orbit like Fig. 1, but demonstrate a fractal nature of the chaotic attractor. In fact, a line in Fig. 7 forms a logarithmic spiral given by Eq.(6), i.e., \(c_k\) and \(\theta_k\) surely keep a certain scale. Note also that \(\psi\) in Eq. (5) is initialized by the value of \(c_{40}\) for this case.

4.2. Example 2

When we fix the parameters as \(a = -0.1, b = -1.7\), a different invariant pattern of the directional coloring is shown in a chaotic attractor. Figure 9 depicts invariant patterns with \(n = 34\) and \(n = 35\) of the directional coloring around a UPP \(x^* = (-0.865097, -0.951606)\). The nearest radical point \(c_k\) is also an unstable \(k\)-periodic point.

For this example, Eq.(6) also acts reasonable. We have a very good fitting between \(\epsilon_n\) and \(c_n\) with this model. The fitting parameters are as follows: \(a = -0.1185, b = 0.41, A = -0.07, \omega_1 = 0.08, \eta_1 = 0.477\pi, B = 0.27, \omega_2 = 0.08, \eta_2 = 0, d = 1.61\).

Figure 10 expresses agreement between \(\tilde{\epsilon}_k\) and \(c_k\). The error between them is invisible. Figure 11 shows estimation errors defined as follows:

\[
\begin{align*}
\epsilon_x &= \log_{10} ||\epsilon_n|| - \log_{10} ||\tilde{\epsilon}_n|| \\
\epsilon_y &= |\theta_n - \bar{\theta}_n|.
\end{align*}
\]

where, \(\epsilon_n = \tilde{c}_n - x^*\), and \(\theta_n\) is an angle determined by \(\tilde{c}_n\) and \(c_{n+1}\). Note that \(\epsilon_x\) is on the logarithm scale, thus actual error between \(\epsilon_n\) and \(\tilde{\epsilon}_n\) becomes exponentially small as \(n\) increases. Actually \(-14\) in logarithm scale approaches the limitation of a double precision register.

The mismatch between \(\theta_n\) and \(\bar{\theta}_n\) can be evaluated good, that is, it is confirmed that the approximated analytic equation Eq. (6) describes the locations of UPPs. The error included in \(\tilde{c}_k\) comes seems to be modeling errors in Eq.
Figure 9: A directional coloring result for Eq. (1) with $a = -0.1$, $b = -1.7$. (a): $n = 34$, (b): $n = 35$.

(3), (4) and (5), however, the oscillation behavior in Fig 11 implies existence of one or more frequency components. We should try to add more sinusoidal term in Eq. (6).

5. Conclusions

In this work, a fractal nature of a chaos in a two-dimensional discrete equation is shown. The analytic formula locating UPPs is derived and estimation errors are discussed. We only pay attention to the center 1-periodic UPP, this method also applicable for any UPP. Applying this method to other maps is a future problem.

References


Stability of a Switched System for Continuous-time Tomographic Image Reconstruction

Omar M. Abou Al-Ola †, Ken’ichi Fujimoto‡, and Tetsuya Yoshinaga‡

†Graduate School of Health Sciences, and ‡Institute of Health Biosciences,
The University of Tokushima, Tokushima 770-8509, Japan
Email: [omar@x, fujimoto@yosinaga@] medsci.tokushima-u.ac.jp

Abstract—We have presented a novel approach for reconstructing tomographic images based on the idea of continuous dynamical methods. The method consists of a continuous-time image reconstruction (CIR) system described by differential equations for solving ill-posed inverse problems. We see that a switched system with a piecewise smooth vector field, which describes a block CIR system, can reconstruct better images than the smooth CIR system [15]. For investigating an essential property of the switched system, we considered a small system and gave the analytic solution of it, and to obtain a local minimum of the objective function, we proposed a continuous dynamical method as an initial

1. Introduction

Tomography is imaging by sections or sectioning, and computed tomography (CT) [1, 2, 3, 4] is a medical imaging method employing tomography created by computer processing. In general, a problem of reconstructed images from a projection operator and a projection data set become ill-posed. Many different reconstruction algorithms are used in medical practice to solve the inverse problem of image reconstruction and most algorithms fall into one of two categories, filtered back projection (FBP) which is a transform method and iterative methods using difference equations that we call them discrete-time image reconstruction (DIR); FBP demands fewer computational resources, while DIR generally produces fewer artifacts at a higher computing cost; moreover, there are continuous dynamical methods that can regularize such ill-posed inverse problems.

We have presented a novel approach [5] for reconstructing tomographic images based on the idea of continuous dynamical methods [6, 7, 8, 9, 10]. The method consists of a continuous-time image reconstruction (CIR) system [5, 11] described by differential equations for solving ill-posed [12, 13, 14] inverse problems. We see that a switched system with a piecewise smooth vector field, which describes a block CIR system, can reconstruct better images than the smooth CIR system [15]. For investigating an essential property of the switched system, we considered a small system and gave the analytic solution of it, and to obtain a local minimum of the objective function, we proposed a continuous dynamical method as an initial

2. Block CIR System [5, 11, 15]

The basic problem of computed tomography (CT) is to calculate the pixel values \( x \in \mathbb{R}^d_+ \) with \( \mathbb{R}_+ \) denoting the set of non-negative real numbers, satisfying

\[
y = Ax,
\]

where \( y \in \mathbb{R}^d_+ \setminus \{0\} \) is the projection value, and \( A \in \mathbb{R}^{d_+}_+ \setminus \{0\} \) is a normalized projection operator. For inconsistent projection data, Eq. (1) is an ill-posed problem, which means that its solution is not unique or does not exist.

In our previous papers, to find a solution \( x \), we formulated an optimization problem described as:

\[
\min_{x(t) \in \mathbb{R}^d_+} V(x(t)), \quad t \in \mathbb{R},
\]

\[
V(x) := \frac{1}{2} \| y - Ax \|_2^2.
\]

and to obtain a local minimum of the objective function, we proposed a continuous dynamical method as an initial
value problem in the following form:
\[ \frac{dx}{dt} = -X \frac{\partial V(x)}{\partial x} \]
_\[= XA^T(y - Ax), \quad (3)\]
\[t \in \mathbb{R}^+, \quad x(0) = x_0,\]
where \(X\) indicates the diagonal matrix of order \(J \times J\) in which the corresponding diagonal elements are elements of \(x\). The nonlinear system (3) has the property that \(V(x)\), which can be a Lyapunov function on the state space \(\mathbb{R}^m\), decreases along the solution \(\phi(t, x_0)\) in time through the initial state \(x_0 \in \mathbb{R}^m_+,\) with \(\mathbb{R}^m_+\) representing the set of positive real numbers.

We also proposed a block CIR system by introducing subsets of projections as in block-iterative DIR methods. Let \(B_m \in \mathbb{R}^{J \times J}\) and \(z_m \in \mathbb{R}^m_+\) be, respectively, a submatrix consisting of \(I_m\) partial rows of \(A\) and a subvector of \(y\) with the same corresponding rows of \(B_m\), for \(m = 1, 2, \ldots, M\), such that there exists an elementary matrix \(Q\) satisfying:
\[ Q \begin{pmatrix} B_1 \\ B_2 \\ \vdots \\ B_M \end{pmatrix} = A \quad \text{and} \quad Q \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_M \end{pmatrix} = y. \quad (4)\]

The block CIR system was defined by
\[ \frac{dx}{dt} = XB_m^T(z_m - B_mx), \quad (5)\]
\[t - k\tau \in [t_{m-1}, t_m), \quad t \in \mathbb{R}^+, \quad x(0) = x^0 \in \mathbb{R}^m_+,\]
for a series of times \(0 = t_0 < t_1 < t_2 < \ldots < t_M = \tau\) and a non-negative integer \(k\). This is a periodic non-autonomous system when \(M \geq 2\) and is an autonomous system described by Eq. (3) when \(M = 1\). The subsystem of Eq. (5) with \(M \geq 2\) defined in the time interval \(t - k\tau \in [t_{m-1}, t_m)\) for any \(m\) and \(k\) is described by:
\[ \frac{dx}{dt} = -X \frac{\partial V_m(x)}{\partial x} \]
\[\quad \text{where,} \quad V_m(x) = \frac{1}{2} ||z_m - B_mx||^2. \quad (6)\]

Each subsystem of Eq. (5) has the property that \(V_m(x)\), which can be a Lyapunov function, decreases in time along the solution starting from \(x(k\tau + t_{m-1}) \in \mathbb{R}^m_+\).

We theoretically, numerically demonstrated that our CIR, Eq. (3), and block CIR, Eq. (5), systems does not produce unphysically negative pixel values for positive initial values; and also, by using examples, we indicated that the quality of the reconstructed images by our systems was better than that from other methods.


Suppose that we are given a family \(f_p, p \in \mathbb{P}\) of functions from \(\mathbb{R}^m\) to \(\mathbb{R}^m\), where \(\mathbb{P}\) is some index set. This gives rise to a family of systems:
\[ \frac{dx}{dt} = f_p(x), \quad p \in \mathbb{P} \quad (8)\]
evolving on \(\mathbb{R}^m\). The functions \(f_p\) are assumed to be sufficiently regular. The easiest case to think about is when all these systems are linear and the index set \(\mathbb{P}\) is finite: \(\mathbb{P} = \{1, 2, \ldots, m\}\). The switched system with time-dependent switching, generated by the above family, can be described by the equation:
\[ \frac{dx}{dt} = f_{\sigma}(x), \quad (9)\]
where the switching signal \(\sigma : [0, \infty) \to \mathbb{P}\) is a piecewise constant function which has a finite number of discontinuities, which we call the switching times, on every bounded time interval and takes a constant value on every interval between two consecutive switching times. The role of \(\sigma\) is to specify, at each time instant \(t\), the index \(\sigma(t) \in \mathbb{P}\) of the active subsystem, i.e., the system from the family (8) that is currently being followed.

It is well known that a necessary condition for asymptotic stability under arbitrary switching is that all of the individual subsystems are asymptotically stable. In fact, this condition is not sufficient for asymptotic stability under arbitrary switching. The following theorem gives additional requirements, on the systems from the family (8), that guarantee asymptotic stability of the switched system (9) for arbitrary switching signals.

Definition 1. Consider a function \(V : \mathbb{R}^m \to \mathbb{R}\). It is called positive definite if \(V(0) = 0\) and \(V(x) > 0\) for all \(x \neq 0\). If \(V(x) \to \infty\) as \(|x| \to \infty\), \(V\) is said to be radially unbounded.

Definition 2. Given a positive definite continuously differentiable function \(W : \mathbb{R}^m \to \mathbb{R}\), we will say that it is a common Lyapunov function for the family of systems (8) if there exists a positive definite continuous function \(W : \mathbb{R}^m \to \mathbb{R}\) such that we have
\[ L_{f_p}V(x) = \frac{\partial V}{\partial x} f_p(x) \leq -W(x), \quad \forall x, \quad \forall p \in \mathbb{P}. \quad (10)\]

Theorem 1. If all systems in the family (8) share a radially unbounded common Lyapunov function, then the switched system (9) is globally uniformly asymptotically stable.

The terminology uniform is employed here to indicate the uniformity with respect to the switching signals, while the term global refers to the fact that it holds for all initial states.
4. Main Result

We studied a block CIR system (5) with \( I = J = M = 2 \), which is defined for \( t \in [k \tau, (k + \frac{1}{2}) \tau) \) by

\[
\frac{dx_1}{dt} = x_1(y_1 - x_1), \tag{11}
\]
\[
\frac{dx_2}{dt} = 0, \tag{12}
\]
and for \( t \in [(k + \frac{1}{2}) \tau, (k + 1) \tau) \) by

\[
\frac{dx_1}{dt} = x_1(y_2 - (x_1 + x_2)), \tag{13}
\]
\[
\frac{dx_2}{dt} = x_2(y_2 - (x_1 + x_2)), \tag{14}
\]

where \( k \) is a non-negative integer, and got its analytic solution which is positive when we start with a positive initial state [16].

Now, we can apply Theorem 1 by looking for a common Lyapunov function \( V(x) \) that would have satisfy Eq. (10). The two subsystems have the point \((y_1, y_2 - y_1)\) as a common equilibrium point which is stable for each of the two subsystems by virtue of existing two Lyapunov functions given by Eq. (7). Consider the following quadratic form:

\[
V(x) = \frac{1}{2} \begin{pmatrix} y_1 - x_1 \\ y_2 - x_1 - x_2 \end{pmatrix}^T \begin{pmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{pmatrix} \begin{pmatrix} y_1 - x_1 \\ y_2 - x_1 - x_2 \end{pmatrix}
= \frac{1}{2} p_{11}(y_1 - x_1)^2 + p_{12}(y_2 - x_1)(y_2 - x_1 - x_2) + \frac{1}{2} p_{22}(y_2 - x_1 - x_2)^2, \tag{15}
\]

for some positive definite matrix

\[ P = \begin{pmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{pmatrix} \]

whose elements are functions of \( x_1 \) and \( x_2 \). For the quadratic form to be positive definite, the elements of the matrix \( P \) must satisfy

\[ p_{11} > 0, \quad p_{11}p_{22} - p_{12}^2 > 0. \tag{16} \]

The derivative \( \dot{V}(x) \) is given by

\[
\dot{V}(x) = p_{12} \left[ -(y_1 - x_1)(y_1 + x_2) - (y_2 - x_1 - x_2)x_1 \right] \\
- p_{11}(y_1 - x_1)x_1 - p_{22}(y_2 - x_1 - x_2)(y_1 + x_2) \\
+ p_{12}(y_1 - x_1)(y_2 - x_1 - x_2) \\
+ \frac{1}{2} p_{11}(y_1 - x_1)^2 + \frac{1}{2} p_{22}(y_2 - x_1 - x_2)^2. \tag{17}
\]

Substituting from Eq. (11) and Eq. (12) into Eq. (17), we get

\[
L_{f_1} V(x) = -x_1(p_{11} + p_{12})(y_1 - x_1)^2 - x_1(p_{12} + p_{22})(y_1 - x_1)(y_2 - x_1 - x_2) \\
+ \frac{1}{2} p_{11}(y_1 - x_1)^2 + p_{12}(y_1 - x_1)(y_2 - x_1 - x_2) \\
+ \frac{1}{2} p_{22}(y_2 - x_1 - x_2)^2. \tag{18}
\]

Substituting from Eq. (13) and Eq. (14) into Eq. (17), we get

\[
L_{f_2} V(x) = -x_1(p_{11} + p_{12})(y_1 - x_1)^2 - x_1(p_{12} + p_{22})(y_1 - x_1)(y_2 - x_1 - x_2) \\
- (x_1 p_{11} + x_1 p_{12} + x_2 p_{22})(y_1 - x_1)(y_2 - x_1 - x_2) \\
+ \frac{1}{2} p_{11}(y_1 - x_1)^2 + p_{12}(y_1 - x_1)(y_2 - x_1 - x_2) \\
+ \frac{1}{2} p_{22}(y_2 - x_1 - x_2)^2. \tag{19}
\]

Now we want to choose \( p_{11}, p_{12}, \) and \( p_{22} \) such that \( L_{f_1} V(x) \) and \( L_{f_2} V(x) \) are negative definite. Since the cross product \((y_1 - x_1)(y_2 - x_1 - x_2)\) is sign indefinite, we will cancel it by taking

\[ p_{22} = -p_{12}, \quad p_{11} = -\left( \frac{x_1 + x_2}{x_1} \right) p_{12}. \tag{20} \]

With these choices, \( p_{12} \) must be negative for \( V(x) \) to be positive definite via the satisfaction of Eq. (16). To simplify our choices, let us take \( p_{12} \) as a constant and so, by using Eq. (20), we get

\[ \dot{p}_{22} = \dot{p}_{12} = 0, \quad \dot{p}_{11} = -\left( \frac{x_2}{x_1} \right) \dot{p}_{12}. \tag{21} \]

which implies that

\[ \dot{p}_{11} = \frac{x_2}{x_1} (y_1 - x_1) p_{12}, \quad \dot{p}_{11} = 0, \tag{22} \]

for the first and second subsystems, respectively. Using the above discussion to get \( L_{f_1} V(x) \) and \( L_{f_2} V(x) \) as

\[
L_{f_1} V(x) = \frac{1}{2} x_2 \left( 1 + \frac{y_1}{x_1} \right) (y_1 - x_1)^2 p_{12}, \quad p_{12} < 0, \tag{23} \]

and

\[
L_{f_2} V(x) = x_2(y_2 - x_1 - x_2)^2 p_{12}, \quad p_{12} < 0. \tag{24} \]

Hence, the common Lyapunov function takes the form

\[
V(x) = -\frac{1}{2} \left( \frac{x_1 + x_2}{x_1} \right) (y_1 - x_1)^2 p_{12} - \frac{1}{2} (y_2 - x_1 - x_2)^2 p_{12} \\
+ (y_1 - x_1)(y_2 - x_1 - x_2) p_{12} p_{12} < 0. \tag{25} \]

If we take \( p_{12} = -1 \), for example, we will get the common Lyapunov function

\[
V(x) = \frac{1}{2} \left( \frac{x_1 + x_2}{x_1} \right) (y_1 - x_1)^2 + \frac{1}{2} (y_2 - x_1 - x_2)^2 \\
- (y_1 - x_1)(y_2 - x_1 - x_2). \tag{26} \]

Now we introduce two figures, the first one, Figure 1, shows the value of the common Lyapunov function given by Eq. (26), while the second one, Figure 2, shows a trajectory approaches the common equilibrium point \((y_1, y_2 - y_1)\).
5. Conclusion

In this study, for investigating an essential property of our new approach, Block CIR system, for reconstructing tomographic images, we considered a small switched system and discussed its stability via the common Lyapunov function approach which shows stability for any switching signal.

References

Neuronal Avalanches Induced by Spike-Timing-Dependent Plasticity

Shuhei Ohno†, Hideyuki Kato† and Tohru Ikeguchi‡,§

†Graduate School of Science and Engineering, Saitama University
‡Brain Science Institute, Saitama University
255 Shimo-okubo, Sakura-ku, Saitama-shi, Saitama, 338-8570, Japan
Email: †[oono.kato]@nls.ics.saitama-u.ac.jp, ‡tohru@ics.saitama-u.ac.jp

Abstract—Recent studies in the neuroscience have reported that neuronal avalanches are observed in cortical areas of the brain. The neuronal avalanches are considered as one of the mechanisms of memory functions in the brain. However, it still remains elusive what is a key mechanism to produce neuronal avalanches. To solve this question, we introduce spike-timing-dependent plasticity (STDP) as a candidate for the mechanism to induce the neuronal avalanches because STDP constructs functional cortical circuits. In this paper, based on this idea, we analyzed neuronal activities in networks constructed through STDP from viewpoints of neuronal avalanche.

1. Introduction

In neural systems, billions of neurons construct neural networks with synapses. It is generally believed that activities of the neural networks realize high level brain functions, for example, memories, learning, development and so on. Synfire chain is characteristic activity of the neural networks in which synchronous neuronal spikes propagate spatiotemporally [1]. Recent physiological studies have reported that synfire chains are observed in cultured slices of neocortex of rat. Distributions of event sizes and durations in the synfire chains obey a power-law with exponents −3/2 and −2, respectively [2, 3]. These power-law exponents are also observed in avalanches of snow mountains. Then the synfire chains with these power-law exponents are called neuronal avalanches. These power-law exponents are often reported in nonlinear dynamical systems in a critical state [4]. The previous studies suggested that the neuronal avalanches may be caused by a critical process which can optimize information processing [2]. In addition, these synchronous activity patterns in neuronal avalanches are not only highly diverse but also repeatable [3]. According to these properties, it is suggested that neuronal avalanches play important roles for the brain functions of memory.

Several neural network models have been proposed to reproduce neuronal avalanche [5]. However, it is unknown what is a key structure to produce the neuronal avalanches, and how such structures are induced. It is important to clarify such an issue to understand brain function of memory.

Synaptic connections in neural networks are modified depending on a relative spike timing between pre- and postsynaptic action potentials [6]. These synaptic modifications are called spike-timing-dependent synaptic plasticity (STDP). Neural networks with STDP have been widely analyzed in the computational neuroscience. The long-term potentiation (LTP) occurs when a postsynaptic action potential arises after a presynaptic action potential, whereas the long-term depression (LTD) occurs in a case of the reverse order of action potentials.

It is natural to expect that STDP might be one of the mechanisms of constructing characteristic structures of the neural networks and these structures can produce neuronal avalanches. Based on this idea, in this paper, we analyzed activities of neural networks with STDP from viewpoints of neuronal avalanche.

2. Methods

2.1. Neural networks with STDP transformation

In our numerical experiments, we used a neuron model whose dynamics is represented by 2-dimensional ordinary differential equations. Dynamics of the i-th neuron are defined as follows:

\[ \dot{v}_i = 0.04v_i^2 + 5v_i + 140 - u_i + I_i^{syn}(t) + I_i^{bg}(t) + I_i^{ext}(t), \]  (1)  
\[ \dot{u}_i = a(bv_i - u_i), \]  (2)

where \( v_i \) and \( u_i \) are the membrane potential and the recovery variable of the i-th neuron, respectively [7]. The variables \( I_i^{syn}(t), I_i^{bg}(t) \) and \( I_i^{ext}(t) \) represent sum of synaptic inputs, background input and external input to the i-th neuron at time \( t \), respectively. The sum of synaptic inputs \( I_i^{syn}(t) \) are modeled as: \( \sum_j g_{ji} \delta(t - t_j^s) \) where \( N \) represents the number of neurons in the network, \( g_{ji} \) represents a synaptic weight from the j-th to the i-th neuron, \( t_j^s \) represents the j-th spike time of the j-th neuron, and \( \delta(t) \) represents the Dirac delta function. If the variable \( v_i \) reaches 30 [mV], the i-th neuron fires and the variables \( v_i \) and \( u_i \) are reset to \( c \) and \( d \). For all neurons, we set the parameters \( (b,c) = (0.2, -65) \). For excitatory neurons, we set \( (a,d) = (0.02, 8) \), while \( (a,d) = (0.1, 2) \) for inhibitory neurons. In the experiments, the number of neurons in the network \( N = 10,000 \) in which \( 4N/5 = 8,000 \) neurons are excitatory and \( N/5 = 2,000 \) neurons are inhibitory. Each neuron has average \( M = 1,000 \) random connections to postsynaptic neurons. There are no connections...
Figure 1: An example of network activities after the STDP learning. In this figure, blue regions represent neuronal avalanches and white regions correspond to blank times. In our analysis, a size of a neuronal avalanche is defined by the number of spikes included in one blue region. A lifetime of a neuronal avalanche corresponds to duration of the blue region.

between inhibitory neurons. All pacemaker neurons (see below) are excitatory. In our neural network, $N_{pm} (= 100)$ pacemaker neurons are included. Each pacemaker neuron has $M_{pm} (= 65)$ postsynaptic neurons which are selected randomly from excitatory neurons except for pacemaker neurons. In the neural network, synaptic weights from the $j$th to the $i$th neurons are modified depending on their activities. The synaptic modification by STDP is described by the following equation [8]:

$$\Delta g_{ij} = \begin{cases} A^+ \exp(-\frac{t_i - t_j}{\tau}) & (t_j < t_i), \\ -A^- \exp(-\frac{t_i - t_j}{\tau}) & (t_j \geq t_i), \end{cases}$$

where $A^+ (= 0.09)$ and $A^- (= 0.1)$ are the learning rates of the LTP and LTD, $\tau (= 10$ [ms]) is a time constant that determines the exponential decays of the LTP and LTD in Eq. (3), and $t_i$ and $t_j$ are the firing time of the $i$th and $j$th neurons, respectively [8]. In addition, nearest-neighbor spikes contribute for the long-term synaptic modifications [9]. The STDP learning is applied to only excitatory synapses from a physiological point of view [10]. The excitatory synapses are additively modified through STDP, so that we limit a range of synaptic weights with hard bounds. The range of synaptic weights is set as $g_{\text{min}} \leq g_{ij} \leq g_{\text{max}}$ where $g_{\text{min}}$ and $g_{\text{max}}$ are 0 and 10, respectively.

In our simulation, we introduced pacemaker neurons in the neural network. Although the pacemaker neurons are generally defined by their intrinsic property to generate rhythmic bursting activity and found in several brain regions, in our network model, the pacemaker neurons fire at a constant frequency and are not affected by inputs from the other neurons. For the pacemaker neurons we always set $I_{i}^{\text{PM}}(t) = 0$ and $I_{i}^{\text{AI}}(t) = 5$ in the neuron model we have already defined in Eq. (1).

2.2. Activities of neural network after STDP transformation

After the STDP learning for 100 [sec], we simulated the neural network without pacemaker neurons and STDP. To drive the neural network, we assumed independent Poisson-process spike trains as background inputs. We set the amplitude $I_{i}^{\text{BG}}(t) = 3.1$ [mV] which corresponds to sub-threshold background input. In addition, we selected an excitatory neuron from the network randomly every 200 [ms] and apply an external input to the neuron. We set the amplitude $I_{i}^{\text{EXT}}(t) = 20$ [mV] as a supra-threshold input for evoking neuronal activities.

We analyze activities of the neural network from viewpoints of neuronal avalanche.

2.3. Measures

Figure 1 shows an example of a network activity after the STDP learning. To characterize the network activities, we use two measures, sizes and lifetimes of neuronal avalanches. In this study, we define duration not less than 4 [ms] in which no spikes are emitted as a blank time (white regions in Fig. 1). The other regions (blue regions in Fig. 1) are regarded as neuronal avalanches. Then, the size of each neuronal avalanche is the number of spikes in the blue region and the lifetime is the duration of the blue region.

3. Results

3.1. Probability distributions

The probability distributions of neuronal avalanche size and lifetime are shown in Fig. 2 when the frequency of the background inputs is varied from 200 to 400 [Hz]. From Fig. 2, the probability distributions of the neuronal avalanche sizes exhibit linearity in a log-log scale. The linearity indicates that the probability distribution obeys a power law $P(s) \sim s^{-\alpha}$, where $s$ is the size of neuronal avalanche, $P(s)$ is the probability of size $s$ of the neuronal avalanche and $\alpha$ is a power-law exponent. In cortical networks, it is often observed that the power-law exponent is $-3/2$ [2]. Blue lines express the slope of $-3/2$ in the size distribution while green lines express the slope of $\alpha$ fitted by the least square approximation from data.

When we applied background inputs to the network after the STDP learning at 200 [Hz] (Fig. 2 (a) upper), the
probability distribution of the neuronal avalanche size obey the power law distribution and its power law exponent of $\alpha = -2.086$ is smaller than $-3/2$, which means that large size neuronal avalanches are rarely observed. On the other hand, when we applied 400 [Hz] background inputs (Fig. 2 (c) upper), the exponent of $\alpha = -0.884$ is larger than $-3/2$, which means that large size neuronal avalanches appear more frequently than the case of 200 [Hz] background inputs. A cut-off point is around 1,000. When we applied 300 [Hz] background inputs (Fig. 2 (b) upper), the exponent of $\alpha = -1.385$ is close to $-3/2$ up to the cut-off point around 1,000, which means that the neuronal activities in this case shows neuronal avalanches in cortical networks.

In cortical networks, the probability distributions of the neuronal avalanche lifetime also show linearity in a log-log scale which means the probability distributions obey a power law $P(T) \sim T^\beta$, where $T$ is the lifetime of a neuronal avalanche, $P(T)$ is its probability and $\beta$ is a power-law exponent. It is often observed that the power-law exponent is $-2$ [2]. Blue lines in Fig. 2 indicate slope of $-2$ while green lines are fitted results from our simulation. When we applied 200 [Hz] background inputs (Fig. 2 (a) lower), the probability distribution of neuronal avalanche lifetimes shows a power-law distribution. A slope of $\beta = -2.113$ is close to $-2$ and an exponential cut-off point is observed around 30 [ms]. These results are almost the same as those observed in the cortical networks [2]. When we applied 300 [Hz] background inputs (Fig. 2 (b) lower), the distribution of neuronal avalanche lifetimes shows $\beta = -0.606$. This is larger than $-2$ and a cut-off point is observed around 30 [ms]. On the other hand, when we applied 400 [Hz] background inputs (Fig. 2 (c) lower), neuronal avalanche lifetimes do not obey power-law distributions. When the lifetimes are from about 10 to 30 [ms], the slope of the distribution is positive, which means that the longer avalanches are observed more frequently than the shorter ones in this lifetime band. From these results, the probability distribution of neuronal avalanche lifetimes is sensitive to frequencies of the background inputs.

3.2. Relationship between synaptic connections in an initial network and activities after the STDP learning

It is clear that spike propagations are always induced if neurons directly connected with pacemaker neurons begin to fire (blue dots in Fig. 3). In contrast, spike propagations do not occur when the other neurons fire (red dots in Fig. 3). In addition, neuronal spike propagations occur only among neurons which are directly connected with pacemaker neurons (Fig. 3) because they are driven by the pacemaker neurons during the STDP learning and the synapses between them are strengthened. It is considered that after the STDP learning, the synapses between them are strong enough to cause spike propagations but the others are not (Fig. 4).
4. Conclusion

In this paper, we analyzed neuronal activities after the STDP learning from viewpoints of neuronal avalanche. As a result, the probability distribution of neuronal avalanche sizes and lifetimes show the power-law whose exponent is $-3/2$ and $-2$, respectively. In addition, we clarified that the exponents vary depending on the frequency of external inputs to the network. If frequencies of external inputs are higher, larger sizes and longer lifetimes of neuronal avalanches are more frequently appear. From these results, it is suggested that exponents of distributions of the size and the lifetime strongly depend on the frequency of background inputs. Furthermore, we show how the neuronal spike propagations occur. In addition, the neuronal spike propagations occur only among neurons which are directly connected with pacemaker neurons in the network. During the STDP learning, the neurons are driven by the pacemaker neurons and the synaptic weights between them are strengthened enough to induce the neuronal spike propagations. However, neuronal avalanches have another property; even though observation times of two neuronal avalanches are different in a few hours, their spatiotemporal patterns are highly correlated [3]. As a future work, we will investigate the correlation between spatiotemporal patterns of neuronal activities in the network after the STDP learning.

Acknowledgments

The research of T.I. is partially supported by Grant-in-Aid for Scientific Research (C) (No.20560352) from JSPS.

References


Estimation of Neural Network Structure by Transforming Spike Sequences to Continuous Time Series

Kaori Kuroda † and Tohru Ikeguchi‡,†

†Graduate School of Science and Engineering, Saitama University
‡Brain Science Institute, Saitama University
225 Shimo-Ohkubo, Sakura-ku, Saitama-city, Saitama 338-8570, Japan
E-mail: kuroda@nls.ics.saitama-u.ac.jp, tohru@mail.saitama-u.ac.jp

Abstract—In neural systems, many complicated behaviors are observed. To understand network dynamics, and reproduce the complicated behavior, it is important to clarify the network structures as well as their dynamics. To resolve this issue, we have already proposed a measure, partial spike time metric. Although this measure exhibits high performance to estimate the network structures, it cannot evaluate negative correlations correctly. In this paper, to resolve this problem, we transform multi-spike sequences into continuous time series to estimate the neural network structures. As a result, our proposed method is more effective for estimating neural network structures than the conventional method.

1. Introduction

Neural systems often produce complicated behaviours due to the interactions among the neurons in them. Usually, such complicated behaviours depend on how the neurons in the network are connected, that is, they depend on the network topology. Thus, to analyze, model or predict such behaviours, it is essential to understand the network structure as well as their dynamics. However, it is not so easy to analyze the anatomical structure of the neural network. On the other hand, recent developments in measurement technologies make it possible to observe multi-spike sequences. It is intuitive to expect that these observed spike sequences reflect essential information about the neural network structure. From this point of view, we have already proposed a measure based on spike time metric [1] and partialization analysis [2, 3]. Although this measure, the partial spike time metric coefficient (PSTMC), exhibits high performance to estimate the network structure [4, 5], it has a drawback: if two spike sequences have anti-phase relation, the PSTMC cannot detect the relation correctly. In this paper, to resolve this issue, we introduce a new strategy: we transform spike sequences into continuous time series to detect their relation. To transform spike sequences to continuous time series, we applied two methods. The first method is an interpolation of inter-spike intervals by sinusoidal waves. The second is a kernel density estimator which uses the Gaussian function as a kernel function. We applied the analytical results for calculating an optimal width of the kernel function [6]. Then, we applied the partial correlation analysis to the transformed continuous time series. The results show that our method exhibits higher performance than the conventional method [4, 5].

2. Transformation method

We first transform spike sequences into continuous time series. To transform the spike sequences, we applied two transformation methods.

2.1. Method 1: Interpolation of inter-spike intervals by sinusoidal waves

Let us describe the \( j \)th spike in the \( i \)th spike sequence as \( t'_i \). Then, the \( i \)th spike sequence is described as \( s_i = \{ t'_i, t'_2, \ldots, t'_{l_i} \} \) where \( l_i \) is the last index of \( s_i \). Firstly, we interpolate the \( j \)th segment (inter-spike interval, ISI) bounded by two adjacent spikes, \( t'_j \) and \( t'_{j+1} \), by the following equation:

\[
x'(t) = \frac{1}{2} + \frac{1}{2} \cos \frac{2\pi}{T_j} (t - t'_j), \quad (t'_j \leq t \leq t'_{j+1}),
\]

where \( T_j = t'_{j+1} - t'_j \). Then, \( x'(t) (j = 1, 2, \ldots, l_i-1) \) is concatenated to produce a transformed continuous time series from the \( i \)th spike sequence. In Fig. 1, a spike sequence and corresponding transformed continuous time series are shown.

Figure 1: Example of transformation of a spike sequence into a continuous time series by the method 1. Blue lines indicate spike timings and the red curve indicates transformed continuous time series.

![Figure 1](image-url)
2.2. Method 2: The Kernel density estimator

In method 2, we transformed continuous time series from spike sequences by using the kernel density estimator. We used the Gaussian function as the kernel function. The kernel density function is described by the following equation:

\[ f_K(t) = \frac{1}{l_w} \sum_{j=1}^{l} K \left( \frac{t - t_j}{w} \right) \]  

where \( w \) is the bandwidth and \( K(\cdot) \) is the kernel function. The Gaussian function with mean zero and variance unity as a kernel function is defined as:

\[ K(t) = \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}}. \]

In Fig. 2, an observed spike sequence and corresponding transformed continuous time series are shown. Here, the bandwidth \( w \) used for the kernel estimation is optimized by the method for selecting a fixed kernel bandwidth [6].

![Figure 2: Example of transformation of a spike sequence into a continuous time series by the method 2. Blue lines indicate spike timings and the red curve indicates transformed continuous time series.](image)

3. Partial correlation coefficient

We used the partial correlation coefficient [7, 8] to the transformed continuous time series to detect the relation between two neurons. The partial correlation coefficient is defined as:

\[ r_{ij} = -\frac{\sigma_{ij}}{\sqrt{\sigma_{ii}\sigma_{jj}}} \]

where \( \sigma_{ij} \) is the \((i, j)\)th entry of the inverse of the correlation matrix of the transformed time series from the \(i\)th and \(j\)th spike sequences. The partial correlation coefficient can quantify the similarity between the two time series with removing spurious correlations between them, for example, in a case that two neurons are driven by a common input from another neuron [4, 5].

4. Simulation

To evaluate the validity of our method, we used a neural network constructed from a mathematical model of the Izhikevich simple neuron model [10] and generated multi-spike sequences. The dynamics of the \(i\)th neuron in the neural network is described by the following equations:

\[ \dot{v}_i = 0.04v_i^2 + 5v_i + 140 - u_i + I_i, \]

\[ \dot{u}_i = a(bv_i - u_i), \]

if \( v_i \geq 30 \text{ [mV]} \), then \( v_i \leftarrow c_i, u_i \leftarrow u_i + d_i \)

where \( v_i \) is the membrane potential, \( u_i \) is the membrane recovery variable; and \( a, b, c_i, \) and \( d_i \) are dimensionless parameters. We set \( a = 0.02, b = 0.2, c_i = -65 + 15 \times U^2, d_i = 8 - 6 \times U^2 \) where \( U \) is uniform random numbers between \([0, 1]\). The variable \( I_i \) is the sum of the external and synaptic inputs from coupled neurons. The synaptic weight is set to six and the amplitude of the external inputs to five times \( G \), where \( G \) is a Gaussian random number with a mean value and standard deviation of zero and unity, respectively. The neural network is composed of only excitatory neurons. For the sake of simplicity, we do not consider conduction delays.

We generated a complex network structure having a regular ring topology with 100 neurons by a random rewiring of the synaptic connections between neurons in the same manner as that described in Ref. [12]. We set the parameters \( k \) (the number of edges in the regular network) to four and \( p \) (rewiring probability) to 0.1.

We conducted numerical experiments according to the following procedures. First, to generate multi-spike sequences, we constructed a neural network using the Izhikevich simple neuron model and applied external inputs to the neural network. Second, we transformed the spike sequences into continuous time series. Third, we calculated the partial correlation coefficient of the transformed continuous time series. If the \(i\)th and the \(j\)th neurons are coupled, the partial correlation coefficient becomes large. On the other hand, if these neurons are not coupled, it becomes small. Finally, we classified coupled pairs and uncoupled pairs by the Otsu thresholding [11] which is based on a linear discriminant analysis.

To evaluate an overall estimation accuracy, we compared the estimated network structure with the true network structure. For this evaluation, we used the following index:

\[ E = \frac{\sum_{i<j} \left( \alpha_{ij} \tilde{\alpha}_{ij} + (1 - \alpha_{ij})(1 - \tilde{\alpha}_{ij}) \right)}{N(N-1)} \]

where \( \alpha_{ij} \) and \( \tilde{\alpha}_{ij} \) are the directional connectivity of the true network structure and the estimated network structure, respectively. If the \(i\)th and \(j\)th elements are (estimated to be) coupled, \( \alpha_{ij} \) and \( \tilde{\alpha}_{ij} \) take a value of unity. If they are not, \( \alpha_{ij} \) and \( \tilde{\alpha}_{ij} \) take a value of zero. If \( E \) approaches unity, the estimation accuracy increases.

5. Results and discussions

To compare the proposed methods which transform spike sequences to continuous time series and the con-
Figure 3: Histograms of (a) PSTMC in the conventional method [4, 5], partial correlation coefficients by (b) the method 1 and (c) the method 2. The number of neurons is 100. The temporal epoch of spike sequences used for estimation is 50 [s]. Histograms of all of the PSTMC and the partial correlation coefficients are indicated in red, and those of the coupled elements are superimposed by blue. The black vertical lines show thresholds decided by the Otsu thresholding. If the PSTMC and the partial correlation coefficients are less than the threshold, corresponding neurons are classified into the uncoupled class.

![Histograms](image)

Figure 4: Estimation accuracy $E$ of the network structure for rewiring probabilities $p$. The number of neurons, the temporal epoch and the coupling strength are 100, 50 [s] and 6. The red line indicates the conventional method, the green line indicates the method 1 and the blue line indicates the method 2. Error bars with 20 trials are also provided.

![Estimation accuracy](image)

Figure 5: Estimation accuracy $E$ of the network structure for several temporal epochs. The number of neurons and the coupling strength are 100 and 6. Red line indicates the conventional method, green line indicates the method 1 and blue line indicates the method 2. Error bars with 20 trials are also provided. The inset shows an enlargement.

![Estimation accuracy](image)

The conventional method which uses PSTMC, we also apply the PSTMC [4, 5] to the same network structure.

Figure 3 shows histograms of PSTMC and partial correlation coefficients. Although the conventional method and the method 1 can separate coupled and uncoupled pairs of neurons, a few uncoupled pairs are estimated as coupled pairs. However, coupled and uncoupled pairs of neurons are more clearly distinguished by the method 2.

In Fig. 4, we show the results when the rewiring probability is changed. In the method 1, the estimation accuracy for the random network ($p = 1.0$) is low. However, the conventional method and the method 2 show high estimation accuracy for all rewiring probabilities.

We examined how the estimation accuracy depends on the temporal epoch for observing spikes (Fig. 5). The conventional method exhibits higher accuracy than other methods when the temporal epoch of the spike sequences is shorter than 20 [s]. However, in method 2, the estimation accuracy is higher than the other methods when the temporal epoch of the spike sequences is longer than 20 [s].

In Fig. 6, we show the results when the coupling strength is changed. If the coupling strength becomes larger than four, the estimation accuracy of the conventional method is high. If the coupling strength becomes larger than five, the estimation accuracy becomes higher in the method 2. If the coupling strength becomes larger than six, the estimation accuracy of the method 2 is the highest.

In addition, we show the results for different network sizes (Fig. 7). In the method 1, the estimation accuracy worsens as the network size increases. However, the estimation accuracy in the method 2 and the conventional method are still high even if the network size is large.

From these results, the estimation accuracy in the method 2 is better than the other methods. In this neural network model, we used different dynamics of neurons; regular spiking, intrinsically bursting, and chattering neurons. If two neurons of different dynamics are coupled, the correlation between the two transformed continuous time series decreases in the conventional method [4, 5] and the
However, in the method 2, true relation can be identified if observed spike sequences of two neurons are transformed with an optimal bandwidth, even though the dynamics of the two neurons differs. The proposed methods are similar to the method of Sameshima and Baccala [14] in the point of transforming spike sequences to continuous time series and using partialization analysis. However, in the method of Sameshima and Baccala [14], parameters are not optimized when spike sequences are transformed. Moreover, the proposed methods can estimate whole network structure, while the method of Sameshima and Baccala estimate interaction between two neurons.

![Figure 6: Estimation accuracy $E$ of the network structure for several coupling strengths. The number of neurons and the temporal epoch are 100 and 50 [s]. Red line indicates the conventional method, green line indicates the method 1 and blue line indicates the method 2. Error bars with 20 trials are also provided.](image)

![Figure 7: Estimation accuracy $E$ of the network structure for several network sizes. The temporal epoch and the coupling strength are 50 [s] and 6. The red line indicates the conventional method, the green line indicates the method 1 and the blue line indicates the method 2. Error bars with 20 trials are also provided.](image)

### 6. Conclusions

In this paper, we proposed new methods for estimating network structures only from the information of observed multi-spike sequences. We transformed spike sequences to continuous time series by two methods and applied the partial correlation analysis to them. As a result, method 2 (which used the kernel density estimator) exhibits higher performance than the conventional method [4, 5] and method 1.

In this paper, although we only consider the case in which neurons are compiled without delay, real spike sequences can be produced from a network in which the neurons interact through coupling delays. Thus, in future works, it is important to treat delays by modifying the estimation method of optimized bandwidth. In addition, we apply the partialization analysis to a different measure, for example, mutual information [15].

### Acknowledgments

This research of K.K. is partially supported by NEC C&C Foundation. The research of T.I. is partially supported by a Grant-in-Aid for Exploratory Research (No. 20650032) from JSPS.

### References

Improvement of Accuracy and Processing Speed of a Maximum Neural Network Algorithm for the Channel Assignment Problem

Kazunori Nemoto† and Junji Kitamichi‡

†School of Computer Science and Engineering, The University of Aizu
‡Graduate School of Computer Science and Engineering, The University of Aizu
Aizuwakamatsu, Fukushima 965-8580, JAPAN
Email: kitamichi@u-aizu.ac.jp

Abstract—In recent years, the popularity of cellular mobile communication systems increases steady. However, the usable frequency spectrum or channels for the Cellular Radio Networks (CRNs) are limited. Thus, algorithms for an efficient utilization of channels: the Channel Assignment Problem (CAP) have become important. In this paper, we propose an improved Neural Network Algorithm (NNA) with parallelism to improve solution accuracy and speed-up processing for the static CAP in CRNs. Our proposed algorithm achieves the improved solution accuracy because of large hill-climbing. In addition, parallel processing using multi-thread can achieve faster processing. We verify performance through simulations using benchmark problems and our proposed algorithm can search better solutions and obtains faster processing speeds than the existing one.

1. Introduction

Recently, the popularity of cellular mobile communication systems increases steady. However, the usable frequency spectrum or channels for the Cellular Radio Networks (CRNs) are limited. Thus, algorithms for an efficient utilization of channels: the Channel Assignment Problem (CAP) have become increasingly important[1]-[4],[6]. CRNs consist of a number of fixed Base Transceiver Stations (BTSs) and a larger number of connection requests from cellular phone users. These users can receive cellular phone service due to the channel assignments allocated by a BTS. The CAP entails allocating channels to cells in the CRN such that the effective assignment of required channel numbers minimizes mutual interference while satisfying electromagnetic compatibility constraints as possible. We consider the following three constraints [6]. (1) For a certain pair of radio cells, the same channel cannot be used simultaneously. (2) Any pair of channels assigned to a cell should have a certain distance between them. (3) The adjacent channels in the frequency domain cannot be assigned to adjacent cells simultaneously.

The CAP can be reduced from the graph-coloring problem which is NP-complete [7], and its computation time grows exponentially for a large-scale network. Thus, an exact search for the optimal solution is impracticable. For this reason, many researchers have investigated approximate algorithms for the CAP. Algorithms using Neural Network Algorithms (NNAs) have been proposed[1]-[4],[6].

In this paper, an expanded Maximum Neural Network (MNN) algorithm, proposed by Ikenaga and colleagues[2], is verified, and for improvement of the solution accuracy, large hill-climbing is suggested. The addition of large hill-climbing causes slower processing speeds than the original one. Thus, speed-up methods using multi-thread are proposed. We verify performance of proposed algorithm through simulations using some benchmarks and our proposed algorithm can search better solutions and obtains faster processing speeds than the existing one.

2. Channel Assignment Problem

The Channel Assignment Problem (CAP) consists of a number of users and Base Transceiver Stations (BTSs). In this paper, a BTS is assumed to have a hexagonal-shaped management domain (cell) as shown in Figure 1. If a user associated with a cell requests a connection, the BTS assigns a channel to accommodate it. Then, the effective channel assignment is required for each cell on the cellular radio network as the CAP.

![Figure 1: Cellular radio network](image)

We assume N cells and M channels in the CRNs. For a demand vector D (the channel requirements for cells) and a $N \times M$ compatibility matrix $C$, a channel assignment minimizing total interference is required. The interference between cells is represented by interference matrix $E$, a three dimensional $N \times N \times M$ matrix. An element $e_{ijk}$ in $E$ indicates interference of any channels as having a distance of $k$ from channel $j$ assigned to cell $i$. The demand vector $D$ has $N$ elements and an element $d_i$ indicates the number of required channels form cell $i$.

In this paper, the interference matrix $E$ assumes that the compatibility matrix $C$ gives $E$ as defined by Smith[1].

$$e_{ijk} = c_{ij}, \quad e_{ijk} = \begin{cases} \ 0 : \ (if \ c_{ij} \leq k) \\ c_{ij} - k : \ (if \ c_{ij} > k) \end{cases}$$

(1)

where $c_{ij}$ ($i=1, \cdots, N$, $j=1, \cdots, N$) is an element of $C$ and is given by Gamst’s compatibility matrix of the CAP.
Figure 2 shows an example of the CAP with four cells. Figure 2(a) shows compatibility matrix \( C \) and demand vector \( D \). Figure 2(b) illustrates interference matrix \( E \). Figure 2(c) shows the corresponding network topology of matrix \( C \). The solution for the minimum number of channels needed for no interference assignment in this example is 11 because 11 channels would be needed for cell 4 as can be seen in Figure 2. Figure 2(d) shows an optimal solution in this example, where the total interference is 1.

\[
C = \begin{pmatrix}
5 & 4 & 0 & 0 \\
4 & 0 & 0 & 1 \\
4 & 5 & 0 & 1 \\
0 & 1 & 2 & 5
\end{pmatrix}, \quad D = \begin{pmatrix}
1 & 1 \\
1 & 1 \\
3 &
\end{pmatrix}
\]

(a) Compatibility matrix and demand vector

\[
e_{ij1} = \begin{pmatrix}
4 & 3 & 0 & 0 \\
3 & 4 & 0 & 0 \\
0 & 0 & 4 & 1 \\
0 & 0 & 1 & 4
\end{pmatrix}, \quad e_{ij2} = \begin{pmatrix}
3 & 2 & 0 & 0 \\
2 & 3 & 0 & 0 \\
0 & 0 & 3 & 0 \\
0 & 0 & 0 & 3
\end{pmatrix}
\]

(b) Interference matrix

(c) The corresponding network topology

(d) Example of a channel assignment result

Figure 2: Example of CAP

### 3. Neural network algorithm and Ikenaga’s algorithm

A Neural Network Algorithm (NNA) is a heuristic algorithm and a mathematical model based on a neural network. A neural network model is decided by ways of aggregates of neurons. A neuron has multi inputs and one output which collects other neurons outputs as input then calculates own state and decides the output. In this paper, the Hopfield neural network [5] as the neural network and an extended maximum neuron as a neuron model [2] are employed. To solve problems using NNA, problems need to be represented by neurons, an energy function that is a non-negative function and intends states of whole neurons need to be set the minimum if state is the optimal or sub-optimal solution, and the minimum solution is searched by a motion equation using the steepest descent method [5].

We describe Ikenaga’s NNA algorithm. A two dimensional \( N \times M \) array is used as neuron representation and a neuron \( ij \), an assignment for cell \( i \) and channel \( j \), has integral \( U_{ij} \) as input and \( V_{ij} \) whose value is 0 or 1 as output. The \( V_{ij} = 1 \) means that channel \( j \) is assigned to cell \( i \) and the \( V_{ij} = 0 \) means that there is no assignment.

\( M \) neurons of each cell is sorted in descending order, and the outputs of top to \( D_{i} \)-th neurons are 1 and others are 0.

\[
V_{ij} = \begin{cases} 
1 & \text{if } U_{ij} \geq U_{i-th} \\
0 & \text{otherwise}
\end{cases}
\]

where \( U_{i-th} \) is the \( d_{i} \)-th largest value in \( U_{1}, \cdots, U_{i,M} \). If more than \( d_{i} \) neurons satisfy \( U_{ij} \geq U_{i-th} \), the neurons which output 0 previously are selected preferentially because the search space is expanded by the changes in the assignment result and more too, neurons which output 1 are determined randomly from acceptable neurons. By employing the expanded maximum neuron, the energy function has only one term which represents the total interference.

\[
E = A \sum_{i=1}^{N} \sum_{j=1}^{M} e_{ij}^{p,q} V_{pq} V_{ij}
\]

where \((i,j) \neq (p,q)\) represents \( q \neq j \) when \( i = p \).

The motion equation is defined as following.

\[
\frac{dU_{ij}}{dt} = -\frac{\partial E}{\partial V_{ij}} = -A_1 \sum_{p=1}^{N} \sum_{q=1}^{M} e_{ij}^{p,q} V_{pq} \\
- A_2 \sum_{p=1}^{N} \sum_{q=1}^{M} e_{ij}^{p,q} V_{pq} V_{ij} \\
+ B \cdot h(\sum_{p=1}^{N} \sum_{q=1}^{M} e_{ij}^{p,q} V_{pq}) \\
+ C(1 - V_{ij})
\]

where the term \( A_1 \) is obtained by differentiating partially the energy function with respect to \( V_{ij} \). In the case of \((t \ mod \ T_{\omega}) \geq \omega \), the term \( A_1 \) is used, and in the other case the term \( A_2 \) is used. The omega function ejects solution from the local minimum by encouraging competition between neurons. The term \( B \) is a hill-climbing term which serves escaping from the local minimum, by assigning channels have no interference from surrounding cells. The function \( h(x) \) is 1 if \( x < 0 \), in other cases \( h(x) = 0 \). The term \( C \) is shaking term. By deterring immobilization of the channel assignment, the shaking term encourage escaping from local minimum.

\[
T_n = T_s \times a^n
\]

where \( T_n \) represents used hours of shaking term, \( T_s \) represents used interval and \( A_1, A_2, B, \) and \( C \) is coefficient.

For advancement of solution accuracy the regular interval assignment for the most congested cell is adapted. The interference is minimized in the cell because cells which have many demands tend to generate interference.

For profiling this algorithm, we implement this MMN in C language. 10 CAP benchmark problems are used in [1] shown in Figure 3. The same parameter values as [2] used in simulations. The compatibility matrix \( C \) and the demand vector \( D \) which are used in the KUNZ problems are obtained by considering only the first 10 regions in the KUNZ1 problem, 15 regions in the KUNZ2, 20 regions in the KUNZ3 and the entire data set in the KUNZ4.
For each benchmark problem, using different initialized values for each neuron, the CAP is simulated 10 times by the implemented algorithm and average maximum renewal times are obtained by results. Where, the maximum renewal time means a maximum taken time for updating minima of total interference. Also, proportion of demands to elements of $N \times M$ matrix is added for data. The simulation results are shown in Table 1. Req represents the total number of requirements, Per does proportion of demands, Sol does the average solving times and Ren does the average maximum renewal times. The Sol and Ren increases in proportion to the Per is seen. For each trial, over 300 updating, escaping from the local minimum tends to difficult because over large value of $U_{ij}$ interrupts.

4. Proposed improvement methods for MNN

To improve the solution accuracy, a large hill-climbing is executed for neurons after a certain period of renewal times. Using the values of $U$ the large hill-climbing makes neurons a partway convergent state and it leads to better convergence for the solution.

\[
R = D_i \times 0.6 \quad \quad \quad (6)
\]

\[
U_{ij} = \begin{cases} 
2 : & (\text{if } U_{ij} \geq U_R) \\
0 : & (\text{else})
\end{cases} \quad \quad \quad (7)
\]

where $R$ is an integer used for thresholds and the eq.(7) shows six out of ten $D_i$ are assigned to $R$. The inputs $U_{ij}$ of neurons in each cell are sorted in the descending order and the beginning to the R-th neurons are selected and there initialized input data is 2 and other neurons input data is 0.

\[
S = 280 \times P + 50 \quad \quad \quad (8)
\]

Eq.(9) is a linear equation relating with the proportion of demands $P$ and the large hill-climbing time $S$ obtained from analyzed result of benchmarks. If the minimum of total interference is not updated for $S$ times, the energy function is assumed to lapse into the local minimum and large hill-climbing is executed to escape.

Multi-thread processing is a measure to perform plural threads in parallel in a process. There is no need to switch memory spaces at threads switching and through the memory shares can data mutually thus processing is executed with few resources and overheads.

The procedures of the NNAs are following.
1. Assigning initial values for neurons input $U$.
2. Deriving neurons output $V$ from $U(U)$. 

Table 1: Data for profiling and simulation results.

<table>
<thead>
<tr>
<th>Problem</th>
<th>N</th>
<th>M</th>
<th>Req</th>
<th>Per</th>
<th>Sol</th>
<th>Ren</th>
</tr>
</thead>
<tbody>
<tr>
<td>EX1</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>2</td>
<td>0.2</td>
<td>8.01</td>
</tr>
<tr>
<td>EX2</td>
<td>5</td>
<td>17</td>
<td>13</td>
<td>0.15</td>
<td>41.1</td>
<td>13.7</td>
</tr>
<tr>
<td>HEX1</td>
<td>21</td>
<td>37</td>
<td>120</td>
<td>0.15</td>
<td>276.3</td>
<td>53.4</td>
</tr>
<tr>
<td>HEX2</td>
<td>21</td>
<td>91</td>
<td>120</td>
<td>0.06</td>
<td>195</td>
<td>40.4</td>
</tr>
<tr>
<td>HEX3</td>
<td>21</td>
<td>21</td>
<td>112</td>
<td>0.25</td>
<td>291</td>
<td>90.3</td>
</tr>
<tr>
<td>HEX4</td>
<td>21</td>
<td>56</td>
<td>112</td>
<td>0.10</td>
<td>188</td>
<td>46.7</td>
</tr>
<tr>
<td>KUNZ1</td>
<td>10</td>
<td>30</td>
<td>72</td>
<td>0.24</td>
<td>249</td>
<td>82.3</td>
</tr>
<tr>
<td>KUNZ2</td>
<td>15</td>
<td>44</td>
<td>113</td>
<td>0.17</td>
<td>238</td>
<td>47.1</td>
</tr>
<tr>
<td>KUNZ3</td>
<td>20</td>
<td>60</td>
<td>140</td>
<td>0.11</td>
<td>198</td>
<td>42.5</td>
</tr>
<tr>
<td>KUNZ4</td>
<td>25</td>
<td>73</td>
<td>167</td>
<td>0.09</td>
<td>177</td>
<td>37.8</td>
</tr>
</tbody>
</table>

(d) Examples of demand vectors.
Figure 3: The benchmark problems.

<table>
<thead>
<tr>
<th>Problem</th>
<th>N</th>
<th>M</th>
<th>Req</th>
<th>Per</th>
<th>Sol</th>
<th>Ren</th>
</tr>
</thead>
<tbody>
<tr>
<td>EX1</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>2</td>
<td>0.2</td>
<td>8.01</td>
</tr>
<tr>
<td>EX2</td>
<td>5</td>
<td>17</td>
<td>13</td>
<td>0.15</td>
<td>41.1</td>
<td>13.7</td>
</tr>
<tr>
<td>HEX1</td>
<td>21</td>
<td>37</td>
<td>120</td>
<td>0.15</td>
<td>276.3</td>
<td>53.4</td>
</tr>
<tr>
<td>HEX2</td>
<td>21</td>
<td>91</td>
<td>120</td>
<td>0.06</td>
<td>195</td>
<td>40.4</td>
</tr>
<tr>
<td>HEX3</td>
<td>21</td>
<td>21</td>
<td>112</td>
<td>0.25</td>
<td>291</td>
<td>90.3</td>
</tr>
<tr>
<td>HEX4</td>
<td>21</td>
<td>56</td>
<td>112</td>
<td>0.10</td>
<td>188</td>
<td>46.7</td>
</tr>
<tr>
<td>KUNZ1</td>
<td>10</td>
<td>30</td>
<td>72</td>
<td>0.24</td>
<td>249</td>
<td>82.3</td>
</tr>
<tr>
<td>KUNZ2</td>
<td>15</td>
<td>44</td>
<td>113</td>
<td>0.17</td>
<td>238</td>
<td>47.1</td>
</tr>
<tr>
<td>KUNZ3</td>
<td>20</td>
<td>60</td>
<td>140</td>
<td>0.11</td>
<td>198</td>
<td>42.5</td>
</tr>
<tr>
<td>KUNZ4</td>
<td>25</td>
<td>73</td>
<td>167</td>
<td>0.09</td>
<td>177</td>
<td>37.8</td>
</tr>
</tbody>
</table>

Table 1: Data for profiling and simulation results.

For each benchmark problem, using different initialized values for each neuron, the CAP is simulated 10 times by the implemented algorithm and average maximum renewal times are obtained by results. Where, the maximum renewal time means a maximum taken time for updating minima of total interference. Also, proportion of demands to elements of $N \times M$ matrix is added for data. The simulation results are shown in Table 1. Req represents the total number of requirements, Per does proportion of demands, Sol does the average solving times and Ren does the average maximum renewal times. The Sol and Ren increases in proportion to the Per is seen. For each trial, over 300 updating, escaping from the local minimum tends to difficult because over large value of $U_{ij}$ interrupts.
3. Updating $U$ with the motion equation ($V$).
4. Calculating total interference and keeping the minimum value ($W$).
5. Goto to 2 until termination conditions are satisfied.

Multi-threaded processing of procedure 2, 3 and 4 for neurons groups is employed. Global variables are used for shared data. Neuron outputs $V$ is shared data and each thread have neurons inputs $U$, copied neuron output $V$ and total interference as local data. While to prevent competing data synchronization is needed, each thread has thread-ID and cells are processed based on thread-ID thus competing not happened at updating $V$ to global from local memory. Hence, synchronization is needed only when shared data reading. Additionally copying local data from global data every renewal to minimize dealing with shared data decreases the synchronization. For additionally faster processing the multi-threaded processing for procedure 4, which is a bottleneck, is proposed. $W$ holds neuron outputs $V$ and total interference as shared data and copied $V$ as local data. In $U$, because of multi-threaded processing, total interference translate into shared data and synchronization are needed at data sharing. Additionally speculative execution of $U$ and $V$ during processing $W$ is used. There is dependency between $U$ and $W$. However, synchronizing start of data read from shared data resolves it. The numbers of threads used for executing $U$, $V$ and $W$ is $K/2$ and $K/2$ ($K$ is even) or $(K−1)/2+1$ and $(K−1)/2$ ($K$ is odd) respectively, where $K$ is the total number of threads in parallel. Figure 4 shows an example if the number of threads is 4.

![Figure 4: Multi-threads processing with speculative exec.](image-url)

5. Simulation Results

The proposed methods are implemented in C language on dual Xeon X5450, 3.0GHz, 4 core, 16G Mem, Linux 2.6.18, GCC 4.1.2 and compiler option -O2, and we simulate 10 times with different initial random values for each benchmark problem. We show the results in Table 2, which are the solution accuracy of the proposed method and Ike-naga’s one. “AV” represents the average of the objective values and “Min” represents the minimum solutions. As for the average values, the proposed algorithm is the same as or better than Ike-naga over all problems and as for the minimum value better values are seen in HEX2 and HEX3.

Table 3 shows the computation times of proposed method and existing algorithm(Ike.), where the number of threads is 2 to 8. Minimum time is seen from 7 threads processing. 7 threads processing leads to better outcomes than above for the all problems and achieve up to twice faster processing and about up to 4 times faster processing than the existing algorithm.

![Figure 4: Multi-threads processing with speculative exec.](image-url)

### Table 2: Simulation results for solution accuracy

<table>
<thead>
<tr>
<th>Problem</th>
<th>Ike.</th>
<th>Proposed algorithm</th>
<th>AV</th>
<th>Min</th>
<th>AV</th>
<th>Min</th>
</tr>
</thead>
<tbody>
<tr>
<td>EX1</td>
<td>6.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>EX2</td>
<td>6.11</td>
<td>0.21</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>HEX1</td>
<td>16.12</td>
<td>46.80</td>
<td>46.80</td>
<td>46.80</td>
<td>46.80</td>
<td>46.80</td>
</tr>
<tr>
<td>HEX3</td>
<td>78.66</td>
<td>77.74</td>
<td>77.74</td>
<td>77.74</td>
<td>77.74</td>
<td>77.74</td>
</tr>
<tr>
<td>HEX4</td>
<td>10.02</td>
<td>10.02</td>
<td>10.02</td>
<td>10.02</td>
<td>10.02</td>
<td>10.02</td>
</tr>
<tr>
<td>KUNZ1</td>
<td>40.81</td>
<td>40.81</td>
<td>40.81</td>
<td>40.81</td>
<td>40.81</td>
<td>40.81</td>
</tr>
<tr>
<td>KUNZ2</td>
<td>31.24</td>
<td>30.98</td>
<td>30.98</td>
<td>30.98</td>
<td>30.98</td>
<td>30.98</td>
</tr>
<tr>
<td>KUNZ4</td>
<td>0.61</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

### Table 3: Execution times of proposed multi-threads processing (sec.)

<table>
<thead>
<tr>
<th>Problem</th>
<th>Ike.</th>
<th>Proposed algorithm</th>
<th>2(1:1)</th>
<th>3(2:1)</th>
<th>4(2:2)</th>
<th>5(3:2)</th>
<th>6(3:3)</th>
<th>7(4:3)</th>
<th>8(4:4)</th>
<th>Ike.</th>
</tr>
</thead>
<tbody>
<tr>
<td>EX1</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>EX2</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>HEX1</td>
<td>2.16</td>
<td>1.55</td>
<td>1.14</td>
<td>0.96</td>
<td>0.89</td>
<td>0.79</td>
<td>0.79</td>
<td>0.79</td>
<td>0.79</td>
<td>3.10</td>
</tr>
<tr>
<td>HEX2</td>
<td>13.65</td>
<td>9.67</td>
<td>6.85</td>
<td>4.24</td>
<td>4.31</td>
<td>4.12</td>
<td>4.01</td>
<td>4.01</td>
<td>4.01</td>
<td>8.48</td>
</tr>
<tr>
<td>HEX3</td>
<td>0.73</td>
<td>0.56</td>
<td>0.44</td>
<td>0.42</td>
<td>0.38</td>
<td>0.41</td>
<td>0.41</td>
<td>0.41</td>
<td>0.41</td>
<td>1.01</td>
</tr>
<tr>
<td>HEX4</td>
<td>5.01</td>
<td>3.52</td>
<td>2.57</td>
<td>2.05</td>
<td>2.06</td>
<td>1.67</td>
<td>1.98</td>
<td>1.98</td>
<td>1.98</td>
<td>7.44</td>
</tr>
<tr>
<td>KUNZ1</td>
<td>0.39</td>
<td>0.38</td>
<td>0.28</td>
<td>0.28</td>
<td>0.24</td>
<td>0.27</td>
<td>0.27</td>
<td>0.27</td>
<td>0.27</td>
<td>0.53</td>
</tr>
<tr>
<td>KUNZ2</td>
<td>1.42</td>
<td>1.25</td>
<td>1.05</td>
<td>0.92</td>
<td>0.81</td>
<td>0.78</td>
<td>0.78</td>
<td>0.78</td>
<td>0.78</td>
<td>2.38</td>
</tr>
<tr>
<td>KUNZ3</td>
<td>5.26</td>
<td>3.75</td>
<td>2.81</td>
<td>2.21</td>
<td>2.01</td>
<td>1.78</td>
<td>1.78</td>
<td>1.78</td>
<td>1.78</td>
<td>7.44</td>
</tr>
<tr>
<td>KUNZ4</td>
<td>12.03</td>
<td>8.26</td>
<td>6.59</td>
<td>4.97</td>
<td>4.59</td>
<td>3.85</td>
<td>4.45</td>
<td>4.45</td>
<td>4.45</td>
<td>13.43</td>
</tr>
</tbody>
</table>

### 6. Conclusions

In this paper, we present a faster and more accurate processing methods for the existing neural network algorithm for the channel assignment problem in cellular radio networks and evaluate it’s performance. A large hill-climbing method is used to improve solution accuracy and multi-thread processing is employed to speed-up processing up to twice faster and about up to 4 times faster than the existing algorithm.

As future works, we will address the dynamic channel assignment problem and hardware implementation with the proposed algorithm.

### References

Effect of Piecewise Linear Function on Maximum-Flow Neural Network

Masatoshi Sato†, Hisashi Aomori‡, and Mamoru Tanaka†

†Department of Electrical and Electronics Engineering, Sophia University
7-1, Kioi-cho, Chiyoda-ku, Tokyo 102-8554, Japan
Phone: +81-3-3238-3878, Fax: +81-3-3238-3321
Email: masatoshi@tlab.ee.sophia.ac.jp
‡Department of Electrical Engineering, Tokyo University of Science
2641 Yamazaki, Noda-shi, Chiba, 278–8510 Japan
Phone: +81–4–7124–1501 (Ext. 3722), Fax: +81–4–7124–9367
Email: aomori@ieee.org

Abstract—In our previous research, the Maximum-Flow Neural Network (MF-NN) was proposed, and we showed that the MF-NN is possible to solve any maximum-flow problems. Each neuron of the MF-NN is connected by nonlinear resistances with the saturation saturation property. However, the conventional MF-NN using the sigmoidal function has problems where the sigmoidal function does not converge to 0, 1 that is saturation value. In this research, the saturation property is improved by using the piecewise linear function. Moreover, this novel method is possible to considerably reduce a calculation cost.

1. Introduction

The spread of the Internet continues to increase in modern society. Thus, having a method to send data quickly with a little loss is very important. This problem is commonly known as the maximum flow problem [1, 2], and the maximum flow algorithm offers the best solution to the problem of determining which route is appropriate to exchange data. Hence, the importance of the maximum flow algorithm is growing from the viewpoint of transportation capacity. The maximum flow problem involves streaming a large quantity of the flow from a starting point to a terminal point in the given network. In general, an information stream in a branch that satisfies the following conditions is called a flow: the flow does not exceed the capacity of the branch (the capacity condition), and the sum of the inflows is equal to the sum of the outflows (the flow preservation condition) on each node. The nodes incident to a branch exist in three-dimensional spaces in a network, and they are connected to the branch such that a graph consists of nodes and branches. The capacity of the branch (communication capacity) and the gain exist in each branch. The capacity indicates the limit of inflow or outflow, and the gain can enlarge or reduce the flow [3].

Over the years, various algorithms have been developed to solve the maximum flow problem. The Ford-Fulkerson algorithm [4, 5] and the preflow-push algorithm [6, 7] are two well known methods developed to solve this maximum flow problem. However, these algorithms are for computation by sequential machines without analog parallelism.

In our previous research, we proposed the Maximum-Flow Neural Network (MF-NN) [8], and we showed that the MF-NN is possible to solve any maximum-flow problems. The MF-NN in which each branch nonlinearity has a saturation characteristic and by which the maximum flow problem can be solved by using analog high-speed parallel processing [9, 10]. In a nonlinear resistive network, each nonlinear resistance with the saturation for the sigmoidal function is defined by

\[ f(x) = \begin{cases} \frac{x}{\max(0, x)} & 0 < f(x) < 1 \\ f(x) & f(x) \leq 0 \text{ or } f(x) \geq 1 \end{cases} \]

The monotone increasing nonlinear function with a saturation value. In our previous research, the Maximum-Flow Neural Network (MF-NN) was proposed, and we showed that the MF-NN is possible to solve any maximum-flow problems. Each neuron of the MF-NN is connected by nonlinear resistances with the saturation saturation property. However, the conventional MF-NN using the sigmoidal function has problems where the sigmoidal function does not converge to 0, 1 that is saturation value. In this research, the saturation property is improved by using the piecewise linear function. Moreover, this novel method is possible to considerably reduce a calculation cost.

2. The maximum flow algorithm

The maximum flow theorem is one of the most basic theorems of the network flow problem. When a positive constant branch capacity \( c_b \) is allocated in each branch \( b \in N \) on network \( N \) as a graph, the \( N \) is a communication network or transportation network. It is shown as \( N = (V(N), E(N)) \) for nodes set \( V[N] \) and branches set \( E[N] \). The notation \( C(N) \) is a set of branch capacity \( \{c_b\} \). A directed branch \( b \) connected from node \( v_i \) to \( v_j \) on \( N \) is denoted by \( b = (v_i, v_j) = b_{ij} \). A flow \( f \) from \( s \in V(N) \) to \( t (t \neq s) \in V(N) \) in the communication network \( N \) is defined by

\[ \sum_{v_i \in F(s)} f(v_i, v_j) - \sum_{v_i \in F^{-1}(v)} f(v_j, v_i) = \begin{cases} F: v_i = s \\ -F: v_i = t \\ 0 : v_i \neq s, t, \end{cases} \]

where \( F \) and \( -F \) are the amounts of inflow and outflow respectively.
\[ 0 \leq f(v_i, v_j) \leq c(v_i, v_j), \]
\[ (v_i, v_j) \in B(N), \] \tag{2}

where

\[ \Gamma(v_i) = v_j | (v_i, v_j) \in B(N), \]
\[ \Gamma^{-1}(v_i) = v_j | (v_j, v_i) \in B(N). \] \tag{3}

\( F = F(f) \) in Eq. (1) is the value of flow \( f \), and node \( s \) and \( t \) are the source and the sink respectively.

Let the left part of Eq. (1) be the flow that flows out from \( v_i \), then the Eq. (1) represents the restriction, where the flow from source \( s \) is \( F \) and the flow from sink \( t \) is \(-F\) and the flow from arbitrary node \( v_i \neq s, t \) is 0. Also, if the flow \( f(v_i, v_j) \) that flows in each \((v_i, v_j) \in B(N)\) is the branch flow, then Eq. (2) represents the restriction where the branch flowing on each branch \( b_{ij} \) flows only in the direction from \( v_i \) to \( v_j \) and it does not exceed the branch capacity \( c(v_i, v_j) \). In the communication network \( N \), a branch set \((X, Y)\) is defined as

\[ (X, Y) = \{(v_i, v_j) \in B(N) | v_i \in X, v_j \in Y\}, \] \tag{4}

where \( X, Y \subset V(N) \). The branch class \((X, Y)\) has a source on \( X \) and sink on \( Y \). For arbitrary flow \( f \), the flow \( f(X, Y) \) which flows \((X, Y)\) is given by

\[ f(X, Y) = \sum_{(v_i, v_j) \in (X, Y)} f(v_i, v_j). \] \tag{5}

In the flow \( f \) of the communication network \( N \), a flow \( f_0 \) that gives the maximum value of \( F(f) \) expressed by \( F_0 = \max[F(f)] \) is called the maximum-flow.

3. The maximum-flow neural network

3.1. Conventional MF-NN

The network topology of the MF-NN is shown in Fig. 1. An input layer and an output layer of the network are corresponding to the start point \( S \) and the terminal point \( T \), and each point corresponds to a single neuron, and layer of inner layer is \( N \). Additionally, the propagation between the neighboring neurons is interactive. That is, the MF-NN has feedback connections. The layer number of inner layer changes depending on how to connect the neurons. The structure is determined by a given transportation network. Fig. 2 shows the connection between the neuron \( v_i \) and the neuron \( v_j \) by the nonlinear resistive network. The \( A_{ij} \) is nonlinear resistance that exists between the neuron \( v_i \) and the neuron \( v_j \). The MF-NN has the saturation characteristic such that the entire network converges to the equilibrium state if a certain amount of the current in the starting point \( s \) goes out. The \( f - V \) branch characteristic from the neuron \( v_i \) to the neuron \( v_j \) is described by

\[ I_{ij} = A_{ij}(u_i - u_j), \] \tag{6}

where

\[ f(x) = \frac{1}{1 + exp(-ax)} \] \tag{7}

\( A_{ij} \) is the maximum capacity \( c(v_i, v_j) \) of the current in \( b_{ij} \) and a positive constant value. \( I_{ij} \) is a current that flows from the neuron \( v_i \) to \( v_j \), and \( u_i \) and \( u_j \) are node voltages of the neuron \( v_i \) and \( v_j \) respectively. The constant \( a \) is the gain. The state equation with respect to the neuron \( v_i \) is given by

\[ F_i(u_i) \equiv C_i \frac{du_i}{dt} = -\sum_{v_j \in \Gamma(v_i)} A_{ij} f(u_i - u_j) + \sum_{v_k \in \Gamma^{-1}(v_i)} A_{ki} f(u_k - u_i). \] \tag{8}

where

\[ \Gamma(v_i) = v_j | (v_i, v_j) \in B(N), \]
\[ \Gamma^{-1}(v_i) = v_j | (v_j, v_i) \in B(N). \] \tag{9}

\( C_i \) is a capacitor that exists between the neuron \( v_i \) and the ground. By solving the differential equation Eq. (8) concerning the neuron \( v_i \) (\( i = 1, 2, \ldots, n \)), the state of each neuron (node voltage) \( u \) can be obtained. As a result, the potential differences between neurons and the branch current value corresponding to the voltage differences between each neuron are obtained.

Combining the equations of the maximum flow algorithm were shown from Eq. (1) to Eq. (4), the state equa-

![Figure 1: The network topology of MF-NN](image1)

![Figure 2: Association between neuron i and j](image2)
tion of the MF-NN, Eq. (8) can be rewritten as
\[
\lim_{t \to \infty} C_i \frac{d u_i}{d t} = - \sum_{v \in \Gamma(v_i)} A_{ij} f(u_i - u_j) + \sum_{v \in \Gamma^{-1}(v_i)} A_{ji} f(u_j - u_i)
\]
\[
= \begin{cases} 
-F : v_i = s \\
F : v_i = t \\
0 : v_i \neq s, t,
\end{cases}
\] (10)
\[
0 \leq A_{ij} f(u_i - u_j) \leq A_{ij} = c(v_i, v_j),
\]
\[
(v_i, v_j) \in B(N).
\] (11)

Since the equilibrium state to fill Eq. (10) is described by
\[
\lim_{t \to \infty} C_i \frac{d u_i}{d t} = 0 \ (v_i \neq s, t),
\] (12)
Eq. (12) shows the state where the network is saturated. That is, only when the MF-NN is saturated, the state equation conform to the maximum flow algorithm. Moreover, since each current value (the flow) \(I_i\) becomes the maximum value in the saturated state, the \(F\) of Eq. (10) shows the maximum flow \(F_0\).

3.2. Problems of Conventional MF-NN

Essentially the nonlinear function \(f(x)\) should be used under the condition that \(f(x)\) satisfies the condition \(0 \leq I_{ij} \leq A_{ij}\). However, the \(I - V\) characteristic of the conventional MF-NN depends on the sigmoidal function as shown in Eq. (6), (7). The sigmoidal function is defined as \(0 < f(x) < 1\), therefore \(f(x)\) is approximately converged in the vicinity of \(f(x) = 0, f(x) = 1\).

In conventional MF-NN, the condition \(0 \leq I_{ij} \leq A_{ij}\) is realized by raising the gain \(a\). For example, the velocity that approaches \(f(x) = 0, 1\) can be changed by the difference of the gain \(a\) and \(f(x)\) is converged early by raising the gain \(a\). However, since differential coefficient approaches 0, it takes a lot of computing time until convergent solutions of the simultaneous differential equation of MF-NN are obtained.

3.3. Novel MF-NN using PWL function

To solve the problem of conventional MF-NN, other nonlinear function \(f(x)\) is used. Ideal \(f(x)\) to fill the maximum flow algorithm is given by
\[
f(x) = \begin{cases} 
1 \ (b \leq x), \\
g(x) \ (0 \leq x < b), \\
0 \ (x < 0),
\end{cases}
\] (13)
where, \(b\) is positive constant and \(g(x)\) is monotonically increasing function. Since \(f(x)\) is constantly a positive function, the backflow phenomenon where a positive current flowed to a negative potential difference was occasionally generated in conventional MF-NN. To avoid this backflow phenomenon, the function that passes point \(f(0) = 0\) and \(f(a) = 1\) is selected. The nonlinear function \(f(x)\) that contains those conditions is defined by,
\[
f(x) = \begin{cases} 
1 \ (b \leq x), \\
x \ (0 \leq x < b), \\
0 \ (x < 0),
\end{cases}
\] (14)
Since the PWL function has the feature with constant differential coefficient, the function has an advantage that the solution can be obtained at the very high speed in the numerical analysis of the nonlinear simultaneous differential equation that uses Newton method.

4. Simulation Results

In this research, we use a network which has mutual coupling as shown in Fig. 3. This network has the start-point node \(u_1\), the inner layer nodes \(u_1, u_2, u_3, u_4\), and the terminal node \(u_5\). The initial voltage of the network analyzed is \(V = 10V\). And, there are mutual coupling branches \(b_{ij}(i, j = 1, 2, 3, 4)\), and the branch capacities \(c_{ij}\) are given on each branch \(b_{ij}\) as shown in Fig. 3. Since \(c_{ij} \neq c_{ji}\), we show that the novel MF-NN can solve complex problems in maximum flow problems.

The problem of the conventional MF-NN is that the error of a maximum flow occurs especially in the case of mutual coupling network analysis. The network shown in Fig. 3 is small-scale, but enough results to compare the performances is obtained.

The influence on the solution by the difference in gain \(a\) of various patterns is entertained. The content \(b\) of the PWL function Eq. (14) is set to \(b = 1\). Under these conditions, the simulation results are compared.

Table 1 is a table where the current value (flow) of each branch by the difference of each gain \(a\) and each maximum flow are shown. Table 2 shows the calculation frequency where the node voltage of node 1 converges. As a result, in the case of conventional MF-NN, the error margin of the maximum flow becomes small when the the gain \(a\) is raised. However, there is a fault of an awful lot the calculation frequency, and taking a lot of converging time. On the other hand, proposed MF-NN using PWL function doesn’t have the error margin of the maximum flow, and the calculation processing time is also very fast compared with conventional MF-NN.

![Figure 3: Analyzed network](image-url)
Table 1: Comparison between result by difference of gain a of conventional MF-NN and result by the novel MF-NN

<table>
<thead>
<tr>
<th>branch from A to B</th>
<th>Nonlinear Resistance</th>
<th>Gain 0.5</th>
<th>Gain 1.0</th>
<th>Gain 1.5</th>
<th>Gain 2.0</th>
<th>Gain 3.0</th>
<th>PWL function</th>
</tr>
</thead>
<tbody>
<tr>
<td>S → 1</td>
<td>60.00 (L)</td>
<td>46.0099 (A)</td>
<td>56.5324 (A)</td>
<td>59.2854 (A)</td>
<td>59.8621 (A)</td>
<td>59.9926 (A)</td>
<td>60.00 (A)</td>
</tr>
<tr>
<td>S → 3</td>
<td>10.00 (L)</td>
<td>9.4537 (A)</td>
<td>9.9741 (A)</td>
<td>9.9999 (A)</td>
<td>10.00 (A)</td>
<td>10.00 (A)</td>
<td>10.00 (A)</td>
</tr>
<tr>
<td>1 → 4</td>
<td>30.00 (L)</td>
<td>25.9011 (A)</td>
<td>26.8279 (A)</td>
<td>29.7894 (A)</td>
<td>29.9518 (A)</td>
<td>29.9998 (A)</td>
<td>30.00 (A)</td>
</tr>
<tr>
<td>1 → 5</td>
<td>50.00 (L)</td>
<td>15.9011 (A)</td>
<td>16.9356 (A)</td>
<td>19.9609 (A)</td>
<td>19.9999 (A)</td>
<td>20.00 (A)</td>
<td>20.00 (A)</td>
</tr>
<tr>
<td>1 → 6</td>
<td>10.00 (L)</td>
<td>9.4993 (A)</td>
<td>9.9821 (A)</td>
<td>9.9999 (A)</td>
<td>10.00 (A)</td>
<td>10.00 (A)</td>
<td>10.00 (A)</td>
</tr>
<tr>
<td>2 → 1</td>
<td>20.00 (L)</td>
<td>2.8659 (A)</td>
<td>3.8914 (A)</td>
<td>4.0064 (A)</td>
<td>4.0329 (A)</td>
<td>4.0011 (A)</td>
<td>4.00 (A)</td>
</tr>
<tr>
<td>2 → 2</td>
<td>50.00 (L)</td>
<td>15.9346 (A)</td>
<td>16.9332 (A)</td>
<td>19.9398 (A)</td>
<td>19.9319 (A)</td>
<td>2.35 (A)</td>
<td>2.35 (A)</td>
</tr>
<tr>
<td>2 → 3</td>
<td>10.00 (L)</td>
<td>7.0013 (A)</td>
<td>7.8841 (A)</td>
<td>8.1280 (A)</td>
<td>8.2131 (A)</td>
<td>8.2299 (A)</td>
<td>7.85 (A)</td>
</tr>
<tr>
<td>3 → 1</td>
<td>10.00 (L)</td>
<td>1.6065 (A)</td>
<td>4.0304 (A)</td>
<td>4.0812 (A)</td>
<td>4.0155 (A)</td>
<td>4.0005 (A)</td>
<td>4.00 (A)</td>
</tr>
<tr>
<td>3 → 2</td>
<td>50.00 (L)</td>
<td>15.9617 (A)</td>
<td>16.9615 (A)</td>
<td>19.9640 (A)</td>
<td>19.9631 (A)</td>
<td>19.9999 (A)</td>
<td>20.00 (A)</td>
</tr>
<tr>
<td>3 → 3</td>
<td>10.00 (L)</td>
<td>9.4993 (A)</td>
<td>9.9821 (A)</td>
<td>9.9999 (A)</td>
<td>10.00 (A)</td>
<td>10.00 (A)</td>
<td>10.00 (A)</td>
</tr>
<tr>
<td>3 → 4</td>
<td>50.00 (L)</td>
<td>36.6314 (A)</td>
<td>36.8314 (A)</td>
<td>39.9999 (A)</td>
<td>40.1801 (A)</td>
<td>39.9999 (A)</td>
<td>2.35 (A)</td>
</tr>
<tr>
<td>4 → 1</td>
<td>20.00 (L)</td>
<td>1.3015 (A)</td>
<td>4.2159 (A)</td>
<td>4.0379 (A)</td>
<td>4.0070 (A)</td>
<td>4.0002 (A)</td>
<td>4.00 (A)</td>
</tr>
<tr>
<td>4 → 2</td>
<td>50.00 (L)</td>
<td>15.9842 (A)</td>
<td>16.9841 (A)</td>
<td>19.9889 (A)</td>
<td>19.9898 (A)</td>
<td>19.9999 (A)</td>
<td>20.00 (A)</td>
</tr>
<tr>
<td>4 → 3</td>
<td>10.00 (L)</td>
<td>7.0092 (A)</td>
<td>8.1824 (A)</td>
<td>5.6484 (A)</td>
<td>5.5334 (A)</td>
<td>5.5099 (A)</td>
<td>5.50 (A)</td>
</tr>
<tr>
<td>5 → 1</td>
<td>20.00 (L)</td>
<td>3.6948 (A)</td>
<td>19.6446 (A)</td>
<td>19.9999 (A)</td>
<td>19.9999 (A)</td>
<td>19.9999 (A)</td>
<td>20.00 (A)</td>
</tr>
<tr>
<td>5 → 2</td>
<td>50.00 (L)</td>
<td>15.9542 (A)</td>
<td>16.9541 (A)</td>
<td>19.9898 (A)</td>
<td>19.9898 (A)</td>
<td>19.9999 (A)</td>
<td>20.00 (A)</td>
</tr>
</tbody>
</table>

Maximum Flow: (1, 1) → (C, 1) 56.8332 (A) 66.4866 (A) 66.9621 (A) 69.9999 (A) 10.00 (A) 10.00 (A)

Table 2: Calculation frequency until converging

<table>
<thead>
<tr>
<th>Gain 0.5</th>
<th>Gain 1.0</th>
<th>Gain 1.5</th>
<th>Gain 2.0</th>
<th>Gain 3.0</th>
<th>PWL function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1220(msec)</td>
<td>1300(msec)</td>
<td>2440(msec)</td>
<td>6000(msec)</td>
<td>60000(msec)</td>
<td>600000(msec)</td>
</tr>
</tbody>
</table>

5. Conclusions

In this paper, a novel MF-NN using PWL function was proposed. Simulation results indicated that convergence of the sigmoidal function of conventional MF-NN greatly influences at accuracy and the convergence time of the solution. The error margin with the correct maximum flow became small by raising the gain $a$. However, there was a fault that takes a lot of computation time. On the other hand, the equivalent result of the maximum flow algorithm was able to be obtained by using novel MF-NN using PWL function. In addition, the computation time was sped up very much. Along with making of more large-scale network, the error margin of the maximum flow and the computation time of conventional MF-NN become larger problems. Therefore, novel MF-NN has been improved to a very superior algorithm.

Acknowledgment

This research was supported by funding from the Open Research Center Project of the Ministry of Education, Culture, Sports, Science and Technology (MEXT) of the Japanese Government(2007-2011)

References


Computer assisted proofs of solutions to Nonlinear elliptic partial differential equations

Akitoshi Takayasu†, Shin’ichi Oishi‡ and Takayuki Kubo§

†Graduate School of Fundamental Science and Engineering, Waseda University
3-4-1 Okubo, Shinjuku-ku, Tokyo 169-8555, Japan
‡Faculty of Science and Engineering, Waseda University & CREST, JST,
3-4-1 Okubo, Shinjuku-ku, Tokyo 169-8555, Japan
§Institute of Mathematics, University of Tsukuba,
Tenodai 1-1-1, Tsukuba, Ibaraki 305-8571 Japan
Email: takitoshi@suou.waseda.jp, oishi@waseda.jp, tkubo@math.tsukuba.ac.jp

Abstract—In this article, a numerical method is presented for computer assisted proofs to the existence and uniqueness of solutions to Dirichlet boundary value problems in a certain class of nonlinear elliptic equations. In a weak formulation of the problem, a weak solution is described as a zero point of a certain nonlinear map. Based on Newton-Kantorovich theorem, a numerical existence and local uniqueness of solutions are proved by our proposed method. Some conditions need to be checked numerically. It is shown that all errors of numerical computations such as discretization errors and rounding errors are figured out by numerical computations with result verification. Finally, an illustrative numerical result is presented for showing the usefulness of proposed method.

1. Introduction

Let $\Omega$ be a bounded convex polygonal domain in $\mathbb{R}^m$ with $m = 2, 3$. This article is concerned with Dirichlet boundary value problems of nonlinear elliptic equations:

\[
\begin{cases}
-\nabla \cdot (a \nabla u) = f(u), & x \in \Omega, \\
u = 0, & x \in \partial \Omega,
\end{cases}
\tag{1}
\]

where $a(x)$ is a smooth function on $\overline{\Omega}$ with $a(x) \geq a_0 > 0$ for some $a_0 \in \mathbb{R}$. Here, $f : H^1_0(\Omega) \to L^2(\Omega)$ is assumed to be Fréchet differentiable. For example, the following function

\[
f(u) = -b \cdot \nabla u - cu + c_2 u^2 + c_3 u^3 + g
\]

with $b(x) \in (L^\infty(\Omega))^m, c, c_2, c_3 \in L^\infty(\Omega)$ and $g \in L^2(\Omega)$ satisfies this condition. We shall propose a numerical method of computer assisted proofs for the existence and local uniqueness of solutions to the problem (1).

In 1988, M. T. Nakao [1] has presented a method of a computer assisted proof for elliptic problems. In 1991, Plum [2] has also presented another method of proving the existence and uniqueness of solutions for the problem (1). Both methods have been demonstrated to be useful in this two decades. On the other hand, this article presents another method of computer assisted proofs for (1) based on the finite element method. In the following section, we describe how to work the proposal method. Then a computer assisted proof algorithm is presented in Section 3. Finally, we can show the illustrative result of our method.

2. Outline of proposal computer assisted proofs

In this part, we shall briefly sketch our proposed numerical method to prove the existence of weak solutions for (1). Proposed method also evaluates guaranteed error bounds in which there is an unique solution of original equations.

Let $H^{-1}(\Omega)$ be the topological dual space of $H^1_0(\Omega)$ the space of linear continuous functionals on $H^1_0(\Omega)$. For $u, v \in H^1_0(\Omega)$, let us define a continuous bilinear form $A(u, v)$ as

\[
A(u, v) = (a \nabla u, \nabla v).
\]

If we fix $u \in H^1_0(\Omega)$, then $A(u, \cdot) \in H^{-1}(\Omega)$. Thus, we can define an operator $\mathcal{A} : H^1_0(\Omega) \to H^{-1}(\Omega)$ by

\[
< \mathcal{A} u, v > = A(u, v).
\]

Let us define

\[
||u||_u = \sqrt{A(u, u)}.
\]

This norm is equivalent to $H^1_0$-norm, i.e., there exist positive constants $c_a$ and $c_a$ satisfying

\[
c_a ||u||_{H^1_0} \leq ||u||_u \leq C_a ||u||_{H^1_0}
\]

for $u \in H^1_0(\Omega)$.

In fact, we can choose $c_a = \sqrt{a_0}$ and $C_a = \sqrt{||a||_{L^\infty}}$. Further, we can define an operator $N : H^1_0(\Omega) \to H^{-1}(\Omega)$ by

\[
< Nu, v > = N(u, v) = (f(u), v),
\]

Then, a weak form of Eq. (1) can be written as

\[
\mathcal{A} u = N u.
\]

Now, let us define the operator $\mathcal{F} : H^1_0(\Omega) \to H^{-1}(\Omega)$ by

\[
\mathcal{F} u = (\mathcal{A} - N) u.
\]
Then, Eq. (2) can be written as
\[ F \hat{u} = 0. \] (2)

In the following, we will discuss how to prove the existence and uniqueness of the solution of Eq. (2), the weak solution of the problem (1). Newton-Kantorovich theorem is applicable to the nonlinear operator equation (2). This theorem gives our desired computer assisted proof for the existence and local uniqueness of solutions to Eq. (2).

In order to apply Newton-Kantorovich theorem, the Fréchet derivative of \( F \) is needed. The Fréchet differentiability of \( F \) is followed by that of \( f \). Moreover, the Fréchet derivative of \( N \) at \( \hat{u} \in H^1_0(\Omega) \) is bounded and enjoys
\[ \| (A^{-1} - P_n A^{-1}) N'(\hat{u}) \|_{L(H^1_0, H^1_0)} \leq L \]
for a certain positive \( \alpha \) and \( \omega \) in Theorem 1. Three constants are needed to evaluate. One is the inverse operator norm estimation.

Eq. (4) becomes
\[ (a(x) \nabla \hat{u}, \nabla \phi_h) = (f(\hat{u}), \phi_h), \quad (\forall \phi_h \in X_n), \]
which is nothing but the finite element approximation [6] of the nonlinear equation (2).

We also discuss how to calculate constants \( \alpha \) and \( \omega \) in Theorem 1. Three constants are needed to evaluate. One is the inverse operator norm estimation.

This is estimated by the following theorem given by S. Oishi [7]. This theorem is based on perturbation lemma of linear operators [8] and given as

**Theorem 2 (Oishi 1995)** Let \( \hat{u} \in H^1_0(\Omega) \) and \( N'(\hat{u}) \) be a linear compact operator. Let \( X_n \) be a finite dimensional subspace of \( H^1_0(\Omega) \) spanned by the finite element bases \( S_h = \{ \phi_1, \phi_2, \ldots, \phi_n \} \). Let \( P_n : H^1_0(\Omega) \rightarrow X_n \) be the Ritz-projection. Assuming that \( P_n A^{-1} N'(\hat{u}) : H^1_0(\Omega) \rightarrow H^1_0(\Omega) \) is bounded and satisfies
\[ \| P_n A^{-1} N'(\hat{u}) \|_{L(H^1_0, H^1_0)} \leq K, \]
the difference between \( A^{-1} N'(\hat{u}) \) and \( P_n A^{-1} N'(\hat{u}) \) is bounded and enjoys
\[ \| (A^{-1} - P_n A^{-1}) N'(\hat{u}) \|_{L(H^1_0, H^1_0)} \leq L. \]
and the finite dimensional operator $\mathcal{P}_n(I - \mathcal{A}^{-1}N'(\hat{u}))|_{X_n}: X_n \to X_n$ is invertible with
\[
\|((\mathcal{P}_n(I - \mathcal{A}^{-1}N'(\hat{u}))|_{X_n})^{-1}\|_{L(H^1_0, H^0_0)} \leq M.
\]
Here, $\mathcal{P}_n(I - \mathcal{A}^{-1}N'(\hat{u}))|_{X_n}: X_n \to X_n$ is the restriction of the operator $\mathcal{P}_n(I - \mathcal{A}^{-1}N'(\hat{u})): H^1_0(\Omega) \to X_n$ on $X_n$. If $(1 + MK)L < 1$, then the operator $\mathcal{A} - N'(\hat{u}): H^1_0(\Omega) \to H^{-1}(\Omega)$ is also invertible and
\[
\|((\mathcal{A} - N'(\hat{u}))^{-1})\|_{L(H^{-1}, H^1_0)} \leq \frac{1}{c^2} \frac{1 + MK}{1 - (1 + MK)L} =: C_1.
\]

For computer assisted proofs, next constant is the residual estimation of the operator equation (2). It is bounded by
\[
\|F\hat{u}\|_{H^{-1}} = \|\mathcal{A}\hat{u} - N\hat{u}\|_{H^{-1}} \leq \|\hat{u} - \mathcal{P}_n\mathcal{A}^{-1}N(\hat{u})\|_{H^1_0} + C_0(h)\|f(\hat{u})\|_{L^2} =: C_2.
\]

Further, the Lipschitz constant is the last constant to evaluate. It is defined through
\[
\|F'(v) - F'(w)\|_{L(H^{-1}, H^1_0)} \leq C_3\|v - w\|_{H^1_0}.
\]
Then it follows that $\alpha \leq C_1C_2$ and $\omega \leq C_2C_3$. If $\alpha \omega \leq C_1^2C_2C_3 < \frac{1}{2}$ is obtained by verified computation, then the existence and uniqueness of the solution are proved numerically. So that Newton-Kantorovich theorem can be applied to the nonlinear operator equation (2) if three constants are estimated.

3. Computer assisted existence test

In this section, summing up the above discussions, an algorithm of computer assisted proofs is described for verifying the existence and local uniqueness of solutions to Eq. (1) in the neighborhood of $\hat{u}$.

**Algorithm 1 (Nonlinear Elliptic Eq.)**

1. Compute an approximate solution $\hat{u} \in H^1_0(\Omega)$ of the problem (4)
2. Compute rigorous upper bound of $\|((\mathcal{A} - N'(\hat{u}))^{-1})\|_{L(H^{-1}, H^1_0)}$ by the following steps:
   1. Compute $\|\hat{u}\|_{L^2}$ and calculate $K, L$ by
   \[
   K = \frac{C_{2,2}}{\alpha_0} \|f'(\hat{u})\|_{L(H^1_0, L^2)},
   \]
   and
   \[
   L = C_0(h)\|f'(\hat{u})\|_{L(H^1_0, L^2)},
   \]
   respectively $^{1}$).

$^{1}$ $C_{2,2}$ denotes Poincaré constant $H^1_0(\Omega) \hookrightarrow L^2(\Omega)$ by Sobolev embedding theorem, ex. $\|u\|_{L^2} \leq C_{2,2}\|u\|_{H^1_0}$ for $u \in H^1_0(\Omega)$.

2.2 Let $D$ and $G$ be $n \times n$ matrices whose $i$-$j$ elements are given by
\[
(a(x)\nabla \phi_j, \nabla \phi_i),
\]
and
\[
(a(x)\nabla \phi_j, \nabla \phi_i) - (f'(\hat{u})\phi_j, \phi_i),
\]
respectively. Let a lower triangular matrix $\hat{L}$ be the Cholesky decomposition of $D$, $D = \hat{L}\hat{L}^T$. If $G$ is invertible, then set
\[
M = \frac{C_a}{c_a}\|\hat{E}^T\hat{G}\|_{L^2}.
\]
When $G$ is not invertible, stop with failure.

2.3 Check whether $(1 + MK)L < 1$ holds or not. If this holds, then by Theorem 2
\[
\|((\mathcal{A} - N'(\hat{u}))^{-1})\|_{L(H^{-1}, H^1_0)} \leq \frac{1 + MK}{\alpha_0(1 - (1 + MK)L)} =: C_1.
\]
Otherwise, stop with failure.

3. Calculate the residual by the formula
\[
C_2 := C_2^2\left(\|\hat{u} - \mathcal{P}_n\mathcal{A}^{-1}N(\hat{u})\|_{H^1_0} + C_0(h)\|f(\hat{u})\|_{L^2}\right).
\]
Set $\alpha = C_1C_2$.

4. Calculate the Lipschitz constant $C_3$ by
\[
C_3 := \left(\frac{C_2}{\alpha_0}\right)^2 C_{c,2}C_L
\]
where $C_L$ is the Lipschitz constant of $f'$.
Set $\omega = C_1C_3$.

5. Check the condition $\alpha \omega \leq \frac{1}{2}$. If this condition is satisfied, there is a solution $u^* \in H^1_0(\Omega)$ of $Fu = 0$ satisfying
\[
\|u^* - \hat{u}\|_{H^1_0} \leq \rho := \frac{1 - \sqrt{1 - 2\alpha \omega}}{\omega}.
\]
Further, the solution $u^*$ is unique in $B(\hat{u}, \rho)$. Otherwise, stop with failure.

4. Computational result

Now, we shall present a numerical result to illustrate the usefulness of our method. All computations are carried out on Mac OS X, 2.26GHz Quad-Core Intel Xeon by using MATLAB 2010a with a toolbox for verified computations, INTLAB [9].

For an application of our computer assisted proof method, we treat a nonlinear Dirichlet boundary value problem on $\Omega = (0, 1) \times (0, 1)$:

\[
\begin{aligned}
-\Delta u &= u^3 + 5 & x \in \Omega, \\
u(x) &= 0 & x \in \partial \Omega.
\end{aligned}
\]

(5)

Obviously, the Fréchet derivative of $f(u) = u^3 + 5 : H^1_0(\Omega) \to L^2(\Omega)$ is given by $f'(u) = 3u^2$. An approximate solution $\hat{u}$ is calculated by the finite element method
(pde toolbox on MATLAB) with uniform piecewise linear elements of the mesh size $\frac{1}{32}$. The approximate solution is bounded on $\Omega$ then $\hat{u}$ is the element of $L^\infty(\Omega)$ in this solution. So that for $\hat{u} \in L^\infty(\Omega) \cap H^1_0(\Omega)$, we can use

$$\|f'(\hat{u})\|_{L^2(H^1_0, L^2)} \leq 3 \min \left\{ C_{c_2} \|\hat{u}\|_{L^\infty}^2, C_{c_3} \|\hat{u}\|_{H^1_0}^2 \right\} \|\hat{c}_3\|_{L^\infty}.$$  

For the Lipschitz continuity of $f'(u)$, for $u \in H^1_0(\Omega)$ and $v, w \in B(\hat{u}, 2\alpha)$ we have

$$\|(f'(v) - f'(w))u\|_{L^2} \leq \|(3C_{c_2}) (v + w)(v - w)u\|_{L^2} \leq (3C_{c_2}^3 \|\hat{u}\|_{L^\infty} \|v + w\|_{H^1_0} \|v - w\|_{H^1_0} \|u\|_{L^\infty}.$$

Since $v, w \in B(\hat{u}, 2\alpha)$, it follows that

$$\|v + w\|_{H^1_0} \leq 2\|\hat{u}\|_{H^1_0} + 4\alpha.$$  

Thus, we have

$$\|f'(v) - f'(w)\|_{L^2(H^1_0, L^2)} \leq C_L \|v - w\|_{H^1_0(\Omega)},$$

with

$$C_L = 6C_{c_3} \|\hat{u}\|_{L^\infty} \left( \|\hat{u}\|_{H^1_0}^2 + 2\alpha \right).$$

Furthermore we focus on the computational cost of proposal method. It is several (up to 10) times more than that of the approximation. An illustrative result with respect to the performance is presented in Table 2. Here, we assume that $t_1$ is a computing time to get approximate solution.

**Table 2:** Comparing the execution time

<table>
<thead>
<tr>
<th>Mesh size: $\frac{1}{h}$</th>
<th>Approximate ($t_1$)</th>
<th>Verification ($t_1$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0.04</td>
<td>2.09</td>
</tr>
<tr>
<td>5</td>
<td>0.11</td>
<td>1.99</td>
</tr>
<tr>
<td>6</td>
<td>0.47</td>
<td>2.21</td>
</tr>
<tr>
<td>7</td>
<td>2.14</td>
<td>3.19</td>
</tr>
<tr>
<td>8</td>
<td>17.25</td>
<td>3.96</td>
</tr>
<tr>
<td>9</td>
<td>89.21</td>
<td>8.64</td>
</tr>
</tbody>
</table>

Table 2 states that only some additional costs (up to 10 times) cause the verified solution by our computer assisted proof method.

**References**


Verified Bounds for Singular Values, in Particular for the Spectral Norm of a Matrix and its Inverse

Siegfried Rump

Hamburg University of Technology, Germany
Email: rump@tu-harburg.de

Abstract—The singular value decomposition and spectral norm of a matrix are ubiquitous in numerical analysis. They are extensively used in proofs, but usually it is not necessary to compute them. However, there are some important applications in the realm of verified error bounds for the solution of ordinary and partial differential equations where reasonably tight error bounds for the spectral norm of a matrix are mandatory. We present various approaches to this together with some auxiliary useful estimations.
Accurate Matrix Singular Values

Takeshi Ogita†

†Division of Mathematical Sciences, School of Arts and Sciences
Tokyo Woman’s Christian University

Abstract—In this paper, an algorithm for accurately calculating singular values of matrices is proposed. The proposed algorithm can treat the cases where the matrices are extremely ill-conditioned, i.e. their condition numbers are allowed to go far beyond the bounds of base precision such as IEEE standard 754 double precision. The algorithm requires standard numerical algorithms, which are commonly implemented in several numerical libraries such as BLAS and LAPACK, and an algorithm for accurate matrix multiplication. Numerical results are presented for illustrating the performance of the proposed algorithm.

1. Introduction

This paper is concerned with accurate singular value decomposition (SVD) of an extremely ill-conditioned matrix: Let $A \in \mathbb{R}^{n \times n}$. Then the SVD of $A$ is expressed by

$$A = U \Sigma V^T,$$

where both $U$ and $V$ are orthogonal matrices consisting of all singular vectors, and $\Sigma$ is a diagonal matrix such that

$$\Sigma = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_n).$$

Here $\sigma_i$ for $i = 1, 2, \ldots, n$ are singular values of $A$.

We regard $A$ as ill-conditioned, if condition number of $A$ is very large. Here the condition number of $A$ is defined by

$$\kappa(A) = \|A\|_2 \cdot \|A^{-1}\|_2 = \frac{\sigma_{\text{max}}}{\sigma_{\text{min}}},$$

where $\sigma_{\text{max}} := \max_{1 \leq i \leq n} \sigma_i$ and $\sigma_{\text{min}} := \min_{1 \leq i \leq n} \sigma_i$.

Let $\mathbb{F}$ be a set of floating-point numbers. Let $u$ denote the unit round-off of floating-point arithmetic. In IEEE 754 double precision (binary64), $u = 2^{-53} \approx 1.1 \times 10^{-16}$. In general, if $\kappa(A)$ is large, then errors of results obtained by floating-point arithmetic for linear systems, eigenvalue problems and singular value problems also become large. For example, let us consider a linear system $Ax = b$ with $A$ being nonsingular. Let $x^* = A^{-1}b$, which is the exact solution of $Ax = b$. Let $\tilde{x}$ denote an approximate solution of $Ax = b$ by a standard numerical algorithm such as Gaussian elimination. Then it holds that [5]

$$\frac{\|x^* - \tilde{x}\|}{\|x^*\|} \approx O(u)\kappa(A), \quad 1 \leq i \leq n.$$
10, 11, 13, 14], that are based on error-free transformation of floating-point arithmetic.

We extend it to matrix multiplication. We use the following notation
\[ C_\ell = [A \cdot B]_\ell, \]
which means for \( A \in \mathbb{R}^{m \times p} \) and \( B \in \mathbb{R}^{p \times n} \) it holds that
\[ |AB - C|_\ell = O(u^\ell)\|AB\| + O(u^{2\ell})\|A\|\|B\|. \]
Here \( |A| \) denotes the matrix of taking the absolute value of \( A \) componentwise. Moreover, inequality for matrices such as \( A \leq B \) means \( A_{ij} \leq B_{ij} \) for all \( (i, j) \). If \( \ell = 1 \), then we abbreviate \( C = [A \cdot B]_1 \).

3. Algorithm for accurate SVD

Several algorithms for SVD [1, 2, 3, 6] have been proposed. Most of the SVD algorithms are backward stable; Let \( \tilde{\sigma}_i \) denote approximations of \( \sigma_i \). Then it holds that
\[ \frac{|\sigma_i - \tilde{\sigma}_i|}{\sigma_i} \approx O(u), \quad 1 \leq i \leq n. \tag{2} \]
It is equivalent to
\[ \frac{|\sigma_i - \tilde{\sigma}_i|}{\sigma_n} \approx O(u\kappa(A)), \quad 1 \leq i \leq n. \]
Thus, if \( \kappa(A) = \sigma_1/\sigma_n \) is large, then the accuracy of relatively small singular values become worse.

We present the following algorithm for accurate SVD. When we input \( A \in \mathbb{R}^{m \times n} \) and a tolerance \( \varepsilon_{tol} > u \), the algorithm outputs \( \tilde{\sigma}_i \in \mathbb{R} \), \( 1 \leq i \leq n \) satisfying
\[ \frac{|\sigma_i - \tilde{\sigma}_i|}{\sigma_i} \leq \varepsilon_{tol}, \quad 1 \leq i \leq n. \]

**Algorithm 1** Accurate singular value decomposition.

- **0:** Input \( A \in \mathbb{R}^{m \times n} \) and tolerance \( \varepsilon_{tol} > u \).
- **1:** Put \( U_0 = V_0 = I \) and \( k = 1 \).
- **2:** \( T \leftarrow \{U_{k-1} \cdot A_k\} \).
- **3:** \( B_k \leftarrow -T \cdot V_{k-1} \).
- **4:** \( \tilde{\sigma}_i = (B_k)_{ii}, \quad g_i = \sum_{j=1}^{m} |B_k_{ij}| \) for all \( i \).
- **5:** If \( \varepsilon_{tol} : \tilde{\sigma}_i \geq g_i \) for all \( i \), then \( U = U_k, \quad V = V_k, \quad \Sigma = diag(\tilde{\sigma}_i) \) and stop.
- **6:** \( \Sigma \) of \( B_k \); \( B_k \approx W_k\Sigma_k V_k^T \).
- **7:** \( U_k \leftarrow \{U_{k-1} \cdot W_k\} \).
- **8:** Update \( k \leftarrow k + 1 \) and return to **2.**

Higher precision arithmetic is needed only in Steps 2 and 7.

For the SVD algorithm in Step 6, it is necessary to satisfy Eq. (2).

4. Algorithm for accurate eigenvalue decomposition

Let \( A = A^T \in \mathbb{R}^{n \times n} \). Let \( \sigma_i \) and \( \lambda_i, \quad i = 1, 2, \ldots, n \) be singular values and eigenvalues of \( A \), respectively. Suppose
\[ |\lambda_1| \geq |\lambda_2| \geq \ldots \geq |\lambda_n| > 0. \]

Then it holds that
\[ \sigma_i = |\lambda_i|, \quad i = 1, 2, \ldots, n. \]
For the singular value decomposition \( A = U\Sigma V^T \) and the eigenvalue decomposition \( A = XDX^T \), there exists a diagonal matrix \( S \in \mathbb{R}^{\times n} \) satisfying
\[ \Sigma = DS, \quad |S_{ii}| = 1, \quad i = 1, 2, \ldots, n. \]
Here eigenvectors and singular vectors of \( A \) are equivalent because \( A \) is symmetric, so that we can put \( X = U \). Then, we have
\[ A = U\Sigma V^T = UDSV^T. \]
If \( D \) is nonsingular, then \( S V^T = X^T = U^T \), which yields
\[ S = U^T V. \]
Since \( S \) is diagonal, it is sufficient to compute
\[ S_{ii} = \sum_{k=1}^{n} U_{ki} V_{ki}, \quad i = 1, 2, \ldots, n \]
and set \( \lambda_i = S_{ii}\sigma_i, \quad i = 1, 2, \ldots, n \).

We present the following algorithm for accurate eigenvalue decomposition. When we input \( A = A^T \in \mathbb{R}^{n \times n} \) and a tolerance \( \varepsilon_{tol} > u \), the algorithm outputs \( \lambda_i \in \mathbb{R}, \quad 1 \leq i \leq n \) satisfying
\[ \frac{|\lambda_i - \lambda^*_i|}{\lambda_i} \leq \varepsilon_{tol}, \quad 1 \leq i \leq n. \]

**Algorithm 2** Accurate eigenvalue decomposition for symmetric matrices.

- **0:** Input \( A = A^T \in \mathbb{R}^{n \times n} \) and tolerance \( \varepsilon_{tol} > u \).
- **1:** Compute an accurate SVD of \( A \) s.t.
  \( A \approx U\Sigma V^T \) by Algorithm 1.
- **2:** for \( i = 1 : n \)
  \( s_i = \text{sign} \left( \sum_{k=1}^{n} U_{ki} V_{ki} \right); \)
  \( \lambda_i = s_i \cdot \Sigma_{ii}; \)
end
- **3:** \( X = U, \quad D = \text{diag}(\lambda_i) \) and stop.

In Step 2, we use the sign function sign for enforcing \( s_i = \pm 1 \), because \( U \) and \( V \) are not exactly orthogonal due to the rounding errors, and the computation \( s_i = \sum_{k=1}^{n} U_{ki} V_{ki} \) also involves rounding errors.

5. Numerical results

We evaluate the performance of the proposed algorithm (Algorithm 2), which also includes Algorithm 1. We use a PC with 1.86 GHz Intel Core 2 Duo CPU and Matlab 2009a with INTLAB 5.5 [12]. All computations are done in IEEE 754 double precision arithmetic (\( u = 2^{-53} \approx 10^{-16} \)). To generate test matrices, we adopt the following Matlab function:
function A = randmatsym(n,cnd)
% A real symmetric n-by-n matrix A is
% generated with a specified condition
% number cnd.

d = logspace(1,log10(cnd),n);
D = diag((-1).ˆ[n].*d);
X = randorth(n); % random orthogonal
k = ceil(log(cnd)/log(2ˆ53));
% Accurate computation of X*D*X' with
% k-fold working precision
T = acc_mm(D,X',{0 k});
A = acc_mm(X,T,{0 k});
% Enforce symmetricity of A
if iscell(A)
    for i=1:length(A)
        L = tril(A{i},-1);
        A{i} = L + L' + diag(diag(A{i}));
    end
else
    A = 0.5*(A + A');
end

Using this function, we can generate a random symmetric
matrix A with specified dimension n and condition number
cnd. Moreover, the matrix A has almost one half each of
positive and negative eigenvalues. If cnd is greater than
u−1, we use a cell array to express A as

A = A{1} + A{2} + ... + A{k}

to avoid the reduction of condition number due to the
rounding errors.

First, we treat a matrix with condition number being al-
most the limit of double precision arithmetic. We set n = 5
and cnd = 10^15 as follows:

> n=5; cnd=1e15; A=randmatsym(n,cnd);
> d=sort(eig(sym(A,'f'))) % symbolic comput.
d =
-1000000000000000.175990519298626
-1000000000000000.00553914639556741955996
-10.01000231100684299751196319367
31622.778285161926155448656332726
316227766016827488099294

Here d is obtained by Symbolic Math Toolbox, so that it
has very high accuracy. Thus, we can regard d as the exact
eigenvalues of A.

We compute eigenvalues by a Matlab's built-in function
eig, which calls LAPACK’s DSYEV using a standard nu-
merical algorithm for symmetric eigenvalue problems.

> dl=sort(eig(A)) % Matlab built-in (LAPACK)
dl =
-1.000000000000000e+15

It can be seen from this result that by the standard numer-
ical algorithm, the accuracy of relatively large computed
eigenvalues is high in terms of relative error, while that
of relatively small ones is low. Therefore, if the condi-
tion number becomes larger, then it is estimated that the
standard numerical algorithm cannot compute a good ap-
proximation for relatively small eigenvalues.

We next compute eigenvalues by the proposed algorithm
(Algorithm 2), which is implemented for Matlab as a func-
tion acceig.

> d2=sort(acceig(A)) % proposed algorithm
k = 2
k = 3

It can be seen that by the proposed algorithm, highly ac-
curate results are obtained, even for relatively small eigenval-
ues.

As the second example, we fix n to 100 and vary cnd as
10^20, 10^40, ..., 10^100. Table 1 displays computing time and
the maximum relative error for approximate eigenvalues
obtained by the proposed algorithm. Moreover, the number of iteration k in Algorithm 1, which is called from Algor-
ithm 2, is also shown. We set εtol = 10^-6 as a tolerance for
the proposed algorithm.

Here the proposed algorithm increases the number of it-
ervations adapting to the condition number. It turns out that
highly accurate eigenvalues can efficiently be computed by
the proposed algorithm even for ill-conditioned problems.
Table 1: Results by the proposed algorithm: Computing time $t$, maximum relative error and the number of iterations $k$ ($n = 100$).

<table>
<thead>
<tr>
<th>$\kappa(A)$</th>
<th>$t$ (sec)</th>
<th>Maximum relative error</th>
<th>$k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{40}$</td>
<td>0.24</td>
<td>$2.78 \times 10^{-11}$</td>
<td>3</td>
</tr>
<tr>
<td>$10^{40}$</td>
<td>1.02</td>
<td>$6.22 \times 10^{-13}$</td>
<td>5</td>
</tr>
<tr>
<td>$10^{60}$</td>
<td>1.96</td>
<td>$2.98 \times 10^{-13}$</td>
<td>6</td>
</tr>
<tr>
<td>$10^{80}$</td>
<td>3.51</td>
<td>$5.11 \times 10^{-11}$</td>
<td>7</td>
</tr>
<tr>
<td>$10^{100}$</td>
<td>6.87</td>
<td>$4.17 \times 10^{-13}$</td>
<td>9</td>
</tr>
</tbody>
</table>

6. Conclusion

In this paper, we proposed accurate numerical algorithms for singular value problems and symmetric eigenvalue problems. In general, the condition number of a given matrix is not known in advance, so that we do not know how high computational precision is required. Therefore, it is difficult for a standard numerical algorithm to compute accurate results for all eigenvalues. Using the proposed algorithms, it is possible to do it adaptively increasing the computational precision.

The proposed algorithms consist of standard numerical algorithms, that are implemented in LAPACK, and algorithm for accurate matrix multiplication. Moreover, it is shown in [10, 11] that accurate matrix multiplication can efficiently be implemented using BLAS. Thus, the proposed algorithms have scalability and portability.

To prove the convergence of the proposed algorithms remains to be solved.

References

Condition Numbers of Two-Dimensional Orientation Problem

Katsuhisa Ozaki†, Takeshi Ogita‡ and Shin’ichi Oishi∗∗∗

†Department of Mathematical Sciences, College of Systems Engineering and Science, Shibaura Institute of Technology 307 Fukasaku, Minumaku, Saitama-shi, Saitama 337-8570, Japan
‡Department of Mathematical Sciences, Tokyo Woman’s Christian University 2-6-1 Zempukuji, Suginami-ku, Tokyo 167-8585, Japan
* Faculty of Science and Engineering, Waseda University 3-4-1 Okubo, Shinjyuku-ku, Tokyo 169-8555, Japan
** CREST, Japan Science and Technology Agency

Email: ozaki@sic.shibaura-it.ac.jp

Abstract—There are robustness problems in the field of computational geometry. A correct result is output by rational arithmetic. However, an inexact result is output due to rounding errors by finite precision arithmetic. A condition number is frequently used for discussions of accuracy of computed results in the area of numerical analysis. In this paper, this concept is innovated to one of the basic geometric predicates ‘two-dimensional orientation problem’.

1. Introduction

This paper is concerned with basic computations in computational geometry. We focus our mind on a two-dimensional orientation problem which is frequently used in applied geometric problems, for example, a convex hull for a set of points, a point-in-polygon problem and so on. Assume that an oriented line passes from \( A(a_x, a_y) \) to \( B(b_x, b_y) \). Let \( C(c_x, c_y) \) be a point. Let \( \mathbb{F} \) be the set of floating-point numbers defined by IEEE 754 [1]. Assume that \( A, B, C \in \mathbb{F}^2 \). The problem is to distinguish which the point \( C \) is left to, right to, or on the oriented line \( AB \). The problem can be solved by evaluating the following sign of the determinant:

\[
\text{sign}(\text{det}(G)), \quad G := \begin{pmatrix} a_x & a_y & 1 \\ b_x & b_y & 1 \\ c_x & c_y & 1 \end{pmatrix}.
\]  

(1)

If the sign of the determinant is positive, then the point is left to the oriented line. If the sign of the determinant is negative, then the point is right to the oriented line. Otherwise, the point is on the oriented line.

If the sign of the determinant (1) is evaluated by pure floating-point arithmetic, then an inexact result may be obtained due to accumulation of rounding errors. When the point is very close to the oriented line, heavy cancellation occurs in the floating-point computations of (1), so that finally an incorrect result may be obtained. It yields serious problems. See [2] for detailed examples (these problems are called robustness problems). To overcome this problem, using multiple precision arithmetic is an option.

However, to apply it straightforwardly makes the computational performance slow down due to software simulation. Therefore, adaptive approaches, for example [3, 4, 5], have been investigated. Initially, a so-called floating-point filter is applied. The filter quickly checks a sufficient condition for correctness of the sign of the determinant. If the filter cannot guarantee correctness of the sign of the determinant, then computational precision is increased and we check the correctness again. These procedures are repeated until the correctness of the result can be guaranteed. Therefore, computing time of the adaptive algorithm is related to difficulty of the problem.

Various sets of three points \( A, B \) and \( C \) are necessary so as to the compare performance of adaptive algorithms. We innovate condition numbers to the two-dimensional orientation problem. The condition numbers are frequently used for discussion of the accuracy of the numerical result. There are several ways to compute \( \text{det}(G) \) in (1), for example,

\[
F_1 = (a_x - c_x)(b_y - c_y) - (a_y - c_y)(b_x - c_x), \quad (2)
\]

\[
F_2 = a_x(b_y - c_y) + b_x(c_y - a_y) + c_x(a_y - b_y), \quad (3)
\]

\[
F_3 = a_xb_y - a_xc_y - c_xb_y - a_xb_y + a_yc_x + c_yb_y. \quad (4)
\]

We define condition numbers for each expression. Let them be \( \text{cond}(F_1), \text{cond}(F_2), \text{cond}(F_3) \). We give an algorithm which randomly outputs a set of three points. The resultant set gives almost same condition numbers for \( \text{cond}(F_1) \), \( \text{cond}(F_2) \), \( \text{cond}(F_3) \).

2. Condition Numbers for Orientation Problem

In this section, we innovate condition numbers to the two-dimensional orientation problem. \( \text{fl}(\cdot) \) means that an expression in parentheses is evaluated by pure floating-point arithmetic. First, we define \( \text{cond}(F_1) \). For \( a, b \in \mathbb{F} \), there is an algorithm which transforms \( a + b \) into \( x + y(x = \text{fl}(a + b)) \) without rounding errors when overflow does not occur in \( \text{fl}(a + b) \). Denote this algorithm as

\[
[x, y] = \text{TwoSum}(a, b).
\]

- 144 -
See [8] for the detail of this algorithm. There is an algorithm which transforms \( a \times b \) into \( x + y (x = f(a \times b)) \) without rounding errors when neither overflow nor underflow occurs in the computation of the algorithm. Let this algorithm denote

\[ [x, y] = \text{TwoProduct}(a, b). \]

See [7] for the detail of this algorithm. Applying \text{TwoSum} to each subtraction in (2) as follows:

\[
\begin{align*}
[t_1, e_1] &= \text{TwoSum}(a_x - c_x), \\
t_2, e_2] &= \text{TwoSum}(b_y - c_y), \\
t_3, e_3] &= \text{TwoSum}(a_y - c_y), \\
t_4, e_4] &= \text{TwoSum}(b_y - c_y),
\end{align*}
\]

(5)

(2) can be transformed into

\[
\text{det}(G) = [(t_1 + e_1)(t_2 + e_2) - (t_3 + e_3)(t_4 + e_4)].
\]

(6)

Expanding (6) straightforwardly yields

\[
t_1t_2 + t_1e_2 + t_2e_1 + e_1e_2 - t_3t_4 - t_3e_4 - t_4e_3 - e_3e_4.
\]

(7)

Applying \text{TwoProduct} for each product in (7) as follows:

\[
\begin{align*}
[p_1, p_2] &= \text{TwoProduct}(t_1, t_2), \\
p_3, p_4] &= \text{TwoProduct}(t_3, t_4), \\
p_5, p_6] &= \text{TwoProduct}(t_1, e_2), \\
p_7, p_8] &= \text{TwoProduct}(t_2, e_1), \\
p_9, p_{10}] &= \text{TwoProduct}(t_3, e_4), \\
p_{11}, p_{12}] &= \text{TwoProduct}(t_4, e_3), \\
p_{13}, p_{14}] &= \text{TwoProduct}(e_1, e_2), \\
p_{15}, p_{16}] &= \text{TwoProduct}(e_3, e_4),
\end{align*}
\]

we finally obtain

\[
\text{det}(G) = \sum_{i=1}^{16} p_i.
\]

(8)

This way is used in robust algorithms [4, 5]. We now introduce the condition number of summation of floating-point numbers. Let \( v \in \mathbb{F}^n \), the condition number of summation is introduced in [6] as follows:

\[
\text{cond}(\sum_{i=1}^{n} v_i) = \frac{\sum_{i=1}^{n} |v_i|}{|\sum_{i=1}^{n} v_i|}.
\]

Therefore, it is natural that the condition number for (2) is defined as

\[
\text{cond}(F_1) = \frac{\sum_{i=1}^{16} |p_i|}{|\sum_{i=1}^{16} p_i|}.
\]

(9)

Next, we define \text{cond}(F_2). Applying \text{TwoSum} for the each subtraction in (3), we obtain

\[
\begin{align*}
[t_5, e_5] &= \text{TwoSum}(b_y - c_y), \\
t_6, e_6] &= \text{TwoSum}(c_y - a_y), \\
t_7, e_7] &= \text{TwoSum}(a_y - b_y).
\end{align*}
\]

After that, we have

\[
\begin{align*}
\alpha_i(t_5 + e_5) + b_i(t_6 + e_6) + c_i(t_7 + e_7) \\
= \alpha_i t_5 + \alpha_i e_5 + b_i t_6 + b_i e_6 + c_i t_7 + c_i e_7.
\end{align*}
\]

(10)

TwoProduct can be applied for each product in the above-mentioned expression as follows:

\[
\begin{align*}
[q_1, q_2] &= \text{TwoProduct}(\alpha_i, t_5), \\
[q_3, q_4] &= \text{TwoProduct}(\alpha_i, e_5), \\
[q_5, q_6] &= \text{TwoProduct}(b_i, t_6), \\
[q_7, q_8] &= \text{TwoProduct}(b_i, e_6), \\
[q_9, q_{10}] &= \text{TwoProduct}(c_i, r_i), \\
[q_{11}, q_{12}] &= \text{TwoProduct}(c_i, e_7).
\end{align*}
\]

Then, we have

\[
\text{det}(G) = \sum_{i=1}^{12} q_i.
\]

(11)

Therefore, the condition number for (3) is defined by

\[
\text{cond}(F_2) = \frac{\sum_{i=1}^{12} |q_i|}{\sum_{i=1}^{12} |q_i|}.
\]

(12)

From similar discussion, \text{cond}(F_3) can be defined. Applying \text{TwoProduct} for each product in (4) as follows:

\[
\begin{align*}
[r_1, r_2] &= \text{TwoProduct}(a_y, b_y), \\
r_3, r_4] &= \text{TwoProduct}(a_y, c_y), \\
r_5, r_6] &= \text{TwoProduct}(b_y, x_y), \\
r_7, r_8] &= \text{TwoProduct}(a_y, -c_y), \\
r_9, r_{10}] &= \text{TwoProduct}(a_y, b_y), \\
r_{11}, r_{12}] &= \text{TwoProduct}(b_y, c_y),
\end{align*}
\]

the condition number for (4) is defined by

\[
\text{cond}(F_3) = \frac{\sum_{i=1}^{12} |r_i|}{\sum_{i=1}^{12} |r_i|}.
\]

(13)

This form is used in Shewchuk’s algorithm ‘Orient2dExact’.

\textbf{Remark 1} There is a set of three points which yields big difference of the condition numbers. For example, let three points be

\[
A = (2, 1), \quad B = (-1, -1), \quad C = (2^{100}, 2^{100}).
\]

(14)

Then, \text{cond}(F_1) = 2.53e + 30, \text{cond}(F_2) = 5, \text{cond}(F_3) = 5. If one library uses the form (2) and the other library uses the form (3), comparing the performance of two libraries by (9) yields an unfair result.

It is possible to give different orders of the expression (2), for example,

\[
F'_1 = (c_y - b_y)(a_y - b_y) - (c_y - b_y)(a_y - b_y),
\]

(15)

\[
F''_1 = (b_y - a_y)(c_y - a_y) - (b_y - a_y)(c_y - a_y).
\]

(16)

Condition numbers for \( F'_1 \) and \( F''_1 \) can be defined from similar discussion. Let them be \text{cond}(F'_1) and \text{cond}(F''_1).
respectively. There is a case that the condition numbers are much different. For example, recalling (9), the condition numbers are \( \text{cond}(F_1) = 2.53e + 30 \), \( \text{cond}(F_1') = 5 \), \( \text{cond}(F_1'') = 5 \).

We can derive the following approximation of the condition numbers

\[
\text{cond}(F_1) \approx \frac{|(a_x - c_y)(b_y - c_y) + |(b_x - c_y)(a_x - c_y)|}{|a_x(c_y - b_y) + b_x(a_y - c_y) + c_y(a_y - b_y)|},
\]

\[
\text{cond}(F_2) \approx \frac{|a_x(b_y - c_y) + b_x(a_x - c_y) + c_x(a_y - b_y)|}{|a_x(b_y - c_y) + b_x(a_x - c_y) + c_y(a_y - b_y)|},
\]

\[
\text{cond}(F_3) \approx \frac{|a_xb_x - a_xc_y - c_xb_y + a_yb_x + a_yc_x - c_yb_x|}{|a_xb_y + a_yb_x|}.
\]

3. How to Generate Set

Let \( X \) be a required condition number. We propose an algorithm which randomly outputs a set of three points. Moreover, the set gives almost same condition numbers: \( \text{cond}(F_1), \text{cond}(F_2) \) and \( \text{cond}(F_3) \). Our algorithm consists on the following three parts.

3.1. Condition Number up to 1e+16

If a required condition number is less than \( 10^{16} \), an algorithm developed in this subsection is used. First, we generate \( a_x, a_y, b_x, b_y, c_x \) at random. Let \( S \) be

\[
S = fl(|a_xb_x| + |a_xc_x| + |b_xb_y| + |b_xc_x|).
\]

A vector \( w \) is generated as follows:

\[
[w_1, w_2] = \text{TwoProduct}(a_x, b_x), \\
[w_3, w_4] = \text{TwoProduct}(b_x, a_y), \\
[w_5, w_6] = \text{TwoProduct}(c_x, a_y), \\
[w_7, w_8] = \text{TwoProduct}(b_y, -c_x).
\]

Here, we introduce the accurate summation algorithm by Rump, Ogita and Oishi [11]. It is guaranteed that accuracy of the result by their algorithm is within 1 ulp (it is called faithful rounding). This function denotes \( \text{AccSum}(p) \) for a vector \( p \in \mathbb{F}^n \) in INTLAB [9]. Let \( d \) be a sum of all elements in \( w \). After that, \( c_y \) is obtained as follows:

\[
d = \text{AccSum}(w), \quad c_y = fl\left(\frac{S/X - d}{b_x - a_x}\right).
\]

3.2. Condition Number up to 1e+32

We generate \( a_x, a_y, b_x, b_y \) at random. Let \( S \) be

\[
S = fl(|a_xb_x| + |a_xb_y|).
\]

The following vector \( w \) is generated as follows:

\[
[w_1, w_2] = \text{TwoProduct}(a_x, b_x), \\
[w_3, w_4] = \text{TwoProduct}(b_x, a_y).
\]

Let \( d \) be a sum of all elements in \( w \). Then \( c_x \) is obtained by

\[
d = \text{AccSum}(w), \quad c_x = fl\left(\frac{S/X - d}{a_x - b_y}\right).
\]

Next, we add four elements to the vector \( w \) as follows:

\[
[w_5, w_6] = \text{TwoProduct}(c_x, -b_y), \\
[w_7, w_8] = \text{TwoProduct}(c_x, a_y).
\]

After that, \( d \) is updated and \( c_y \) is obtained as follows:

\[
d = \text{AccSum}(w), \quad c_y = fl\left(\frac{S/X - d}{b_x - a_x}\right).
\]

3.3. Condition Number over 1e+32

If \( a_x, a_y, b_x \) and \( b_y \) are given at random, then it is almost impossible to get a set of points which gives condition numbers up to \( 10^{32} \). First, we apply an ill-conditional matrix generator by Rump [10].

\[
A = \text{randmat}(2, X/1e + 32);
\]

This function is supported by INTLAB [9]. Then, let \( a_x, a_y, b_x \) and \( b_y \) be

\[
A = \begin{pmatrix} a_x & a_y \\ b_x & b_y \end{pmatrix}.
\]

Next, we generate a vector \( w \) as

\[
[w_1, w_2] = \text{TwoProduct}(a_x, b_x), \\
[w_3, w_4] = \text{TwoProduct}(b_x, -a_x).
\]

Let \( d \) and \( c_y \) be defined as follows:

\[
d = \text{AccSum}(w), \quad c_y = fl\left(\frac{-d}{a_y - b_y}\right).
\]

We extend a vector by adding four terms as follows:

\[
[w_5, w_6] = \text{TwoProduct}(c_x, -b_y), \\
[w_7, w_8] = \text{TwoProduct}(c_x, a_y).
\]

Then, \( d \) is updated and \( c_y \) are computed by

\[
d = \text{AccSum}(w), \quad c_y = fl\left(\frac{S/X - d}{b_x - a_x}\right).
\]

This algorithm able to output a set of three points which gives the condition numbers up to \( \text{cond} = 10^{64} \).

4. Numerical Examples

We show numerical examples by our algorithm. We used MATLAB’s built-in function \text{randn} for generating points [12]. In Table 1, the item ‘cnd’ shows a required condition number. The items ‘\text{cond}(F_1)’, ‘\text{cond}(F_2)’ and ‘\text{cond}(F_3)’ show the condition numbers defined in Section 2. The results are averages of 100 examples. It is confirmed from Table 1 that our algorithm can output a set of points

\[
\text{cond} \leq 10^{64}.
\]
Table 1: Comparison of the condition numbers.

<table>
<thead>
<tr>
<th>cnd</th>
<th>cond(F₁)</th>
<th>cond(F₂)</th>
<th>cond(F₃)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10⁰</td>
<td>2.39e+05</td>
<td>1.62e+05</td>
<td>2.47e+05</td>
</tr>
<tr>
<td>10¹</td>
<td>3.07e+10</td>
<td>2.22e+10</td>
<td>2.93e+10</td>
</tr>
<tr>
<td>10²</td>
<td>2.47e+15</td>
<td>1.66e+15</td>
<td>2.49e+15</td>
</tr>
<tr>
<td>10³</td>
<td>1.75e+20</td>
<td>1.76e+20</td>
<td>3.05e+20</td>
</tr>
<tr>
<td>10⁴</td>
<td>1.71e+25</td>
<td>1.78e+25</td>
<td>2.91e+25</td>
</tr>
<tr>
<td>10⁵</td>
<td>1.51e+30</td>
<td>1.76e+30</td>
<td>2.78e+30</td>
</tr>
<tr>
<td>10⁶</td>
<td>8.82e+36</td>
<td>8.73e+36</td>
<td>8.82e+36</td>
</tr>
<tr>
<td>10⁷</td>
<td>4.99e+40</td>
<td>4.99e+40</td>
<td>4.99e+40</td>
</tr>
<tr>
<td>10⁸</td>
<td>1.35e+46</td>
<td>1.35e+46</td>
<td>1.35e+46</td>
</tr>
<tr>
<td>10⁹</td>
<td>3.51e+50</td>
<td>3.51e+50</td>
<td>3.51e+50</td>
</tr>
</tbody>
</table>

Table 2: Difference of condition numbers.

<table>
<thead>
<tr>
<th>cnd</th>
<th>max</th>
<th>mean</th>
<th>min</th>
</tr>
</thead>
<tbody>
<tr>
<td>10⁰</td>
<td>11.5</td>
<td>3.03</td>
<td>1.01</td>
</tr>
<tr>
<td>10¹</td>
<td>24.5</td>
<td>3.01</td>
<td>1.04</td>
</tr>
<tr>
<td>10²</td>
<td>9.68</td>
<td>2.70</td>
<td>1.00</td>
</tr>
<tr>
<td>10³</td>
<td>30.0</td>
<td>3.91</td>
<td>1.03</td>
</tr>
<tr>
<td>10⁴</td>
<td>75.4</td>
<td>5.29</td>
<td>1.00</td>
</tr>
<tr>
<td>10⁵</td>
<td>43.7</td>
<td>9.63</td>
<td>1.00</td>
</tr>
<tr>
<td>10⁶</td>
<td>1.91</td>
<td>1.05</td>
<td>1.00</td>
</tr>
<tr>
<td>10⁷</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>10⁸</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
</tbody>
</table>

with almost required condition number for each \( \text{cond}(F₁) \), \( \text{cond}(F₂) \) and \( \text{cond}(F₃) \). Next, we check the difference among \( \text{cond}(F₁) \), \( \text{cond}(F₂) \) and \( \text{cond}(F₃) \). Table 2 shows the maximum, mean and minimum of

\[
\frac{\max(\text{cond}(F₁), \text{cond}(F₂), \text{cond}(F₃))}{\min(\text{cond}(F₁), \text{cond}(F₂), \text{cond}(F₃))}
\]

for 100 examples. From Table 2, the difference among condition numbers of \( F₁ \), \( F₂ \) and \( F₃ \) is not a lot.

Next, we checked difference of \( \text{cond}(F₁) \), \( \text{cond}(F′₁) \) and \( \text{cond}(F''₁) \). Table 2 shows the maximum, mean and minimum of

\[
\frac{\max(\text{cond}(F₁), \text{cond}(F′₁), \text{cond}(F''₁))}{\min(\text{cond}(F₁), \text{cond}(F′₁), \text{cond}(F''₁))}
\]

for 100 examples. It is confirmed from Table 3 that there is not so much difference among condition numbers except \( \text{cnd} = 10⁵⁰ \). When \( \text{cnd} = 10⁵⁰ \), we found that \( b₁ \) and \( b₂ \) are relatively large. Then \( \text{cond}(F′₁) \) is approximately \(|b₁b₂|\). There is no \( b₁b₂ \) in \( F₁ \) and \( F₁′′ \). Therefore, it yields big difference.

Table 3: Difference of condition numbers.

<table>
<thead>
<tr>
<th>cnd</th>
<th>max</th>
<th>mean</th>
<th>min</th>
</tr>
</thead>
<tbody>
<tr>
<td>10⁰</td>
<td>7.83</td>
<td>3.15</td>
<td>2.03</td>
</tr>
<tr>
<td>10¹</td>
<td>9.09</td>
<td>3.23</td>
<td>2.01</td>
</tr>
<tr>
<td>10²</td>
<td>11.5</td>
<td>3.32</td>
<td>2.00</td>
</tr>
<tr>
<td>10³</td>
<td>12.0</td>
<td>3.57</td>
<td>2.01</td>
</tr>
<tr>
<td>10⁴</td>
<td>23.1</td>
<td>3.39</td>
<td>2.01</td>
</tr>
<tr>
<td>10⁵</td>
<td>17.8</td>
<td>3.30</td>
<td>2.00</td>
</tr>
<tr>
<td>10⁶</td>
<td>3.74</td>
<td>2.32</td>
<td>2.00</td>
</tr>
<tr>
<td>10⁷</td>
<td>3.82</td>
<td>2.39</td>
<td>2.00</td>
</tr>
<tr>
<td>10⁸</td>
<td>3.48</td>
<td>2.42</td>
<td>2.00</td>
</tr>
<tr>
<td>10⁹</td>
<td>161</td>
<td>112</td>
<td>77.0</td>
</tr>
</tbody>
</table>

References


GPGPU Accelerated Scene Segmentation Using Nonparametric Clustering

Balázs Varga and Kristóf Karacs

Dept. of Information Technology, Pázmány Péter Catholic University
Práter str. 50/a, H-1083, Budapest, Hungary
E-mail: varga.balazs@itk.ppke.hu, karacs@itk.ppke.hu

Abstract—This paper presents the parallel implementation of a nonparametric image segmentation method: the mean shift algorithm using joint spatial-range feature space. By considering spatial information, the mean shift can distinguish topographically differing objects in the scene, but this feature costs additional computational demand through increased number of kernel functions. The proposed algorithm runs the mode-defining kernel iterations parallel by utilizing the many-core graphics processing unit (GPGPU). We use our own voting procedure for pixel-cluster assignment. Numerical evaluation showed that our solution efficiently speeds up the image clusterization procedure.

1. Introduction

Scene segmentation is one of the most common, yet most versatile tasks of image processing. Among today’s most advanced segmentation approaches, such as graph cuts [1][2], normalized cuts [3], or various types of k-means [4][5], the mean shift segmentation is one of the most studied and applied nonlinear techniques. Basically having no a-priori knowledge demand, it can dynamically set the number of segmented clusters through a nonparametric framework. Our solution gives a notable speed-up to the mean shift method, by parallelizing its internal structure, and instead of running it on a CPU unit, we use a many-core programmable general-purpose graphics programming unit (GPGPU) [6]. For now, we did not aim to overcome the quality of the original algorithm implemented on the CPU, rather to show that our GPGPU-based implementation can achieve similar segmentation accuracy with considerably lower time demand.

The paper is organized as follows: in the first part of the second section we give an overview of the generic, joint feature space mean shift paradigm. The second part of the section briefly summarizes related works about possible speed-up strategies; while the third part describes the disadvantage behind the usage of the spatial domain. The third section explains our solution of speeding up this task. Section four presents the evaluation framework, which is followed by the numerical results in section five. The paper is closed with a short summary in the sixth section.

2. Algorithmic background

After the mean shift procedure was introduced by Fukunaga and Hostetler [7] in 1975, it was Cheng [8] who pointed out 20 years later that the mode seeking process of the algorithm is a parallel hill climbing method, applying the clusterization algorithm in the Hough space. Following another half a dozen of years of smoldering, Meer and Comaniciu gave an extensive overview [9] of the segmentation framework, using it for image segmentation and discontinuity preserving smoothing. Their approach concerning color images is briefly summarized in subsection 2.1.

2.1 Mean shift in the joint feature space

The mean shift procedure considers its feature space as an empirical probability density function. A local maximum of this function (namely, a region over which it is highly populated) is called a mode. Mode calculation is formulated as an iterative scheme of mean calculation, which takes a certain number of feature points and calculates their mean value by using a weight kernel function. Meer and Comaniciu used a composite feature space consisting of both topographical (spatial) and color (range) information of the image. As a result, each feature point in this space is represented by an \( \chi = (x, y, c_R, c_G, c_B) \) vector which consists of the corresponding pixel’s \( x, y \) 2D position in the spatial lattice, and its \( c_R, c_G, c_B \) 3D color value in the applied color space (e.g., the \( x_c=(YChCr) \) coordinates). The iterative scheme for the calculation of a mode is as follows: let \( \chi_i \) and \( \chi_{i+1} \) be the 5D input and output points in the joint feature space for all \( i=1, ..., n \), \( n \) being the number of pixels in color image \( I \).

Then for each \( i \):
1. Initialize \( k = 1 \)
2. Compute the iterative formula

\[
\chi^{k+1}_i = \frac{\sum_{j=1}^{n} \chi_j \mathbf{g} \left( \frac{x_{ij} - x_{ij}^k}{h_i} \right) \left( \frac{x_{ij} - x_{ij}^k}{h_i} \right)^k}{\sum_{j=1}^{n} \mathbf{g} \left( \frac{x_{ij} - x_{ij}^k}{h_i} \right) \left( \frac{x_{ij} - x_{ij}^k}{h_i} \right)^k}
\]
until the mean shift vector \( \| \chi_k^{i+1} - \chi_i^k \| \) falls under a
given threshold, where \( g \) denotes the Gaussian kernel
function, with \( h_1 \) and \( h_r \) being the spatial- and range
bandwidth parameters respectively.

3. Allocate \( z_i = \chi_i^{k+1} \); that is, output value \( z_i \) is given by
feature point \( \chi_i \) after the final \((k+1)^{th}\) step.

Those \( z_i \) points, which are adequately close to each
other, are concatenated resulting discrete, non-overlapping
clusterization of the input image. All pixels in the cluster
inherit the color of its mode.

The main advantage of applying the joint feature space
is that the algorithm became capable of discriminating
scene objects based on their color and position; making
mean shift a multi-purpose, nonlinear, nonparametric tool
for image segmentation.

On the other hand the disadvantage of the algorithm, as
it was specified earlier by Cheng is its high computational
complexity of \( O(n^2) \).

2.2 Acceleration strategies

Comaniciu and Meer introduced an efficient technique
called the coarse grid [10] in order to highly reduce complexity. Briefly, they perform random tessellation of the feature space with \( m < n \) kernels, execute mean shift segmentation (resulting \( z_i, i=[1,m] \) modes), merge close modes (as in [9]), then assign each \( \chi_i \) feature point to the
closest \( z_j \) cluster-defining mode. They showed that this
approach is capable of producing technically equal segmentation quality with a computation demand of
\( O(m^2 n) \ll O(n^2) \).

Ever since, several alternative techniques have been
proposed to achieve speed-ups, e.g. through space
discretization and downsampling [11], local subsets [12],
expectation-maximization [11][13], hierarchical solutions
[14][15], and the Newton iteration method [16][17].
Although our current system does not use these alternative
techniques, later on most of them can easily be added to
our framework enhancing its speed and reducing its time
demand.

2.3 Over-segmentation: the advantage’s tradeoff

The clear advantage brought by the usage of the spatial
domain is the topographical discriminative potential: objects with similar or even the same color can be
distinguished, if they are topographically distinct. On the
other hand spatial discrimination of two discrete objects
requires the usage of two kernels. Therefore in the case of
very detailed images, spatial filtering involves a
computational tradeoff: proper coverage of the feature space necessitates the usage of numerous kernels.

3. Our approach

As a result, both the original and the improved
algorithms follow the bottom-up strategy, when the output
is a result of over-segmentation, which is followed (off-
line), or accompanied (on-line) by a cluster merging
procedure.

We considered the off-line mean shift algorithm, which is
divided into two main subtasks: the mode calculation
and the cluster merging procedure. Mode calculation
algorithm is a highly data parallel [18] task: the same
iterative procedure is performed on the elements of the
feature space with each kernel having a different seed point.
Nowadays many-core GPGPUs are publicly available at a reasonable price, having more than a
hundred distinct stream processors, which can effectively
handle data-parallelism-related tasks. We implemented the
mode seeking task on a GPGPU, and compared its
performance with the CPU implementation of the
procedure.

3.1 GPGPU implemented mode calculation

Our algorithm applies Comaniciu and Meer’s coarse
grid technology [10] on the joint feature space. But instead
of running the iteration specified by eq. 1 on a single
kernel until saturation, we extend this computational framework by running the iterative algorithm simultaneously on several mean shift kernels, which we
call multiple simultaneous mode seeking (MSMS).

The MSMS begins by selecting \( m \) initial mean points
randomly in the joint feature space. The mean shift
iteration is then started from these seed points by using
Gaussian kernel functions having a common \((h_r,h_s)\) spatial-
range bandwidth parameterization for each kernel. The
procedure is terminated, when the length of each kernel’s
mean shift vector becomes smaller than a pre-defined
threshold value.

In order to properly assign all feature points to the
corresponding mode, we constructed a voting system. In
every iteration of mode seeking, for each kernel we
compute pixel-wisely the following cumulative confidence
value (CCV):

\[
C_{i,j}^{k+1} = C_{i,j}^k + g \left( \frac{x_{ij} - x^k_j}{h_r} \right)^2 g \left( \frac{x_{ij} - x^k_i}{h_s} \right)^2 \quad (2)
\]

with

\[
C_{i,j}^0 = 0 \quad (3)
\]

where \( C_{i,j} \) denotes the confidence value of pixel \( i \) at the
\( k^{th} \) iteration for kernel \( j \). Note that the calculation of the
CCV does not require additional computation, as it is a
part of the mean shift iteration’s numerator. Let \( C_{i,j} \) denote
the final CCV computed in the last iteration and let \( CID_i \)
stand for the \( i^{th} \) pixel’s cluster ID. After the modes are
retrieved and every \( C_{i,j} \) has been obtained, each image
pixel gets associated with a mode using the following rule:

\[ CID_i = \arg \max_j (C_{i,j}) \]  \hspace{1cm} (4)

### 3.2 Cluster merging procedure

The number and structure of final clusters are constructed with cluster merging, which currently runs on the CPU. Cluster \( i \) and \( j \) are joined, if they satisfy the criteria:

- **C1.** The two clusters have a common border in terms of the eight-neighbor connectivity.
- **C2.** \( \| x_{i,j} - x_{r,j} \| < h_j \).

In this case the position of the mode is recalculated. Let \( NP_i \) and \( NP_j \) denote the number of pixels in clusters \( i \) and \( j \) respectively, and let us say that both \( C1 \) and \( C2 \) criteria holds and the two are merged into cluster \( k \). Then the color information carried by mode \( z_k \) of the newly formed cluster is

\[ z_k = \frac{NP_i \cdot z_i + NP_j \cdot z_j}{NP_i + NP_j} \]  \hspace{1cm} (5)

i.e. it is a weighted average of the conjoined duet. The procedure runs iteratively until no classes can be merged.

### 4. Evaluation framework

To provide comparative results to the proposed GPGPU algorithm, the coarse grid CPU-optimized mean shift procedure was implemented with as few differences from the GPGPU version as possible. The main difference is that while the GPGPU runs the MSMS version, the CPU does the mode calculation one by one (single simultaneous mode seeking, SSMS).

Image segmentation is an ill-posed task [19], but several publicly available image corpora exist to make algorithms comparable. For evaluation purposes, we selected 50 color images from the Berkeley Segmentation Dataset (BSDS), [20] for which multiple human-made segmentation maps are provided as reference. Let the name best parameter pair (BPP) denote a pair of \((h_v, h_i)\) kernel bandwidth parameters that result in a closest-to-the-reference segmentation for a given image of our evaluation image set.

In order to determine such pairs, the CPU algorithm was utilized the following way: 64 alternative segmentations were made for each evaluation image using \( 8 \times 8 \) different \((h_v, h_i)\) bandwidth values for each of the 50 images. During the process, the algorithm computed and logged the number of clusters before and after the merging procedure, and the elapsed time of the mode seeking. Then, depending on the number of provided BSDS references 1 to 4 BPPs were selected for each image, resulting a total of 117 pairs. Next, the GPGPU algorithm was run on the evaluation set using the corresponding BPPs, and finally the elapsed time of the mode seeking was compared to the CPU algorithm. It is worth to note that the GPGPU runtime includes the time-demanding CPU to GPGPU and GPGPU to CPU data transfers. Furthermore, since the number of mean shift iterations depends on the (randomly selected) initial kernel position, we ran both algorithms multiple times with the same parameterization, and compared the average running times. Moreover cluster merging was done using the same CPU-based algorithm; therefore time consumption of the mode merging procedure was not part of the comparison.

All calculations were run on a single PC equipped with 2GB of RAM and an Intel E6400 CPU running at 2.13GHz. The GPGPU was an nVidia G92 GPU operating with 112 stream processors and 1024MB of video RAM.

### 5. Results

On average, the GPGPU was able to segment the scene 2.997 times faster under the same circumstances. Fig. 1/a displays an example image from the BSDS among with a segmentation made by human observer on 1/d, and our segmentation results made on the different computational platforms on 1/b and 1/c.

### 6. Conclusion

We successfully implemented the nonparametric mean shift clustering algorithm using the joint spatial-range feature space onto the many-core GPGPU, for which we used our own voting procedure for mode selection. The GPGPU algorithm proved to run almost three times faster than its CPU variant. Later on we plan to enhance the system using the acceleration strategies described in Section 2.2.

### References

Figure 1. An example of segmentation made on the different computation platforms. Segmented images are result of mode seeking followed by cluster merge. $(h_s, h_r) = (0.08, 0.01)$, $m = 54$ clusters were merged into 4 (CPU case) and 5 (GPGPU case).


Architecture of The Next Generation Real Time CNN Processor: RTCNNP-v2

Evren Cesur†, Nerhun Yildiz‡ and Vedat Tavsanoglu†

†Dept. of Electronics and Communications Engineering, Yildiz Technical University
Barbaros Bulvari, 34349, Besiktas, Istanbul, Turkey
Email: nerhuny@yildiz.edu.tr, ecesur@yildiz.edu.tr, tavsanav@yildiz.edu.tr

Abstract—This paper is the continuation of our previous work reported in [1], where the local control structure of the Second–Generation Real–Time Cellular Neural Network Processor (RTCNNP-v2) was covered. The system is primarily designed for high resolution, high speed real–time video image processing. In this paper the block diagram of the new processor and an improvement over the local control structure are presented. The proposed structures are coded in VHDL and realized on an FPGA.

1. Introduction

Standard Cellular Neural Network (CNN) architecture is a two dimensional array of parallel processors [2]. In analog implementations the processing elements are neural cells which are organized as two dimensional spatial grids with temporal analog memory nodes. Spatial interconnections of the neighboring neurons define the spatio-temporal dynamics of the grid. The output is taken from a non-linear activation function which vary for different applications [3].

While the analog architecture has many benefits like high speed and low power dissipation, digital emulation methods are becoming more feasible as the digital technology tends to be faster and cheaper. Different CNN emulation infrastructures are proposed in the literature such as FPGAs, ASICs, DSPs, GPUs, etc.. Among the alternatives the FPGAs are preferred for our implementation, as their speed is comparable with ASICs, regularity is similar to the GPUs, and they are highly parallel, reconfigurable and reusable [4], [5].

This paper is the continuation of [1] where the first design specifications of the Second–Generation Real–Time Cellular Neural Network Processor RTCNNP-v2 are proposed. The general structure of the design is based on the first–generation processor, RTCNNP-v1 [6]. The system is designed as a real–time, high speed, high resolution video processing array.

First a brief mathematical overview of the CNN model used in both RTCNNP versions is given. Then the architecture and the control structure of the processor array is presented. Finally the FPGA realization results are discussed.

2. Mathematical Overview

Mathematical model of the CNN depends on the number of spatial dimensions, activation function, number of layers, etc.. Although the proposed architecture is capable of processing many variations of the CNN with some modifications and adjustments, the simpler two dimensional, single layer, space invariant CNN model using saturated linear activation function is used for simplicity [3].

2.1. Continous–Time CNN Model

The mathematical model of the continuous–time CNN is defined with the state equation

\[
\frac{dx_{ij}(t)}{dt} = -x_{ij}(t) + A \odot Y_{ij}(t) + B \odot U_{ij} + z_{ij}, \quad (1)
\]

where

\[
y_{ij}(t) = f(x_{ij}(t)) = 0.5 \left( \left| x_{ij}(t) + 1 \right| - \left| x_{ij}(t) - 1 \right| \right)
\]

and

\[
x_{ij}, y_{ij}, u_{ij}, z_{ij}, n \text{ are the variable indices of the } i^{\text{th}}, j^{\text{th}} \text{ cell,}
\]

\[
A \odot Y_{ij}, B \odot U_{ij} \text{ are the feed–forward cloning templates, }\odot \text{ is the template–}
\]

\[
dot \text{–product operator, } i \text{ and } j \text{ are the index variables of the neural cell.}
\]

2.2. Discrete–Time and Full Signal Range CNN Models

The discretization of (1) with forward–Euler approximation

\[
\frac{dx_{ij}(t)}{dt} \approx \frac{x_{ij}^{n+1} - x_{ij}^{n}}{T_s}
\]

yields

\[
x_{ij}^{n+1} = x_{ij}^{n} + T_s \left( -x_{ij}^{n} + A \odot Y_{ij}^{n} + B \odot U_{ij} + z_{ij} \right). \quad (3)
\]

Although direct implementation of (3) is possible, Full Signal Range (FSR) CNN model is easier to implement [7]. The FSR model is obtained by taking \( \hat{x}_{ij} = y_{ij}^{n} \) in (3) as

\[
x_{ij}^{n+1} = (1 - T_s)y_{ij}^{n} + T_s A \odot Y_{ij}^{n} + T_s B \odot U_{ij} + z_{ij}. \quad (4)
\]

This equation can be written as

\[
x_{ij}^{n+1} = \hat{A} \odot Y_{ij}^{n} + \hat{B} \odot U_{ij} + \hat{z}_{ij}
\]

where

\[
\hat{A} \odot Y_{ij}^{n} = (1 - T_s)y_{ij}^{n} + T_s A \odot Y_{ij}^{n},
\]

\[
\hat{B} = T_s B, \quad \hat{z}_{ij} = T_s z_{ij}.
\]
2.3. Decomposition of the CNN Model

In equation (5) the term

\[ g_{ij} = \bar{B} \odot U_{ij} + \bar{z}_{ij} \]  

(6)
is calculated only once for each frame and used as a constant in every Euler iteration

\[ x_{ij}^{n+1} = \bar{A} \odot Y_{ij}^{n} + g_{ij}, \]  

(7)
The equations (6) and (7) can be mapped to B–Processing Unit (BPU) and A–Processing Unit (APU) respectively. The computation flow in time can be given as

\[
\begin{align*}
\text{BPU} & = g_{ij} = \bar{B} \odot U_{ij} + \bar{z}_{ij} \\
\text{APU(1)} & = x_{ij}^{1} = \bar{A} \odot Y_{ij}^{0} + \text{BPU} \\
\text{APU(2)} & = x_{ij}^{2} = \bar{A} \odot \text{APU(1)} + \text{BPU} \\
\vdots \\
\text{APU(N)} & = x_{ij}^{N} = \bar{A} \odot \text{APU(N-1)} + \text{BPU}
\end{align*}
\]

(8)

where \( N \) is the total number of Euler iterations. In short each Euler iteration is carried out by a different processor. The computed state is passed through the activation function at the output of each processor before entering to the next one. The initial condition \( Y_{ij}^{0} \) is either a constant or the input image for the most CNN algorithms.

3. Structure of The RTCNNPv2

The system consists of a video source, video decoder, FPGA, video encoder and a video sink (Figure 1). The video source can be any analog (VGA) or digital (DVI, HDMI, etc.) progressive video stream. The video sink may also be any analog or digital monitor, TV, etc. that accepts progressive video stream.

![Figure 1: Block diagram of the system.](image)

3.1. General Structure

A feed–forward video processing array is embedded in an FPGA that emulates the CNN (Figure 2). Video input block accepts control signals from the video decoder and translates them for the video processing array. Similarly, the video output block translates the control signals according to the video sink specifications.

Video processing array is an array of BPU and APU processors. The flow is organized according to (8). The BPU result is propagated through APUs. Each APU takes the results of the previous APU and the BPU as inputs and gives the new APU and the old BPU results to the next APU.

![Figure 2: Block diagram of the FPGA implementation.](image)

BPU and APU processors are designed as a single programmable Processing Unit (xPU) (Figure 4). APU/BPU input configures the processor as BPU or APU. clk, clka and clkp clocks are the processing, programming and pixel clock respectively. data_in, const_in, data_out and const_out are the data inputs and outputs. Their functionality depends on the xPU type, BPU or APU. sdata_in and sdata_out are the serial programming input and output respectively. hframe_in, hframe_out, vframe_in and vframe_out signals are the control signals that control the xPUs and propagate through APUs.

3.2. Local Control Structure

The main problem of controlling a processor array is the complexity of the central control unit. Every address input and every multiplexer select signal of each BPU and APU should be controlled independently and synchronously.

For 1–neighborhood CNN, consider the optimized and non–optimized data RAM structures in three consecutive APUs (Figure 3). The non–optimized memory structure is easier to control as the pixels at the same column are being processed by the processors. The optimized memory structure consumes less RAM in exchange for the control complexity as every control signal of each processor is different from one another. Furthermore, the control signals will be affected if the internal structure of the processors or number of iterations change.

![Figure 3: Memory usage of consequent APUs (light gray), and pixels that are being processed (dark gray).](image)

As seen from our previous processor RTCNNP-v1, the Block RAMs (BRAMs) and multipliers are the bottlenecks of such FPGA implementations and only about 30% of the logic blocks are used beside them [6]. This fact suggests
that there are plenty of logic elements to use in the processors. Consequently, a local control structure is proposed instead of a huge, complex and non-flexible central control unit [1].

There are two control signals that control each xPU: horizontal frame signal $hframe$ and vertical frame signal $vframe$. There is a blank area around the real video signal defined by the video standards. In this case there are nine different areas on the frame (Figure 5). $hframe$ and $vframe$ control signals define not only the visible area, but also define starting and ending points of the frame and lines, required for control. Video input block generates these control signals and controls the BPU, BPU propagates these signals according to its latency and control the first APU, and so on. Finally video output block receives these signals and translates them to standard video control signals of DVI, VGA, etc..

![Figure 4: Simplified block diagram of the xPU.](image)

Data and constant RAMs are optimized as seen in Figure 3b. They generate their own addresses, but reset and enabled by the local control. The RAM structure looks like circular buffers or shift registers from the functional perspective. There is also a boundary condition generation block that passes the input data or the boundary condition to the data RAM.

Frequency of the processing clock $clk$ is obtained by multiplication of the pixel clocks $clkp$ with a positive integer. By making the processing clock faster means less multiplier resources may be used, e.g. if clock frequency multiplier is 3, 3 multipliers are enough to calculate 9 multiplications in 3 clocks instead of 9 multiplications in 1 clock. $clka$ is the auxiliary clock used for serial programming of the template RAM, boundary condition, etc. The serial programming signal is also propagated through the processors in order to avoid long global interconnections.

The data and constant outputs are defined as

\[
\begin{align*}
data_{out} &= data_{in}, \\
const_{out} &= data_{in} \odot B + const_{in}
\end{align*}
\]

for the BPU. The data output is the initial condition of the first APU, which is the input pixel in many cases, and the processed data is the constant input of the first APU. On the other hand, for APUs

\[
\begin{align*}
data_{out} &= data_{in} \odot \bar{A} + const_{in}, \\
const_{out} &= const_{in}
\end{align*}
\]

as the processed data is the next APU’s data input, and the constant output is the BPU result that propagates through the APUs.

4. FPGA Implementation Results

The proposed system is coded in VHDL and partially implemented on a high-end Altera® Stratix® IV GX 230 FPGA. The unimplemented part is the serial programming feature. Template and boundary conditions are entered directly into the VHDL code and fixed during the synthesize operation.

Templates of the Global Connectivity Detection is chosen for the implementation as all synaptic weights are non-zero and the algorithm is suitable for prototyping [9]. The templates and bias are

\[
\bar{A} = \begin{bmatrix} 6 & 6 & 6 \\ 6 & 9 & 6 \\ 6 & 6 & 6 \end{bmatrix}, \quad B = \begin{bmatrix} -3 & -3 & -3 \\ -9 & 9 & -3 \\ -3 & -3 & -3 \end{bmatrix}, \quad \bar{z} = -4.5
\]

(9)

which are represented with signed 16-bits. The input and the intermediate states are chosen to be 8-bits, and the BPU output is propagated as 16-bits.
### Table 1: Resource usage of the prototype.

<table>
<thead>
<tr>
<th>Blocks</th>
<th>Combinational ALUTs</th>
<th>Dedicated Logic Registers</th>
<th>Block Memory (Kb)</th>
<th>DSP 18-bit Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>One xPU</td>
<td>221-380 (0.1-0.2%)</td>
<td>370-491 (0.2-0.3%)</td>
<td>minimum 8 (0.6%)</td>
<td>6 (0.5%)</td>
</tr>
<tr>
<td>100 xPUs</td>
<td>23635 (12.9%)</td>
<td>35651 (19.6%)</td>
<td>871 (67.6%)</td>
<td>600 (46.6%)</td>
</tr>
<tr>
<td>Glue logic</td>
<td>1072 (0.6%)</td>
<td>668 (0.4%)</td>
<td>0 (0%)</td>
<td>0 (0%)</td>
</tr>
<tr>
<td>Total</td>
<td>24707 (13.5%)</td>
<td>36319 (19.9%)</td>
<td>871 (67.6%)</td>
<td>600 (46.6%)</td>
</tr>
</tbody>
</table>

The properties of the test video signal are Full HD 1080p@54Hz (1920x1080) with reduced blanking 134 MHz pixel clock frequency. The clock frequency multiplier is 2 that gives 268 MHz processing clock that is easier to synthesize with little or no further optimization. 99 Euler iterations are embedded in the FPGA, hence 99 APUs and 1 BPU resides in the prototype (100 xPUs).

The system is fully pipelined. The throughput of the system is the same as its input. The latency is about 200 lines of a video frame for 100 iterations, which corresponds to 3.5 ms for the video properties given above.

Resource usage statistics are given in Table 1. The number of multipliers used in each xPU should be even due to the architectural limitations of the FPGA. Hence even if 5 multipliers are sufficient to do 9 multiplications in 2 clock cycles in theory, the architecture of the FPGA forces the usage of 6 multipliers.

### 5. Conclusion

Structure of the next generation RTCNNP-v2 is presented with more details. The structure is also coded in VHDL and a prototype is realized on an FPGA. Only serial configuration feature is not implemented. Central address and control generation is not necessary as each block controls the next block in a synchronous pipelined manner. The local control design has proved to make the processor array fast, reconfigurable and reusable.

It is also worth to state that this is the fastest CNN implementation for the time being that even outperforms the analog CNN chips in many algorithms. Although analog CNN makes the processing in almost zero time, video input/output is their primary bottleneck. This design targets specifically the input/output data bandwidth problem. 100 CNN iterations of the 54 Frames Full HD 1080p video signal means that 100 billion Multiply–ACcumulate (MAC) operations or 200 billion total additions and multiplications are done in one second. It means that the system has 1/5th the computational power of a supercomputer.

Finally this design is implemented on a high-end FPGA for benchmark purpose. It is possible to do downsampling in order to fit the design in a smaller FPGA with few changes.

### Acknowledgment

This research was supported by The Scientific and Technological Research Council of Turkey — TÜBİTAK under project number 108E023.

### References


Hardware-Software Co-design of Nonlinear Active Wave Generator
with Microblaze Soft Core Processor

Selman Ergünay, Ramazan Yeniçi and Müştak E. Yalçın

Department of Electronics and Communication Engineering,
Istanbul Technical University,
Istanbul, Maslak, TR-34469, Turkey.
Email: {selman.ergunay, ramazan.yeniceri, mustak.yalcin}@itu.edu.tr

Abstract—This paper introduces a hardware - software co-design of a Cellular Nonlinear Network (CNN) emulator. The Cellular Nonlinear Network is an two-dimensional array of locally coupled Nonlinear Processing Elements (NPEs). The network is used to observe evolution of spatio-temporal waves such as autowaves, travelling waves and spiral waves. The software part of the design controls the custom NPEs which are the hardware part of the design. In order to reach high performance, the software part provides control flexibility on hardware part which is vital for developing and executing active wave based computing algorithms. The system has been implemented on Xilinx Spartan 3E1600E FPGA with Microblaze soft core processor using Embedded Development Kit (EDK) design tools.

1. Introduction

Active wave based computing algorithms have received considerable attentions in the recent years because of new findings on biological systems. Biological systems can be considered as continuous distributed information processing system and process biological information. Retina is a good example for such a system which uses spatio-temporal waves for processing the visual information [2, 1].

In order to mimic the way biological systems process biological signals a practical network model is needed. Most recently a two dimensional reaction-diffusion Cellular Neural Network (CNN) [4] which consists of relaxation oscillators has been presented for wave computing applications. A fast FPGA implementation of real time autowave generation has been introduced in [5] and its application in robotic using an active wave computing algorithm has been presented in [7].

This paper is organized as follow: Section 2 the NPE which is the cell implementation of the network and its mathematical model is described. In Section 3, the hardware architecture of the network is explained and the software design is given in Section 4. The paper is concluded in Section 5.

2. Nonlinear Processing Element

A nonlinear processing element of the CNN model is described by the differential equation in (1).

\[
\begin{align*}
\dot{x}_{i,j} &= \alpha x_{i,j} + \beta y_{i,j} + g(x_{i,j}) + I_{i,j} + u_{i,j}, \\
y_{i,j}(k+1) &= y_{i,j}(k) + \tau (\epsilon x_{i,j} + \sigma y_{i,j}).
\end{align*}
\]

Where \( x \) and \( y \) are the state variables, \( g(\cdot) \) is the nonlinear function:

\[
g(x_{i,j}) = \begin{cases} 
\mu \cdot (x_{i,j} - \lambda) & \text{if } x_{i,j} > \lambda; \\
0 & \text{if } |x_{i,j}| \leq \lambda; \\
\mu \cdot (x_{i,j} + \lambda) & \text{if } x_{i,j} < -\lambda;
\end{cases}
\]

and the synaptic law \( I_{i,j} \) of the model is given by

\[
I_{i,j} = a_{i,j+1}x_{i,j+1} + a_{i-1,j}x_{i-1,j} + a_{i,j-1}x_{i,j-1} + a_{i+1,j}x_{i+1,j}.
\]

The given network model (1) which has been introduced by Yalcın in 2008 [4] is a locally coupled relaxation oscillators. Any cell on this network can be fixed to constant value \( x_{i,j} = x_{\text{fixed}} \). This is one of the essential features (called fixed-state) which has been already used in the active wave computing algorithms. Figure 1a and 1b show two snapshots depicting the obtained autowaves during the time evolution for two different parameter sets [4].

![Figure 1: Propagation of the autowaves on the network.](image)

The discrete-time model of the model (1) is given by

\[
\begin{align*}
x_{i,j}(k+1) &= x_{i,j}(k) + \tau \left[ \alpha x_{i,j}(k) + \beta y_{i,j}(k) + g(x_{i,j}(k)) + I_{i,j}(k) + u_{i,j} \right], \\
y_{i,j}(k+1) &= y_{i,j}(k) + \tau \left[ \epsilon x_{i,j}(k) + \sigma y_{i,j}(k) \right].
\end{align*}
\]

- 157 -
where
\[ g(x_{i,j}(k)) = \begin{cases} 
  m \cdot (x_{i,j}(k) - \lambda) & \text{if } x_{i,j}(k) > \lambda; \\
  0 & \text{if } |x_{i,j}(k)| \leq \lambda; \\
  m \cdot (x_{i,j}(k) + \lambda) & \text{if } x_{i,j}(k) < -\lambda; 
\end{cases} \]

and
\[ I_{i,j}(k) = a_{i,j+1}x_{i,j+1}(k) + a_{i-1,j}x_{i-1,j}(k) + a_{i,j-1}x_{i,j-1}(k) + a_{i+1,j}x_{i+1,j}(k), \] (6)

The simple Forward Euler Method has been used to discretize the model. The discrete-time model has been already implemented on an Field Programmable Gate Array in [5]. In order to use the network as an active wave computer, an improved design has been presented in [6].

The operations to compute the next-time value of the state variables \( x \) and \( y \) are done by a special digital hardware called NPE. In Figure 2, a detailed block diagram of the NPE circuit is given.

Half-precision floating-point number arithmetic is used for the NPE implementation. The parameters of the network is first loaded into the NPE and then NPE calculates the state values of the next-time at a short interval with respect to its arithmetic workload.

In the following section how NPEs are organized to emulate the CNN is described and the observed dynamical behavior on the network is presented.

3. System Architecture

The NPE are organized as \( 5 \times 5 \) processor array in [5] and \( 4 \times 4 \) processor array in [6]. These designs are introduced a full hardware solution to emulate the CNN. The operation of the NPE arrays in these works are controlled by specially designed hardwares which only improves performance but do not have flexibility. While designing pure hardware it is required to change digital circuitries, connections between them and consider time scheduling to change the dynamic behavior of the network emulator.

In this work, there NPEs are operated by a Microblaze processor. The nonlinear network is emulated by NPE hardwares and the software runs on the Microblaze that loads values to NPEs, runs and collects computed values from them. If it is asked to add supplement expressions to the network model such as a new nonlinearity, new synaptic law or a new state equation, this requirement can be met by just changing the software. Hence a tradeoff between flexibility and speed exists. Block diagram of the design is given in Figure 3.

In this paper, Xilinx Embedded Development Kit (EDK) [3] which is a set of tools that enables to design and implement processor based digital system in Xilinx FPGAs is used to build up the system. One of these tools is Xilinx Platform Studio (XPS) to make hardware project of system and the other one is Platform Studio Software Development Kit (SDK) for preparing software part of design. While configuring microprocessor and interconnection with peripherals, XPS is used. SDK provides creation and verification C software application.

A 32-bit MicroBlaze soft core processor which is a reduced instruction set computer (RISC) and optimized for Xilinx FPGAs is added to the design. Instruction and data accesses are done in separate address spaces because MicroBlaze is implemented with Harvard memory architecture. These address spaces are located in 64 Kbyte Block RAM. Owing to BRAM is chosen as boot memory, reset vector of MicroBlaze points to the start address of Block RAM. When bitstream of design is sent to FPGA, Microblaze starts to execute the software.

In this design, Microblaze has two different types of data buses, Processor Local Bus (PLB) to communicate with serial port and NPEs, LMB (Local Memory Bus) to access BRAM for execution of the software.
Instruction side and data side LMB buses enable to connect dual ported BRAM via separate LMB BRAM interface controllers.

Processor Local Bus (PLB) provides a high-speed interface between the processor and high-performance peripherals. Designed custom Intellectual Properties (here NPEs) are connected to MicroBlaze via PLB. 14 software accessible registers are created in top module of NPE, so inputs can be loaded and outputs can be read via these registers. In this work, parameters of NPE and initial values of $x$ and $y$ state variables are loaded to these registers and control signals of NPE are managed by changing corresponding bits of related register.

After connecting modules to buses, bitstream file of hardware project is generated by using ISE tools. Obtained address map of system is used to generate library and Board Support Package (BSP) files, and these files enable to control software accessible registers of NPEs.

The number of used NPEs in this design can be increased while considering the area consumption of the Field Programmable Gate Array and service time of the Microblaze processor to each NPE. It is efficient to increase the number of NPEs if the execution time of NPE is much more than the service time of the MicroBlaze to each NPE. Service time is defined as the time required while the processor is loading data into the NPE before it starts to run.

4. Active Wave Generation Software

Many applications can be implemented by changing only the software. When (4) is analyzed, only a sum operation or a multiplication can be calculated by loading proper parameters and initial values to the NPE by just changing the software. Coupling defined by the synaptic law (6) can be changed by adding new neighbor signals on to the state value calculated by NPE using the software flexibility. Also, pipeline stages, CNN dimensions, initial values and parameters can be easily adjusted by the software. In this work, the $9 \times 9$ cellular nonlinear network (1) is emulated and spatio-temporal waves on the network are observed in real-time.

In the sequential CNN emulator architecture, after loading initial values of the current cell on the network, the next cell value can only be calculated after the response of the NPE for the current cell. Instead of having this sequential hardware architecture, a pipeline is implemented. After initialization of one NPE with the data of a cell, without waiting the NPE's response, MicroBlaze passes to load next cell data to the other NPE. Number of NPE determines pipeline stage number, so in this work 3 stage pipeline structure is used. Scheduling of three NPE in the pipeline is shown in Figure 4.

Figure 4: Scheduling of three NPE by the Microblaze controller.

There are 4 matrices that have $9 \times 9$ sizes in 2-dimension to save both current and calculated next time values of $x$ and $y$ state variables.

After the computing and saving obtained values to next-value-matrix for each state variable of each cell on the CNN, pointers of current and next-time matrices interchanges and so next value matrix becomes current value matrix. After the coded numbers of iteration is executed by the emulator, the calculated values are sent to a computer via serial port.

Figure 5: $9 \times 9$ network emulation.

5. Results and Conclusion

In this paper, a hardware - software co-design of the CNN is introduced and an implementation of this design is presented. When a comparison is done between this emulator design and previous pure hardware designs, it is obvious that the performance of the emulator worsens but the flexibility improves. In order to use the active wave computing based algorithms in the applications, hardware implementations of the wave computer are vital to deal with computational cost of the wave generating part of the algorithm. However full hardware implementation restricts the application specific part of the active wave computing algorithm. Currently, the application specific part of the
algorithms are implemented on the computer. Communication between computer and the wave computer hardware is another aggravating effect on performance. The proposed hardware-software co-design might help to designer to have a better performance and flexibility for their active wave computing algorithms.

Acknowledgments

This work was supported by The Scientific and Technical Research Council of Turkey, under project 105E103.

References


Standard C++ Compiling to GPU with Lambda Functions

Ádám Rák† and Gergely Feldhoffer†‡ and Gergely Balázs Soós‡ and György Cserey†∗

†Faculty of Information Technology, Pázmány Péter Catholic University,
Budapest, Práter u. 50/a. 1083, Hungary
‡StreamNovation Ltd. Budapest, Práter u. 50/a. 1083, Hungary
* Infobionic and Neurobiological Plasticity Research Group,
Hungarian Academy of Sciences, Pázmány Péter Catholic University, Semmelweis University,
Budapest Práter u. 50/a, H-1083, Hungary
Email: [rakad,flugi,soos,cserey]@itk.ppke.hu

Abstract—In this paper, a new method of a compiler application to GPU is introduced. In this method, a hybrid executable is generated from the C++ lambda function based code. Our compiler plugin creates GPU accelerated subroutines from code using our library. A C++ runtime library is designed embedding the generated GPU code into the original project.

1. Introduction

New generation hardwares contains more and more processors and the trends show that these numbers will intensely increase in the future. The question is how could we program these systems and may we port earlier codes on them? There is a huge need for this today as well as in the forthcoming period. Our new approach of the automation of software development may change the future techniques of computing science.

Exploiting the advantages of the new architectures needs algorithm porting which practically means the complete redesign of the algorithms. New parallel architectures can be reached by “specialized” languages (CUDA, OpenCL, Verilog, VHDL, etc.), for successful implementation, programmers must know the fine details of the architecture. After a twenty years long evolution, efficient compiling for CPU does not need detailed knowledge about the architecture, the compiler can do most of the optimizations. Can we develop as efficient GPU (or other parallel architecture) compilers as the CPU ones? Will it be a two decade long development period again or can we make it in less time?

The specification of a problem describes a relationship from the input to the output. The most explicit and precise specification can be a working platform independent reference implementation which actually transforms the input from the output. Consequently, we can see the (mostly) platform independent implementation, as a specification of the problem.

Parallelization must preserve the behaviour in the aspect of specification to give the equivalent results, and should modify the behaviour concerning the method of the implementation. Automated hardware utilization has to separate the source code (specification) and optimization techniques on parallel architectures.

There are different trends and technical standards emerging. Without the claim of completeness, the most significant contributions are the following: OpenMP [1] - supports multi-platform shared-memory parallel programming in C/C++ and Fortran, practically it uses pragmas for existing codes. OpenCL [2] - is an open, standard C-language extension for the parallel programming of heterogeneous systems, also handling memory hierarchy. Threading Building Blocks of Intel [3] - is a useful optimized block library for shared memory CPUs, which does not support automation. One of the automation supported solution providers is the PGI Accelerator Compiler [4] of The Portland Group Inc. but it does not support C++. There are problem-software or language specific implementations on many-core architectures, one of them is a GPU boosted software platform under Matlab, called AccelerEyes’ Jacket [5]. Overviewing the growing area, there are successful partial solutions, but there is no universal product and still there are a lot of open problems.

Our aim is machine learning boosted OpenCL parallelization of any standard C++ source code by separating programming and parallelization meta-programming. This presentation shows that the basic technological problems (OpenCL source code generation, host code generation and insertion) are manageable: a C++ library is introduced, which can be compiled with every C++0x standard [6] compatible compiler, and produces CPU code. Our compiler plugin and C++ library creates GPU accelerated executables. This approach is methodically one step after the Intel Thread Building Blocks, because the parallelization schemes and memory access patterns are still fixed and provided by our library, but the building blocks themselves become completely user defined in the form of lambda functions.

This paper is organized as follows. After the introduction, in section 2 a general overview of the architecture of the new generation GPUs is given. The lambda functions in the new standard C++ are depicted in section 3. In section 4 we introduce our Minotaurus project which is a gcc based
Figure 1: In the case of a GPU, most of the parts of a normal CPU are sacrificed to place the maximum amount of processing units on the chip. In most cases, one core is completely reduced to a simple 32bit FPU/ALU pair, and many cores use the same execution control units on the chip. Results and working demonstrations are presented in section 5.

2. GPU architecture

Complex real-time 3D rendering needs considerable computing power, orders of magnitude greater than what one CPU can provide. But fortunately the algorithms are all data-parallel, which means that the same code must be executed on all the threads, just the processed data is different. These requirements gave rise to the massively SIMD parallel GPU architectures nowadays. Most of the parts of a normal CPU are sacrificed to place the maximum amount of processing units on the chip. In most cases one core is completely reduced to a simple 32bit FPU/ALU pair, and many cores use the same execution control units on the chip. The pipelines are generally very deep, further allowing more optimization of the architecture. While CPUs need serious trickery, both in hardware (branch prediction, instruction reordering) and sometimes in software too (compilers) to deal with deep pipelines, GPUs do not need this because rendering specific algorithms utilize massively huge amount of threads, much more than the number of cores, which makes it very easy to fill the pipelines. This is possible because every thread runs independently on different data, so there is no dependency between them, so on every core, on every pipeline stage a different thread can be executed. The scheduling of threads is done in the hardware to reduce the overhead.

OpenCL provides us an abstraction of the massively parallel hardwarees, where both the computing resources (cores) and the memory is hierarchial. This approach was introduced by the hardware manufacturers and it seems that the multicore industry is heading this way. It is suspected [7] that currently this is the optimal trade-off between programmability and performance, where the highest performance is represented by the FPGAs (Field Programmable Gate Array) where everything is parallel, and the maximal programmability by the single core CPU with a single thread running. An important feature is that the memory can be accessed in 1D, 2D or 3D topography, accelerated by the 2D aware hardware caching, and the virtual indexing of the threads can also follow this scheme.

3. Lambda functions in C++

The use of "lambda" originates from functional programming and lambda calculus, where a lambda abstraction defines an unnamed function. In the new standard of C++ (known also as C++0x) the syntactic element of lambda function is introduced to improve functor usability in templates. The lambda function is an inline expression of a functor object. It is nameless, only a few syntactic units can be given: the captured variables, the parameters, the type of the returning value, and the function body. The created functor will have the captured variables as members, and the constructor will assign the values. The operator() will be created with the parameters and the given function body. The function body is limited to use local variables, the parameters, and the captured variables. There is a convenient way to capture all of the local stack variables in the context as well.

Lambda functions are designed to be used where functors are passed but there is no need to reuse the functor class anywhere else, and building a whole class in order to fulfill syntactic requirements for only one use is circuitous.

4. Minotaurus project

The flowchart in Figure 3 shows the main components of the compiler using our plugin called Minotaurus. As usually, the gcc compiler has three parts, a frontend, a middle end and the backend. Minotaurus connects to the middle end using the inner representation of a gcc compiler besides the GPU application output, generating an OpenCL
code based GPU-accelerated application output.

Simplified problem: the programmer specifies the code parts that can be run efficiently on GPU in lambda functions. Minotaurus compiles the lambda function to extract the data and control flow, and synthesizes the OpenCL source code which is semantically equivalent to the lambda function. The programmer picks a template function to express the pattern of use (scan primitive for example), and gives the input data. The template function contains the host code which feeds the GPU kernel function. This is where we are standing right now.

4.1. Standard C++ code input

With Minotaurus, it is possible to compile CPU only executable and CPU-GPU mixed executable as well, using only standard C++ language elements in the common source code. This is useful if the debugging process is more complicated with GPU codes, which is usually far more complicated indeed. There are small differences between the resulting executables concerning floating point precision for example, but the theoretic correctness of the implemented method can be checked.

4.2. Using lambda functions to specify kernels

The fundamental benefit of using lambda functions for compiling to CPU-GPU mixed executable is the clear separation of function and parameter data. For the compilation of the general code, the compiler must explore the full data dependency of the given function to transport the required data to the GPU platform before code execution. This data dependency can be hard to follow because of reading global variables, pointers to globals, etc. Lambda functions are closed in this term, besides local variables, only the captured values and the parameters are accepted inside the function. These variables are given explicitly so any template function can handle the memory transfer to the GPU, so the data dependency of the GPU targeted code can be satisfied.

4.3. Automatic code generation

The function body of the lambda function may contain elements of C++, such as complex<> type, or references. Minotaurus can convert these elements to a semantically equivalent OpenCL code, using pointers instead of references for example. Lambda functions will be converted to OpenCL functions.

The host code is also generated, the memory transfer can be handled based on the lambda functions’ members. The converted lambda functions are called from generated kernel functions based on the parameter set of the lambda function. The host code copies the actual data to the kernel functions, enqueues the kernel, and reads GPU memory to the returning variables.

4.4. Extendable technology to other languages

Since Minotaurus works in the inner representation of the compiler, most of the functionality does not depend on the input language.

4.5. Towards automatic GPU code generation

Some of the hard work is done on Minotaurus right now, but there is plenty of work ahead. Automatic data dependency exploration of any code segment is required for general CPU-GPU hybrid compilation. Functional dependency and guaranteed traces of control are needed in order
to select a section of the code which can be compiled to GPU function (entry point of the generated function and the returning points - it is trivial in lambda functions).

4.6. GPU code generation is functional

We can now compile C/C++ code to OpenCL functions and host code, so we are able to create a hybrid executable from purely C++ code. The performance gain is heavily task dependent, it can be even 80x speed-up. This is notable since the conversion is purely mechanical, no additional tweaking is done with OpenCL local variables, and other sophisticated techniques yet.

5. Results and demonstrations

Figure 4. demonstrates the usage of the lambda function in our system. The fill_matrix() function works as a solution template. The solution template contains hardware specific parallelization schemes and the memory access patterns. The lambda function is defined as a parameter of the function. The implementation of the algorithm is coded by the lambda function. In this given case, it generates a julia-set demonstrating the exploitation of the C++ advantages (see Figure 5.).

The GPU accelerated version reached up to 25x performance gain on the same source code, utilizing the parallel GP-GPU technology (NVIDIA GTX 280) compared to the OpenMP Intel i7 4 cores implementation. This approach provides C++ support in the kernel code and shows a proof of concept for automatic GPU code (OpenCL) generation.

Acknowledgement

The Operational Program for Economic Competitiveness (GVOP KMA), the support of NVIDIA Professor Partnership Program and the Bolyai János Research Scholarship is gratefully acknowledged. The authors are also grateful to Professor Tamás Roska for discussions, his suggestions and his never ending patience.

References

Suppression of Nonlinear Traffic Pattern Generation using Dynamic Bandwidth Management Method

Shigeaki Tanimoto† and Yohsuke Kinouchi‡
†Faculty of Social Systems Science, Chiba Institute of Technology
2-17-1, Tsudanuma, Narashino-shi, Chiba, 275-0016 Japan
‡Institute of Technology and Science, The University of Tokushima
2-1 Minami-Josanjima, Tokushima-shi, Tokushima, 770-8506 Japan
Email: shigeaki.tanimoto@it-chiba.ac.jp, kinouchi@ee.tokushima-u.ac.jp

Abstract—Generally, the elements in a network must be duplicated for securing reliability. However, in a private network, such as in a company, a partially duplicated method, which carries out dual-homing of the terminal to a node, is used in addition to considering reliability. The reason is that cost is important when considering the design of private network. With this method, if a certain node breaks down, the terminal will be switched to another node. In a dual-homing node, twice the amount of traffic will be quickly generated. In other words, a nonlinear traffic pattern is generated. Although current methods, such as a redundant bandwidth management method and a congestion-control method, are used to suppress nonlinear traffic pattern generation, they have advantages and disadvantages in bandwidth utilization ratio or service convenience.

We propose a new dynamic bandwidth management method, which combines high bandwidth utilization ratio and good service convenience. Specifically, our method uses the distributed management method, which divides the bandwidth of the entire network into two, for actual communication, and for spare communication, and the traffic-predicting management method, which is used to observe communication in real time in each node and to adjust the bandwidth of each node just enough for reserved bandwidth. We evaluated our method’s effectiveness using a traffic simulation.

1. Introduction

Generally, elements in a network must be duplicated for guaranteeing reliability. That is, to secure reliability, it is necessary to duplicate all the elements that constitute communications, such as nodes and channels. On the other hand, in a private network, such as in a company, a partially duplicated method, which carries out dual-homing of the terminal to a node, is used in addition to considering reliability since cost is important when considering the design [1]-[3]. In this dual-homing method, when a certain node breaks down, the terminal will switch attribution to another node. In this case, the traffic of the node by which dual-homing was carried out increases rapidly and becomes about twice the usual amount. That is, a nonlinear traffic pattern is generated. Thus, if a nonlinear traffic pattern is generated, bandwidth will run short and service convenience will deteriorate, and call loss will occur. The redundant bandwidth management method and the congestion-control method have been studied to solve the problem of generating a nonlinear traffic pattern. However, in the redundant bandwidth management method, since bandwidth is too secure, the bandwidth utilization ratio worsens. In the congestion-control method, because of the restriction traffic in dual-homing, service convenience deteriorates.

We propose a dynamic bandwidth management method, which combines high bandwidth utilization ratio and good service convenience. This method divides the bandwidth of the entire network into two, for actual communication, and for spare communication. The bandwidth assigned to actual communication measures the amount of bandwidth used in real time by each node and predicts a future traffic pattern. From the prediction results, the excess bandwidth of a node is returned to the bandwidth for spare communication, and a node with insufficient bandwidth requests bandwidth from spare communication. We conducted a traffic simulation of the proposed method to evaluate its effectiveness. The result of this simulation shows that the proposed method, which combines high bandwidth utilization ratio and good service convenience, suppresses nonlinear traffic pattern generation.

2. Multimedia LAN systems

Here, a multimedia LAN system is shown as the network configuration, which is the requisite for our proposed method. This system integrates a Private Branch Exchange (PBX) and the Internet, which is as general network configuration of a company. In this system, the node on the backbone LAN is set to the station (STN), and the terminal is set to the PBX and branch line LAN. The terminal, which carries out dual-homing, is the PBX.

2.1. Network configuration

As shown in Fig. 1, a multimedia LAN system consists of an optical loop network to relay PBXs and branch LANs. It achieves switching system scalability, in terms of the number of users, by distributing PBXs, being capable of integrating circuit and packet switching in multimedia LANs and contains highly reliable duplicated optical transmission lines.
Abbreviations:
A-SVS: Active SVS
S-SVS: Standby SVS
SVS: Supervisory station
STN: Station

(a) Multimedia LAN system (b) Main Specification

Fig. 1 Multimedia LAN system

The hashed area in Fig. 1 (a) is the optical loop network. It consists of stations and duplicated optical transmission paths. Two kinds of stations, usually STNs and supervisory stations (SVSs), can be used depending on the hardware configuration.

An STN possesses basic functions for transmission line access management and for connecting PBXs and branch LANs. An SVS has supervisory functions for frame creation, clock generation, and management of the entire optical loop network. To increase system reliability, several SVSs are set up in the loop. One of the SVSs is active (A-SVS) and manages the optical loop network, and the others are on standby (S-SVSs).

Figure 1 (b) shows the main specifications of this system.

2.2. Frame structure

The frame structure of the multimedia LAN system consists of a frame header (FH), an A-SVS address (SV), an information control (CT) area, a circuit-switching (CX) area, a packet-switching (PX) area, and a boundary value (BV) for the CX and PX areas. Since this structure provides circuit switching, a transmission frame has a period of 125 microseconds and consists of timeslots fixed at a length of 64 Kbps for all frames but the frame header.

In this frame structure, the CX and PX areas can be dynamically changed for effective bandwidth use, and maintenance operation information (e.g., loop control information) and user data can be individually exchanged by separating the CT area and the data area (Fig. 2).

We propose a dynamic bandwidth management method in the data area shown in Fig. 2, including the CX and PX areas.

3. Suppression method of a nonlinear traffic pattern generation

When dual-homing, shown in Fig. 3 (b), occurs in an STN1, twice as much traffic occurs rapidly. That is, a nonlinear traffic pattern is generated. First, current suppression methods to this nonlinear traffic pattern are explained. However, there are some problems in these methods. Therefore, next, we explain the proposed method for solving these problems.

3.1 Current suppression methods

3.1.1 Redundant bandwidth management method

A redundant bandwidth management method always assigns a required bandwidth, when dual-homing occurs. By this method, even if generating of dual-homing, i.e., a nonlinear traffic pattern, occurs, since the bandwidth is assigned enough, so service convenience to a terminal is not deteriorated. However, in usual, an excessive bandwidth will exist and bandwidth use efficiency is bad.

3.1.2 Congestion-control method

By a congestion-control method, when dual-homing occurs, it directs to suppress traffic from STN to PBX. As a result, service convenience will be deteriorated.

Fig. 2: Frame structure

2.3. Dual-homing composition

As shown in Fig. 3 (a), in order to secure the reliability of communication between PBX, dual-homing composition is adopted in the multimedia LAN system. With dual-homing composition, as shown in Fig. 3 (b), the case where a certain failure occurs in STN2, PBX2 can be immediately switched to normal STN1 and it can continue communication between PBX.
However, in the case of the usual operation, since an excessive bandwidth does not exist, bandwidth use efficiency is high.

### 3.1.3 Problem of current methods

The problem of current methods explained until now is summarized as shown in Table 1.

#### Tab.1 Problem of current methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Service convenience</th>
<th>Bandwidth use efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Redundant bandwidth</td>
<td>good</td>
<td>low</td>
</tr>
<tr>
<td>management method</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Congestion-control</td>
<td>poor</td>
<td>high</td>
</tr>
<tr>
<td>method</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 3.2 Dynamic bandwidth management method

As shown in Table 1, it is difficult to be simultaneously satisfied with the current methods in terms of service convenience and bandwidth use efficiency. We propose a dynamic bandwidth management method for satisfying these requirements at the same time.

#### 3.2.1 Distributed management method

This proposed method installs two management authorities into the timeslots on a multimedia LAN. For the first authority, each STN individually manages the total number of timeslots of an STN (TSSTN) for distributed management to achieve real-time securing. As a result, at each individual STN, the timeslots can be allocated for the traffic generated.

For the other management authority, an A-SVS manages the remaining timeslots (TS_{SVS} = TS_{all} – n * TS_{STN}, where TS_{SVS} is the total number of timeslots of an A-SVS, TS_{all} is the total number of timeslots of the multimedia LAN system, and n is the number of STNs) for centralized control to efficiently manage timeslots. That is, only the timeslots necessary for normal operation at each station are allocated, and the A-SVS controls all of the remaining timeslots sharing them with the STNs. Thus, when the number of timeslots allocated at each STN decreases to a certain point, a timeslots-replenishment request is sent to the A-SVS beforehand. Conversely, when unused timeslots remain, each STN returns them to the A-SVS.

In this method, each STN can secure real time just as with the distributed management method by individually allocating timeslots to the communication between PBX.

#### 3.2.2 Traffic-predicting processing

Our dynamic bandwidth management method predicts traffic at each STN. Predicting traffic dynamically controls the surplus and shortage of assigned timeslots by counting free TSs upon assignment and release. Figure 4 shows the algorithm for this method. Each STN counts the free timeslots when they are assigned or released. For assignment of timeslots, the system requests supplement timeslots (TS_{sup} number of demand TSs from an STN to the A-SVS) from the A-SVS if the number of free TSs falls below the threshold (TS_{min}: lower limit); for release, the system returns the surplus timeslots (TSrel: number of release timeslots from an STN to the A-SVS) to the A-SVS if the number of free timeslots (TS_{free}: number of free timeslots at an STN) exceeds the threshold value (TS_{max}: Upper limit).

#### 3.2.3 Detailed explanation of proposed method

Figure 5 qualitatively explains the algorithm of Fig. 4. With increasing traffic to stations, that is, requests for timeslots assignment, free timeslots in STNs decrease. The system requests supplement of timeslots from the A-SVS as background processing separately from call processing when the number of free timeslots falls below the minimum value (TS_{min}) by predicting traffic, then receives supplement timeslots from the A-SVS (Fig. 5 (a)). Conversely, free timeslots increase when traffic decreases, so the system returns surplus timeslots to the A-SVS when the number exceeds the maximum value (TS_{max}) by estimating traffic estimate (Fig. 5 (b)). Therefore, this simulation ignores it because the delay time is shorter than the time scale of the CX traffic to be considered insignificant. In our dynamic bandwidth management method, traffic predicting allows the system to continuously follow traffic fluctuation with room for several timeslots. As a result, proposed method simultaneously makes service convenience and bandwidth use efficiency.

---

**Note:** Number of free time slots (TS_{free}) adjusts by addition and return in background.
Fig. 5 Qualitative explanation of proposed method

4 Evaluations

4.1 Simulation condition

As shown in Table 2, in order to simulate a switching system, we established the simulation conditions based on the design condition of PBX [5].

<table>
<thead>
<tr>
<th>Item</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traffic pattern</td>
<td>Circuit traffic: 1 TS/call</td>
</tr>
<tr>
<td>Generation of call</td>
<td>Poisson distribution</td>
</tr>
<tr>
<td>Holding time</td>
<td>Exponential distribution</td>
</tr>
<tr>
<td>Mean holding time</td>
<td>80 sec/call</td>
</tr>
<tr>
<td>Mean mass of call</td>
<td>0.11 erl/terminal</td>
</tr>
</tbody>
</table>

4.2 Traffic simulation

The traffic simulation was performed for evaluating the proposed method. A nonlinear pattern was generated by carrying out dual-homing.

The traffic simulation result based on the simulation conditions of Table 2 is shown in Fig. 6. Although dual-homing, i.e., a nonlinear traffic pattern was generated, as shown in Fig. 6 (a), this nonlinear traffic pattern generation was suppressed with our proposed method. That is, call loss did not occur (TS_all is always larger than TS_free) even during dual-homing with a traffic-predicting method since the bandwidth was increased in advance. Therefore, service convenience of the proposed method is better than the congestion-control method.

After dual-homing, i.e., generation of a nonlinear traffic pattern, completed (Fig. 6 (b)), excess timeslots were returned to the bandwidth for spare communication. As a result, the bandwidth-use efficiency of the entire network improved more than with the redundant bandwidth management method. These results are listed in Table 3.

<table>
<thead>
<tr>
<th>Method</th>
<th>Service convenience</th>
<th>Bandwidth use efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed method</td>
<td>good</td>
<td>high</td>
</tr>
</tbody>
</table>

5. Conclusion

We proposed a dynamic bandwidth management method, which can be used to simultaneously achieve high bandwidth-use efficiency and good service convenience, for suppressing nonlinear traffic pattern generation. Our proposed method uses the distributed management method and the traffic-predicting management method. The effectiveness of the proposed method was confirmed using a traffic simulation. The results of this evaluation confirmed that our proposed method is better than current methods and our method is also suitable for suppressing nonlinear traffic pattern generation in a private network.

References

Extraction of Image Regions Using Oscillatory Responses in Chaotic Neuronal Network

Mio Musashi†, Ken’ichi Fujimoto‡ and Tetsuya Yoshinaga‡

†Graduate School of Health Sciences and ‡Institute of Health Biosciences
The University of Tokushima
3-18-15, Kuramoto, Tokushima 770-8509, Japan
Email: musashi@anan-nct.ac.jp, {fujimoto | yoshinaga}@medsci.tokushima-u.ac.jp

Abstract—For dynamic image segmentation, which is used to extract connected components in an image and to exhibit them in a time series, we have developed a network model consisting of discrete-time chaotic neurons. According to an n-phase oscillatory response of neurons, the network model can segment n image regions within one processing. Based on the bifurcation analysis of periodic points in a reduced model of the network, we propose a successive algorithm for dynamic image segmentation of a gray-level image with an arbitrary number of isolated image regions, using our network model with two- and three-phase oscillatory responses.

1. Introduction

For dynamic image segmentation, which is used to extract connected components between pixels, i.e., isolated regions or blobs, in an image and to exhibit them in time series, we have developed a network model [1] consisting of discrete-time chaotic neurons [2,3]. A neuron can generate a similar oscillatory response formed by a periodic point with high order of period to an oscillation as observed in a continuous-time dynamical system. According to an n-phase oscillatory response of neurons, the network model can segment n image regions within one processing. In our previous studies [4–6], we derived reduced models, which are two- and three-coupled neuron systems, and analyzed the bifurcations of various kinds of synchronized oscillatory responses observed in the models.

This paper presents a successive algorithm for dynamic image segmentation of a gray-level image with an arbitrary number of isolated image regions, using our network model with two- and three-phase oscillatory responses. In our algorithm, the values of system parameters and initial values were determined based on the results of bifurcation analysis [4–7] so that two- and three-phase oscillatory responses can appear. Although the maximum number of image regions to be segmented is three in one processing, all image regions can be segmented by repeating the process. We demonstrate that the algorithm for dynamic image segmentation works for a gray-level image with six isolated image regions.

2. Scheme of Dynamic Image Segmentation

We need a network model consisting of N chaotic neurons and a global inhibitor for an N-pixel input image. As shown in Fig. 1, the N neurons are arranged in a grid so that neurons correspond to pixels in an input image and can connect to neurons in its neighborhood including itself; the global inhibitor connects to all neurons.

The dynamics of the ith neuron with two state variables (x_i, y_i), i = 1, 2, . . . , N is described as

\[ x_i(t + 1) = k_f x_i(t) + d_i + C_i(t) \]  \hspace{1cm} (1)

\[ y_i(t + 1) = k_y y_i(t) - \alpha \cdot g(x_i(t) + y_i(t), 0) + a \]  \hspace{1cm} (2)

where t denotes the discrete time. d_i denotes a direct-current input value (DC input value) corresponding to the ith pixel value. C_i represents the sum of excitatory couplings from neurons in its neighborhood including itself and an inhibitory coupling from the global inhibitor. It is described as

\[ C_i(t) = \sum_{k \in L_i} \frac{W}{M_i} g(x_k(t) + y_k(t), 0) - W \cdot g(z(t), 0.5) \]  \hspace{1cm} (3)

where L_i denotes a set of neurons corresponding to pixels with similar gray levels at the ith pixel and its four-neighborhood. M_i is the number of elements in L_i. The g(·, ·) represents the output function of a neuron or a global inhibitor and is described by

\[ g(u, \theta) = 1 / (1 + \exp(-u/\theta)) \]  \hspace{1cm} (4)

The dynamics of a global inhibitor is described as

\[ z(t + 1) = \phi \left\{ g \left( \sum_{k=1}^{N} g(x_k(t) + y_k(t), W), 0 \right) - z(t) \right\} . \]  \hspace{1cm} (5)

The system parameters except for k_f, φ, and d_i, i = 1, 2, . . . , N were set to k_f = 0.5, φ = 4, a = 0.5, W = 15, and ε = 0.1 according to Ref. [8].

Neurons corresponding to only pixels with high pixel values can generate oscillatory responses. Therefore, as seen in Fig. 1, segmented images can be outputted based on responses of neurons in a time series.
To segment an image with \( n \) isolated image regions, it requires an \( n \)-phase oscillatory response. However, only two- or three-phase oscillatory responses could be generally observed in our system from our preliminary experiments. This indicates that the number of image regions to be segmented using an oscillatory response is up to three.

3. Successive Algorithm

Since the number of image regions to be segmented by an oscillatory response is up to three, we need a successive algorithm using our system to extract all image regions from an image with many image regions. Figure 2 illustrates the flow chart of a successive algorithm we proposed to extract all image regions from an image with \( n \) isolated image regions. In a routine of the algorithm, the generation of two- or three-phase oscillatory responses is needed. Our previous studies [5, 6] showed a parameter region in which in-phase oscillatory responses do not occur and two- or three-phase oscillatory responses are always generated for appropriate initial values.

Let us consider the dynamic extraction of image regions from a gray-level image with five isolated image regions in Fig. 3(a). We assumed that the gray levels of pixels in an image region are same. At first, for the input image, the DC-input values of all neurons are assigned according to the values of all pixels. Based on a three-phase oscillatory response, the three segmented images in Fig. 3(b)–(d) are obtained in the first step. The each segmented image becomes an input image for our system in the next step.

In the second step, Figs. 3(e) and 3(f) are made from Fig. 3(b); subsequently, Figs. 3(g) and 3(h) are made from Fig. 3(c). The each segmentation was achieved according to a two-phase oscillatory response. The respective images in Figs. 3(d)–(h) have only one isolated image region. Since a non-oscillatory response, which corresponds to a fixed point, always occurs at the parameter values we set for an image with only one image region, we can obtain black images in Figs. 3(i)–(m). When we have black images in all flows, our algorithm is terminated.

4. Experimental Result of Image-Region Extraction

We considered dynamic image-region extraction for \((6 \times 6)\)-pixel a gray-level image with six isolated regions in Fig. 4(a). Based on the results of bifurcation analysis [4–6], we set the unfixed system parameters as \((k_r, \phi) = (0.885, 0.8)\) in which a fixed point or two- or three-phase periodic point appears for appropriate initial values to all chaotic neurons and the global inhibitor. The gray levels of pixels in each image region correspond to DC input values from 1.9 to 2.0 with the intervals of 0.02. In the first step, as seen in Fig. 4(b), the output images were obtained based on a three-phase oscillatory response. In the figure, the images sequentially appeared from the top-left to the bottom-right; their appearances in each line also started from the left. Subsequently, by removing redundant images from the output images in a time series, we picked out the three images in Fig. 4(c). In addition, because the output images were generated based on a three-phase 54-periodic point observed in a reduced model consisting of three neurons and a global inhibitor, the appearance period of each image region was also 54.

In the second step, each picked out image was inputted to our dynamic image segmentation system again. Based on two-phase periodic points, two segmented images were outputted as shown in Fig. 4(d) for every input images in Fig. 4(c). As the results, we finally obtained the six images. Therefore, it was demonstrated that our successive algorithm could dynamically extract all the image regions
from a gray-level image with six isolated image regions.

5. Concluding Remarks

We presented a successive algorithm using our dynamic image segmentation system to extract all image regions from a gray-level image with any number of isolated regions. We demonstrated that all the image regions in a gray-level image with six isolated image regions were extracted using our algorithm. In this paper, we dealt with the case that the gray levels of pixels in an image region were same. Since the assumption is not satisfied for general gray-level images, we will investigate synchronization phenomena in the responses of neurons corresponding to pixels in a gradational image region.

References


Figure 4: Results of dynamic image-region extraction based on our successive algorithm using our network model.

(a) (6 × 6)-pixel gray-level input image with six isolated regions.

(b) Output images performed by dynamic image segmentation in the first step.

(c) Picked out images from output images in Fig. 4(b).

(d) Picked out images from output images for input images shown in Fig. 4(c) in the second step.
Dynamical Micro-Bead Pattern Forming Using Laser Manipulation Techniques

Yoshio Tanaka†, Shogo Tsutsui‡, Mitsuru Ishikawa†, Hiroyuki Kitajima‡

†National Institute of Advanced Industrial Science and Technology (AIST), AIST Shikoku
2217-14 Hayashi-cho, Takamatsu, 761-0395, Japan
‡ Faculty of Engineering, Kagawa University,
2217-20 Hayashi-cho, Takamatsu, 761-0396, Japan
Email: yo-tanaka@aist.go.jp, kitaji@eng.kagawa-u.ac.jp

Abstract — Micro-bead arrays/patterns have potential for sensing tools in various scientific fields such as medicine, biochemistry, and so on. Laser trapping is an eminently suitable tool for dynamical handling of micro-beads. Here we demonstrate the fully automated assembly of micro-bead patterns by the control of an isolated-point-type laser trapping field in time domain. Several tens of micro-beads automatically form the specified patterns from random initial positions, and subsequent metamorphoses succeed.

1. Introduction

DNA chips using micro-spots of bio-molecules on a glass slide are valuable tools in genomics. Dynamic micro-arrays/patterns using mobile substrates, usually micro-beads coated with bio-molecules/chemicals, offer greater flexibility and have the potential to be used as sensing tools in not only scientific fields such as genomics and molecular biology but also engineering and medical fields such as lab-on-a-chips, drug discovery, diagnostics, and so on [1]. In the several demonstrated methods, laser manipulation techniques, namely optical tweezers, are eminently suitable for dynamic handling of micro-beads, since they have several advantages [2].

In this paper, we demonstrate the fully automated assembly of micro-bead patterns by the control of an isolated-point-type laser trapping field in time domain, that is, optical tweezers based on intelligent control techniques. First, we mention the principle of optical tweezers, and outline the experimental setup for the demonstrations. Secondly, we describe the algorithm for dynamical micro-bead patterning. Finally, the experimental results using the three-beam laser manipulation system demonstrate the performance of our approach.

2. Laser Manipulation

2.1. Principle

Laser trapping, first demonstrated in 1970 by Ashkin [3] and well known as optical tweezers [4], has been further extended to line-scanning [5], holographic [6], time-sharing [7], and others. Many of optical tweezers techniques except for generalized phase contrast (GPC) [8] are derived from strongly focused laser beams. Conventional optical tweezers, which have an isolated-point-type trapping field, use forces exerted by a strongly focused laser beam to trap small objects ranging in size from sub-micro-meters to several tens micrometers. The basic principles are straightforward for objects much smaller than the wavelength of light, or much larger [4]. Larger object acts as lens, refracting the rays of light and redirecting their photons’ momentum, \( \Delta P \), shown schematically in Fig. 1. The resulting recoil draws them toward the higher flux of photons near the focus. For example, in Fig. 1, the refraction of a pair of rays A and B of the trapping beam gives forces \( F_A \) and \( F_B \) whose vector sum \( F \) is always restoring for axial and transverse displacements of the particle from the trap focus \( f \) [9].

![Figure 1: Principle of the laser trapping of a micro particle. F: trapping force, f: focal point of a laser beam, \( \Delta P \): change of photons’ momentum.](image-url)
bead patterns, since they are simple to rapidly change multiple trapping positions. The laser scanning methods with high-reflective mirrors also enable the use of powerful trapping beams. Hence, we have developed a three-beam Time-Sharing Synchronized Scanning (3B-T3S) optical tweezers system, which can translate the patterns to the XY-plane at the desired Z-coordinate. Figure 2 shows the optical configuration and control system of the 3B-T3S optical tweezers system. Our 3B-T3S system is configured around an inverted microscope with high-NA oil-immersion objectives. The laser trapping sources are a continuous wave (CW) Nd:YAG laser (1064nm, 16W(max)) and a CW Cr:Forsterite laser (1250 nm,800mW(max)), and the former laser beam is split into s- and p-polarized beams by polarizers. The powers of these beams are adjusted by half-wave retardation plates. The three trapping beams are introduced coaxially into the microscope via shutters, lenses, 2-axis steering mirrors, a relay lens and the fluorescence port, and they are reflected upward by a dichroic mirror to the objective. The time-shared focal positions on the XY-plane are controlled by the 2-axis steering mirrors. The Z-coordinate of each time-shared beam is also controlled by the lens mounted on each PC-controlled linear stage which can move parallel to the optical axis. An image processor (Hitachi IP5005) digitizes the images (512 x 512 pixels) from a color CCD camera in real time for feature recognition. Software developed for the image processing and device control is executed by a personal computer (PC).

3. Dynamic Micro-bead Pattern

3.1. Control Algorithm

Dynamical micro-bead patternings and their transformation require the complex coordinated movements of a large number of beads. The movements are controlled with multi-beam optical tweezers under the bead detection and path generation algorithm. The Hough technique, which is the most common algorithm known to be robust under noisy conditions such as microscope images, is applied to the bead detection. The trajectories for parallel transportation without collisions are gradually generated by the following algorithm. First, the algorithm assigns pre-designed destinations, \( \mathbf{d}_{ik} \), to initial positions, \( \mathbf{i}_{ik} \), which are detected by the circular Hough transform. Second, new positions in the next time-step are gradually updated by

\[
\mathbf{p}_{ik}(t_{s+1}) = \mathbf{p}_{ik}(t_s + \tau) + \mathbf{\delta p}_{ik}, \quad s = 0, \ldots, n - 1,
\]

where \( \mathbf{\delta p}_{ik} \) is the step size vector of \( k \)-th bead, \( \mathbf{p}_{ik}(t_0) = \mathbf{i}_{ik} \), and \( \mathbf{p}_{ik}(t_n) = \mathbf{d}_{ik} \). Since we have found that an optimal step size for smooth transportation using optical tweezers techniques is less than the radius of the bead, each \( \mathbf{\delta p}_{ik} \) is determined by the following equations:

\[
\mathbf{\delta p}_{ik} = [\delta x_k, \delta y_k]^{\top} = (\mathbf{d}_{ik} - \mathbf{p}_{ik}) / n,
\]

\[
n \geq \max \left( \frac{d_{ik} - \mathbf{p}_{ik}}{r} \right), \quad k = 1, \ldots, M,
\]

where \( r \) is the radius of the bead, \( M \) is the number of destinations, and \( n \) is the smallest integer satisfying the Eq. (3). Under Eqs. (1)-(3), particle collisions cannot be avoided.

![Figure 2: Three-beam time-sharing synchronized scanning optical tweezers system. λ/2: half-wave retardation plate, PBS: polarizing beam splitter.](image)
since each bead is transported directly toward its destination along the linear trajectory. Therefore, we have adopted a rule for avoiding collisions; if a collision occurs in the next time-step, we simply exchange both the step size vectors and the destinations as shown in Fig. 3(a), and add one to the $k$-th collision counter $c_k$. If once collision occurs, the beads corresponding to this collision cannot reach their destinations at $t_n = n$, that is, $\delta p_k \neq 0$. Finally, therefore, the procedure for final approach to their destinations is necessary if the collision occurs. The positions for this final approach are updated by

$$
\mathbf{p}_k(t_{n+1}) = \mathbf{p}_k(t_n + \tau) = \mathbf{p}_k(t_n) + \delta^e \mathbf{p}_k, \quad s = n, \cdots, n + n_e, \quad (4)
$$

$$
\delta^e \mathbf{p}_k = (\mathbf{d}_k - n \mathbf{p}_k) / n_e, \quad (5)
$$

where $\delta^e \mathbf{p}_k$ is the step size vector for the final approach, and $n_e$ is the smallest integer satisfying the Eq. (6). The diagram of the algorithm is shown in Fig. 3(b). Note that the adequate assignment of destinations is important for less potential collisions. The assignment method for lattice pattern is proposed in our previous paper [2].

3.2. Demonstrations

On the basis of the control algorithm mentioned in the above section, we demonstrated the fully automated dynamical patterning of micro-beads. The sample was glass spheres (borosilicate, 2.5µm), and the beam powers were adjusted to roughly 200mW (beam1, beam2, and beam3 were 205mW, 212mW, and 156mW, respectively). The scanning dwell time for stable trapping was also adjusted to 0.01 seconds for each bead. Figure 4 shows a sequence of images recorded with the CCD camera showing the result of automated assembly of a dynamic bead array with specified pattern and its metamorphoses. First, the center positions of micro-beads dispersed in water droplets on the

$$
n_e = \max \left\{ \frac{d_{i,n} - n \mathbf{p}_k}{r} < 4 \max (c_k), \quad k = 1, \cdots, M \right\},
$$

where $\delta p_k$ is the step size vector for the final approach, and $n_e$ is the smallest integer satisfying the Eq. (6). The diagram of the algorithm is shown in Fig. 3(b). Note that the adequate assignment of destinations is important for less potential collisions. The assignment method for lattice pattern is proposed in our previous paper [2].

Figure 3: Algorithm for dynamical micro-bead patterning without collisions. (a): Schematic of collision avoiding, (b) diagram of updating bead positions.

Figure 4: Video frame sequence of dynamical forming of a pattern and its metamorphoses.
cover glass were detected by the circular Hough transform, then all twenty-four beads in the field of view were, automatically and simultaneously, trapped at the initially detected positions using 3B-T3S optical tweezers (Fig. 4(a)). Second, the twenty-four trapped beads were simultaneously transported to pre-designed destinations where three sets of eight beads formed a circle, while avoiding collisions each other (Fig. 4(b)-4(c)). Third, after the circle pattern was rotated in a counterclockwise direction (Fig. 4(d)), each set of eight beads was translated to Z-coordinate direction (Fig. 4(e)). Fourth, the circle was transformed to a stellar pattern (Fig. 4(e)-4(g)). Finally, the stellar pattern was transformed again to a circle after the rotation and translation (Fig. 4(h)).

In another demonstration shown in Fig. 5, thirty-three beads in the pipetted droplets were automatically patterned to form the Japanese character ‘katakana’. First, each laser beam which configured one T3S optical tweezers automatically trapped the set of eleven beads, and each set of eleven beads formed one character in the katakana syllabary, where three syllabics meant first author’s name ‘tanaka’ (Fig. 5(a)-5(d)). Second, syllabics were sequentially translated to Z-coordinate direction in order of ‘ta’-‘na’-‘ka’ (Fig. 5(e)). Finally, the pattern was transformed to the different syllabics which meant second author’s name ‘tsutsui’ (Fig. 5(h)), while avoiding collisions between each bead. Figures 5(f) and 5(g) show the intermediate status of beads through the transforming.

4. Conclusion

We have demonstrated the fully automated assembly of micro-bead patterns by the control of an isolated-point-type laser trapping field in time domain. Our approach can be applied to not only micro-beads but also biological materials and colloidal structures. Furthermore, the dynamical forming of micro-bead arrays with not only lattice [2] but also specified, arbitrary patterns may be valuable sensing tools for lab-on-a-chips and be actuators for micro-fluidic devices.

Acknowledgment

This work was partly supported by Grants-in-Aid for Scientific Research (C, #20560252) from the Japan Society for the Promotion of Science.

References

Support Vector Machines with Online Unsupervised Learning Method and its Application to surface-Electromyogram Recognition Problems

Hiroki Tamura†, Takeshi Yoshimatu† and Koichi Tanno†

†Department of Electrical and Electronic Engineering, University of Miyazaki
1-1 Gakuen Kibana Dai Nishi, Miyazaki-city, Miyazaki, 889-2192 Japan
Email: htamura@cc.miyazaki-u.ac.jp

Abstract— The support vector machine (abbr. SVM) is known as one of the most influential and powerful tools for solving classification and regression problems, but the original SVM does not have an online learning technique. Therefore, many researchers have introduced online learning techniques to the SVM. In this paper, we propose the new online unsupervised learning method using a technique of self-organized map for a SVM. Furthermore, the proposed method has a technique for the reconstruction of a SVM. We compare its performance with the original SVM, and also test our proposed method on surface-electromyogram (abbr. s-EMG) recognition problems with changes in the electrode position.

1. Introduction
s-EMG signals are detected over the skin surface and are generated by the electrical activity of the muscle fibers during contraction [1]. The load of s-EMG that rests upon the user for non-erosion is less than that of other biological signals. Therefore the application that uses s-EMG is actively developed. s-EMG recognition of using the conventional neural network is a method which learns the relation between s-EMG patterns and is reproduced using a neural network. In the recognition system, there are some problems that the s-EMG changes by the muscle wasting. In general, the muscle wasting will cause a decrease in the frequency of s-EMG and the tension. This is assumed to be the one due to the decrease at the muscle fiber conduction velocity [2]. Therefore, an additional learning function that corresponds to the muscle wasting is necessary for the s-EMG application. Support Vector Machine is known as one of the most influential and powerful tools for solving classification and regression problems [3]. But original SVM does not have online learning technique. Therefore, online learning techniques of SVM were proposed by many researchers [4][5][6][9][10].

In this paper, we propose the new online unsupervised learning method using self organized map (abbr. SOM) [7] for SVM. Furthermore, the proposed method has the technique of restructuring of SVM. Our proposed method has the advantage of small required memory size and small computational complexity. We test our proposed method to the s-EMG recognition problems with changes in the electrode position.

2. SVM with Online Unsupervised Learning Method
In this section, we introduce SVM and propose unsupervised learning method for SVM based on SOM.

2.1. Introduction of SVM
In this subsection, we summarize support vector machines for two-class problems. Assume the training sample \( S = \{(x_1,y_1), \ldots, (x_N,y_N)\} \) consisting of vectors \( x_i \in \mathbb{R}^d \) with \( i = 1, \ldots, N \), and each vector \( x_i \) belongs to either of the two classes. Thus it is given a label \( y_i \in \{-1,1\} \). The pair of \((w,b)\) defines a separating hyper-plane of equation as follows:

\[
(w, x) + b = 0
\]

However, Eq. (1) can possibly separate any part of the feature space, therefore one needs to establish an optimal separating hyper-plane (abbr. OSH) that divides \( S \) leaving all. The points of the same class are accumulated on the same side while maximizing the margin which is the distance of the closest point of \( S \). The closest vector \( x_i \) is called support vector (abbr. SV) and the OSH \( w, b \) can be determined by solving an optimization problem. We explain how to select candidates for SV. The solution of this optimization problem is given by the saddle point of the Lagrangian.

Maximize margin \( \frac{1}{2} (w \cdot w) \)

Subject \( y_i ((w \cdot x_i) + b) \geq 1 \)

To solve the case of nonlinear decision surfaces, the OSH is carried out by nonlinearly transforming a set of original feature vectors \( x_i \) into a high-dimensional feature space by mapping \( \Phi : x_i \rightarrow z_i \) and then performing the linear separation. However, it requires an enormous computation of inner products \( \langle \Phi(x) \cdot \Phi(x_i) \rangle \) in the high-dimensional feature space. Therefore, using a Kernel function which satisfies the Mercer’s theorem given in Eq. (2) significantly reduces the calculations to solve the nonlinear problems. In this paper, we used the Gaussian kernel given in Eq. (3) as the kernel function. The SVM decision function \( g(x) \) and output of SVM are as given in Eq. (4) and Eq. (5).
\[(\Phi(x) \cdot \Phi(x)) = K(x, x) \quad (2)\]
\[K(x, x) = \exp \left( -\frac{||x - x||^2}{2\sigma^2} \right) \quad (3)\]
\[g(x) = \sum_{j=0}^{N} w_iK(x_j, x) + b \quad (4)\]
\[O = \text{sign}(g(x)) \quad (5)\]

In this paper, we extended the standard SVM based on a pairwise coupling method for multiclass pattern recognition.

2.2. Proposed Method

The SOM algorithm was introduced by Kohonen [7]. SOM is a kind of artificial neural network that is trained using unsupervised learning. In the basic version, only one map winner at a time is activated corresponding to each input. And, the vector corresponding to the map vector that is called a reference vector was adjusted by learning rule. This model and its variants have been very successful in several real application areas. In this paper, the training vector is used as learned object instead of the reference vector. When SVM maps input data to a nonlinear space, training vectors have very importance action. However, the changing input data cannot be correctly mapped using SVM with the training vector at the beginning. The recognition mistake happens when the recognition data changes in the time series like the muscle wasting of s-EMG. To solve this problem, the training vectors are adjusted sequentially according to the SOM algorithm. The possibility of not satisfying the solution of the margin maximization that is the feature of SVM is caused by updating SOM algorithm. Therefore, this problem is solved by retraining SVM based on changing SV. Moreover, the number of training vectors must be limited for real problems of memory size. Then, we proposed unsupervised online learning method using SOM algorithm for SVM and restructure technique.

Let the input space be denoted by \(x_i \in \mathbb{R}^n\). \(x_i \notin \{i = 1, \ldots, N\}\) is the input vector without the label. The training vectors are included in kernel function, \(x_i\) with \(i = 1, \ldots, N\), belongs to either of the two classes. Thus these are given a label \(y_i \in \{-1, 1\}\). Each training vector has the same dimension of input space.

Next, the flows of our proposed method are shown.

**Step 1:** To find the smallest distance of the input vector \(x_i\) with the training vectors \(x_j\), the Euclidean distance between \(x_i\) and each \(x_j\) is computed (Fig.1.a).

Then the \(x_{\text{win}}\) with the smallest distance is selected as
\[\text{win} = \arg \min_{1 \leq i \leq N} ||x_{\text{win}} - x_i|| \quad (6)\]
This selected training vector is called \(x_{\text{win}}\), and \(d_{\text{win}}\) is the Euclidean distance between \(x_i\) and \(x_{\text{win}}\).

**Step 2:** The following processing (Step 3–4) are not done to \(x_{\text{in}}\) when the label of \(x_{\text{win}}\) is not the same as the label of the output result of SVM of \(x_{\text{in}}\).

**Step 3:** To find the smallest distance of the \(x_{\text{win}}\) with the training vectors \(x_j\), the Euclidean distance \(x_{\text{win}}\) and each \(x_j\) is computed (Fig.1.b).

Condition: \(x_j\) should be a different class from \(x_{\text{win}}\). The \(x_j\) which becomes the smallest distance is selected. This selected training vector is called \(x_{\text{other}}\), and \(d_{\text{other}}\) is the Euclidean distance between \(x_{\text{win}}\) and \(x_j\).

**Step 4:** If \(d_{\text{win}}\) is condition of rule of Eq. (7), \(x_{\text{win}}\) is updated according to the learning rule of Eq. (8) (Fig.1.c and Fig.1.d).

\[d_{\text{other}} \leq d_{\text{win}} \quad (7)\]
\[x_{\text{new}} = x_{\text{old}} + \eta(x_{\text{win}} - x_{\text{other}}) \quad (8)\]

Parameter \(\eta\) is update parameter. The idea of this rule is an idea near the adaptive resonance theory-like.

**Step 5:** Step1–4 are done to all input vector.

**Step 6:** Proposed method repeats \(\text{Num}\) cycles these processing (Step-1-5) using same inputs vector \(x_{\text{in}}\).

Figure 1. The flow of proposed method using SOM algorithm.

If SV changed after the update, SVM is restructured using the updated training vectors. Even if training vectors changes using the Step 1-5, maximizing the margin of SVM is kept from this restructuring processing.
3. Experiments
In this section, the system configuration for recognition experiments of forearm motions using EMG is explained. Next, the result of computer simulations is described.

3.1. Experimental Condition
S-EMG of each movement pattern is measured with electrode sensors, and the feature quantity is extracted from the s-EMG. The feature quantity is given to the recognition machine as an input and each movement pattern that generates s-EMG is presumed. The feature quantity uses minimum-maximum (abbr. min-max) values and integration values [8]. Paper [8] showed that technique of min-max values and integration values are more easy and superior to FFT processing. The sampling frequency of the measurement data is 1 KHz. And the band is from 0 KHz to 500 KHz.

3.2. Experiments of Forearm Muscles
We experimented on the effectiveness of the proposed method by the s-EMG recognition problem that the feature quantity changes by the electrode position. We compared proposed method performance with the original hard-margin SVM, soft-margin SVM (C-SVC) and \( k \)-NN method.

The experimental subjects are 4 healthy men (T.Y, K.F, S.Y, T.M). The subjects sit on a chair. The recognition experiment of the 6 motions pattern is conducted by using s-EMG obtained from four sensors set in the arm of the right hand (Fig.2). Moreover, the input given to the identification machine is eight inputs. The experiments are conducted for one day.

The experiment method, first acquires the training data from s-EMG concerning the movement of forearm. Next, SVM and C-SVC learn the relation between s-EMG and motion from the training data (the training vectors). And, each motion is identified 60 times. Next, the object moves the electrode position (sensor 1) by 2mm. And, additional unsupervised learning data (the input vector: each motion is 40 times) is obtained from each motion. Afterwards, test data for recognition rate calculation is identified 20 times of each motions. The experiments tested the measurement four times in total by moving the electrode position of 2mm, 5mm, 7mm and 10mm (Fig.3).

The base of proposed method is hard-margin SVM using Eq. (3). Gaussian kernel parameters of SVM were decided from the evaluation that used training data. Subject T.Y was 0.7, K.Y was 2.0, S.Y was 0.9, and T.M was 0.3. In these experiments, the value of parameter \( \eta \) typed two of 0.1 and 0.6. Moreover, the value of \( \text{Num} \) tested five types (\( \text{Num}=1, 5, 10, 15, \) and \( 20 \)).

3.3. Experimental Result
We showed the comparison experiment results (average result of four subjects) of the value of \( \text{Num} \) and the value of parameter \( \eta \) in Fig.4. Fig.4 shows that \( \text{Num}=15 \) and parameter \( \eta = 0.1 \) had the better performance. We compared proposed method with SVM and the simulation results (average result of four subjects) are Fig.5. Proposed method is better than original SVM.

Figure 2. Image figure of forearm motion. 6 motions patterns of forearm are conducted by using s-EMG obtained from four sensors set in the right arm.

Figure 3. s-EMG recognition problems with changes in the electrode position (2mm, 5mm, 7mm and 10mm).

Figure 4. The comparison experiment results of the value of \( \text{Num} \) and the value of parameter \( \eta \).

Figure 5. The comparison experiment results of SVM and proposed method (\( \text{Num}=15 \) and \( \eta = 0.1 \)).
In this paper, we proposed unsupervised learning method using SOM algorithm for SVM corresponding for s-EMG recognition problems. The experiment results showed that the proposed method was effective to s-EMG recognition problem with changes in the electrode position. SVM had improved by using our proposed method. Especially, the proposed method had the good performance for the sensor gap of several mm that often happened about a real problem. In future work, we will improve parameter decision method.

4. Conclusions

Next, each subject's results were shown in Fig.6. From Fig.6, our proposed method had the best performance in the sensor gap of 2mm. Moreover, our proposal method had the better results in four techniques in subject K.Y, S.Y and T.M. From these experiment results, we think that proposed method was effective in overcoming the s-EMG recognition problem with changes in the electrode position.

References

Kohonen Feature Map Probabilistic Associative Memory based on Weights Distribution and Area Neuron Increase and Decrease

Takahiro HADA† and Yuko OSANA†

†School of Computer Science, Tokyo University of Technology
1404-1 Katakura, Hachioji, Tokyo, 192-0982, Japan Email: osana@cs.teu.ac.jp

Abstract—In this paper, we propose a Kohonen Feature Map Probabilistic Associative Memory based on Weights Distribution and Area Neuron Increase and Decrease. This model is based on the conventional Kohonen Feature Map Probabilistic Associative Memory based on Weights Distribution. In the proposed model, the winner neuron is selected from the neurons in the Map Layer whose connection weights are similar to the input pattern, and the associations based on weights distribution are realized. Moreover, the weight distribution in the Map Layer can be modified by the increase and decrease of neurons in each area. This model has enough robustness for noisy input and damaged neurons. We carried out a series of computer experiments and confirmed the effectiveness of the proposed model.

1. Introduction

Recently, neural networks are drawing much attention as a method to realize flexible information processing. Neural networks consider neuron groups of the brain in the creature, and imitate these neurons technologically. Neural networks have some features, especially one of the important features is that the networks can learn to acquire the ability of information processing.

In the field of neural network, many models have been proposed. In these models, the learning process and the recall process are divided, and therefore they need all information to learn in advance. However, in the real world, it is very difficult to get all information to learn in advance, so we need the model whose learning process and recall process are not divided. As such model, Grossberg and Carpenter proposed the ART (Adaptive Resonance Theory)[1]. However, the ART is based on the local representation, and therefore it is not robust for damaged neurons. While in the field of associative memories, some models have been proposed[2]-[4]. Since these models are based on the distributed representation, they have the robustness for damaged neurons. However, their storage capacities are small because their learning algorithm is based on the Hebbian learning.

On the other hand, the Kohonen Feature Map (KFM) associative memory[5] has been proposed. Although the KFM associative memory is based on the local representation as similar as the ART[1], it can learn new patterns successively[6], and its storage capacity is larger than that of models in refs.[2]-[4]. It can deal with auto and hetero associations and the associations for plural sequential patterns including common terms[7]. Moreover, the KFM associative memory with area representation[8] has been proposed. In the model, one-to-many associations are realized by refractoriness of neurons. Moreover, the Kohonen Feature Map Probabilistic Associative Memory with Refractoriness based on Area Representation (KFMAM-R-AR)[10] has been proposed. In the model, one-to-many associations are realized by refractoriness of neurons. Moreover, the Kohonen Feature Map Probabilistic Associative Memory with Weights Distribution and Area Neuron Increase and Decrease (KFMPAM-WD)[11] has been proposed. It is based on the conventional KFM associative memory with area representation[8] and can realize probabilistic association for the training set including one-to-many relations.

In this paper, we propose a Kohonen Feature Map Probabilistic Associative Memory based on Weights Distribution and Area Neuron Increase and Decrease. This model is based on the conventional Kohonen Feature Map Probabilistic Associative Memory based on Weights Distribution[11]. In the proposed model, the winner neuron is selected from the neurons in the Map Layer whose connection weights are similar to the input pattern, and the associations based on weights distribution are realized. Moreover, the weight distribution in the Map Layer can be modified by the increase and decrease of neurons in each area. This model has enough robustness for noisy input and damaged neurons.

2. KFM Probabilistic Associative Memory based on Weights Distribution and Area Neuron Increase and Decrease

Here, we explain the proposed Kohonen Feature Map Probabilistic Associative Memory based on Weights Distribution and Area Neuron Increase and Decrease.

2.1. Structure

Figure 1 shows the structure of the proposed model. As shown in Fig.1, the proposed model has two layers; (1)
2.2. Learning Process

In the learning algorithm of the proposed model, the patterns are learned as follows:

1. In the network with the Map Layer composed of $x_{\text{max}} \times y_{\text{max}}$ neurons, the connection weights are initialized randomly. Here, $x_{\text{max}}$ is the initial number of neurons of a horizontal direction, and $y_{\text{max}}$ is the initial number of neurons of a vertical direction. In the initial state, $x_{\text{max}} \times y_{\text{max}}$ neurons are arranged at the coordinates $(0, 0), (1, 0), \cdots, (x_{\text{max}} - 1, 0), (1, 0), \cdots, (x_{\text{max}} - 1, y_{\text{max}} - 1)$.

2. The Euclidean distance between the learning vector $X(p)$ and the connection weights vector $W_i$, $d(X(p), W_i)$ is calculated.

$$d(X(p), W_i) = \sqrt{\sum_{k=1}^{M} (X_k(p) - W_{ik})^2} \quad (1)$$

where $M$ is the number of neurons in the Input/Output Layer. If $d(X(p), W_i) > \theta^*$ is satisfied for all neurons, the input pattern $X(p)$ is regarded as an unknown pattern. If the input pattern is regarded as a known pattern, go to (6).

3. The neuron which is the center of the learning area $c$ is determined as follows:

$$c = \arg \min_i d(X(p), W_i) \quad (2)$$

In this equation, the neuron whose Euclidean distance between its connection weights and the learning vector is minimum in the neurons which can be take areas without overlaps to the areas corresponding to the patterns which are already trained. In Eq.(2), $F$ is the set of the weight-fixed neurons, $d_c$ is the distance between the neuron $i$ and the weight-fixed neuron $z$. And $D_{ij}$ is the radius of the ellipse area whose center is the neuron $i$ for the direction to the neuron $i$ for the direction to the neuron $j$, and is given by

$$D_{ij} = \begin{cases} 
\sqrt{\frac{a_i^2 b_j^2}{b_i^2 + m_{ij}^2 a_j^2}} (m_{ij}^2 + 1), \\
\frac{a_i}{b_j}, \\
\frac{b_i}{a_j}, \\
(d_{ij}^* \neq 0 \text{ and } d_{ij}^* \neq 0) \\
(d_{ij}^* = 0), \\
(d_{ij}^* = 0). 
\end{cases} \quad (3)$$

where $a_i$ and $b_i$ are the long and short radius of the ellipse, and $m_{ij}$ is the slope of the line through the neurons $i$ and $j$, and is given by

$$m_{ij} = \frac{d_{ij}^*}{d_{ij}^*}, \quad (d_{ij}^* \neq 0). \quad (4)$$

4. If $d(X(p), W_i) > \theta^*$ is satisfied, the connection weights of the neurons in the ellipse whose center is the neuron $c$ are updated as follows:

$$W_i(t + 1) = \begin{cases} 
X(p), \\
W_i(t) + H(d_{ci})(X(p) - W_i(t)), \\
\frac{\theta_1^{\text{learn}}}{H(d_{ci})} < \theta_1^{\text{learn}} \text{ and } H(d_{ci}) < \theta_1^{\text{learn}}) \\
W_i(t), \quad \text{(otherwise)} 
\end{cases} \quad (5)$$

where $\theta_1^{\text{learn}}$ and $\theta_2^{\text{learn}}$ are the thresholds, and $i^*$ is the nearest weight-fixed neuron from the neuron $i$. $H(d_{ci})$ and $H(d_{ci})$ are the semi-fixed functions and are given by

$$H(d_{ci}) = \frac{1}{1 + \exp \left( \frac{d_{ci} - D}{\epsilon} \right)} \quad (6)$$

where $\epsilon$ is the steepness parameter of the function $H(d_{ci})$, and $D (1 < D)$ is the size of the neighborhood area. $d_{ci}$ is the normalized distance between the center neuron of the area $c$ and the neuron $i$, and is given by

$$d_{ci} = \frac{d_{ci}}{D_{ci}} \quad (7)$$

If there is no weight-fixed neuron,

$$H(d_{ci}) = 0 \quad (8)$$

is used.

5. The connection weights of the neuron $c$, $W_c$ are fixed.

6. (2)–(5) are iterated when a new pattern set is given.

2.3. Neuron Increase and Decrease

In the proposed model, the weights distribution in the Map Layer can be modified by the increase and decrease of neurons in each area.
2.3.1. Neuron Increase

In the proposed model, the neuron is added at the position corresponding to the neurons that exist in the initial Map Layer.

When the \( N_z \) neurons whose connection weights are same as that of the center neuron are in the area \( z \), the neuron is added as follows. Here, \( N_z^{ini} \) is the number of the neurons when the area \( z \) is generated, \( N_z^{min} \) is the minimum number of the neurons in the area \( z \), and \( N_z^{max} \) is the maximum number of the neurons in the area \( z \).

\[(1) \ N_z^{ini} \leq N_z < N_z^{max}\]

If \( N_z^{ini} \leq N_z < N_z^{max} \) is satisfied, the new neuron is added in the area \( z \).

The reference neuron \( j^* \) corresponding to the adding neuron \( j \) is given by

\[j^* = \left( (N_z - N_z^{ini}) \mod (N_z^{ini} - 1) \right) + 1\]  

(9)

where \( \mod \) shows remainder operation, and \( j^* \) shows the sequential serial number for the neurons except for the center neuron which exist when the area is generated (See Fig. 2). The coordinates of the new neuron is given by

\[x_j = x_{j^*} - \text{sgn}(x_{j^*}) \cdot \frac{S_{add}^j}{C_z}\]  

(10)

\[y_j = y_{j^*} - \text{sgn}(y_{j^*}) \cdot \frac{S_{add}^j}{C_z}\]  

(11)

where \( x_{j^*}, y_{j^*} \) are the coordinates of the reference neuron \( j^* \), and \( \text{sgn}(\cdot) \) is the sign function. \( C_z \) is the coefficient for the distance between neurons in the area \( z \), and is given by

\[C_z = \left\lceil \frac{N_z^{max} - N_z^{ini}}{N_z^{ini} - 1} \right\rceil + 1.\]  

(12)

\( S_{add}^j \) is

\[S_{add}^j = \left\lceil \frac{N_z - N_z^{ini}}{N_z^{ini} - 1} \right\rceil .\]  

(13)

where \( \lceil \cdot \rceil \) is the ceiling function and \( \lfloor \cdot \rfloor \) is the floor function.

Figure 2: Sequential Serial Number for Original Neurons in Area.

(2) \( N_z^{min} < N_z \leq N_z^{ini} \)

If the number of the neurons in the area \( N_z \) is smaller than \( N_z^{ini} \), the connection weights of the neuron in the area (the neuron which satisfy \( d_{zi} \leq D_z \)) and whose connection weights are same as that of the center neuron) are initialized randomly. The connection weights are updated in the neuron whose sequential serial number is \( N_z = 1 \).

2.4. Recall Process

In the recall process of the proposed model, when the pattern \( X \) is given to the I/O Layer, the output of the neuron \( i \) in the Map Layer, \( x_i^{map} \) is calculated by

\[x_i^{map} = \begin{cases} 1, & (i = r) \\ 0, & \text{otherwise} \end{cases}\]  

(14)

where \( r \) is selected randomly from the neurons which satisfy

\[\frac{1}{N_{io}} \sum_{k \in C} g(X_k - \theta_k) < \theta_{map}^i\]  

(15)

where \( N_{io} \) is the number of neurons which receive the input in the I/O Layer. \( g(\cdot) \) is given by

\[g(h) = \begin{cases} 1, & (|h| < \theta^i) \\ 0, & \text{otherwise} \end{cases}\]  

(16)

where \( \theta^i \) is the threshold, and \( \theta_{map}^i \) is the threshold of the neuron in the Map Layer.

In the proposed model, one of the neurons whose connection weights are similar to the input pattern are selected randomly as the winner neuron. So, the probabilistic association can be realized based on the weights distribution. For example, if the training patterns including the common term \( \{X, Y_1\}, \{X, Y_2\} \) are memorized, and the number of the neurons whose connection weights are similar to the pattern pair \( \{X, Y_1\} \) is larger than the number of the neurons whose connection weights are similar to the pattern pair \( \{X, Y_2\} \), then the probability that the pattern pair \( \{X, Y_1\} \) is recalled is higher than the probability that the pattern pair \( \{X, Y_2\} \) is recalled.

When the binary pattern \( X \) is given to the I/O Layer, the output of the neuron \( k \) in the I/O Layer \( x_k^{io} \) is given by

\[x_k^{io} = \begin{cases} 1, & (W_{ik} \geq \theta^i_b) \\ 0, & \text{otherwise} \end{cases}\]  

(17)

where \( \theta^i_b \) is the threshold of the neurons in the I/O Layer.

When the analog pattern \( X \) is given to the I/O Layer, the output of the neuron \( k \) in the I/O Layer \( x_k^{io} \) is given by

\[x_k^{io} = W_{ik}.\]  

(18)

3. Computer Experiment Results

Here, we show the computer experiment results to demonstrate the effectiveness of the proposed model.
Table 1: Relation between Area Size and Recall Times.

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
<th># of Neurons</th>
<th>Recall Times</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pattern 1</td>
<td>Pattern A</td>
<td>35 (3.2)</td>
<td>270 (3.3)</td>
</tr>
<tr>
<td></td>
<td>Pattern B</td>
<td>11 (1.0)</td>
<td>82 (1.0)</td>
</tr>
<tr>
<td></td>
<td>Pattern C</td>
<td>19 (1.7)</td>
<td>148 (1.8)</td>
</tr>
<tr>
<td>Pattern 2</td>
<td>Pattern D</td>
<td>23 (1.4)</td>
<td>208 (1.5)</td>
</tr>
<tr>
<td></td>
<td>Pattern E</td>
<td>17 (1.0)</td>
<td>152 (1.1)</td>
</tr>
<tr>
<td></td>
<td>Pattern F</td>
<td>17 (1.0)</td>
<td>140 (1.0)</td>
</tr>
</tbody>
</table>

Table 2: Relation between Area Size and Recall Times (After Resizing of Areas).

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
<th># of Neurons</th>
<th>Recall Times</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pattern 1</td>
<td>Pattern A</td>
<td>105 (9.5)</td>
<td>294 (9.8)</td>
</tr>
<tr>
<td></td>
<td>Pattern B</td>
<td>55 (5.0)</td>
<td>176 (5.9)</td>
</tr>
<tr>
<td></td>
<td>Pattern C</td>
<td>11 (1.0)</td>
<td>30 (1.0)</td>
</tr>
<tr>
<td>Pattern 2</td>
<td>Pattern D</td>
<td>35 (5.0)</td>
<td>176 (5.2)</td>
</tr>
<tr>
<td></td>
<td>Pattern E</td>
<td>70 (10.0)</td>
<td>290 (8.5)</td>
</tr>
<tr>
<td></td>
<td>Pattern F</td>
<td>7 (1.0)</td>
<td>34 (1.0)</td>
</tr>
</tbody>
</table>

3.1. Relation between Area Size and Recall Times

Here, we examined the relation between the area size and the recall times of the proposed model. In this experiment, six pattern pairs including 1-to-3 relations were memorized in the proposed model. Table 1 shows the relation between the area size and the recall times. And Table 2 shows the relation between the area size and the recall times after the resizing of areas. As shown in these tables, the proposed model can realize probabilistic association based on the weights distribution, and can modify the weights distribution by the increase and decrease of neurons in each area.

3.2. Robustness for Noisy Input/Damaged Neurons

Figure 3 shows the robustness for noisy input/damaged neurons in the proposed model. As shown in these figures, we confirmed that the proposed model has robustness for noisy input/damaged neurons.

4. Conclusions

In this paper, we have proposed the Kohonen Feature Map Probabilistic Associative Memory based on Weights Distribution and Area Neuron Increase and Decrease. We carried out a series of computer experiments and confirmed the effectiveness of the proposed model.

References

A reinforcer learning is a sub-area of machine learning concerned with how an agent ought to take actions in an environment so as to maximize some notion of long-term reward[1]. Reinforcement learning algorithms attempt to find a policy that maps states of the world to the actions the agent ought to take in those states.

The Temporal Difference (TD) learning is one of the reinforcement learning algorithm. The TD learning is a combination of Monte Carlo ideas and dynamic programming (DP) ideas. TD resembles a Monte Carlo method because it learns by sampling the environment according to some policy. TD is related to dynamic programming techniques because it approximates its current estimate based on previously learned estimates. The actor-critic method[2] is the method based on the TD learning, and consists of two parts; (1) actor which selects the action and (2) critic which evaluate the action and the state.

On the other hand, neural networks are drawing much attention as a method to realize flexible information processing. Neural networks consider neuron groups of the brain in the creature, and imitate these neurons technologically. Neural networks have some features, especially one of the important features is that the networks can learn to acquire the ability of information processing. The flexible information processing ability of the neural network and the adaptive learning ability of the reinforcement learning are combined, some reinforcement learning methods using neural networks are proposed[3]-[5].

In this paper, we propose the reinforcement learning method using Improved Kohonen Feature Map Probabilistic Associative Memory based on Weights Distribution (IKFMPAM-WD)[6]. The proposed method is based on the actor-critic method, and the actor is realized by the IKFMPAM-WD. The IKFMPAM-WD is based on the self-organizing feature map, and it can realize successive learning and one-to-many associations. The proposed method makes use of this property in order to realize the learning during the practice of task. We carried out a series of computer experiments, and confirmed the effectiveness of the proposed method in the path-finding problem.

2. Reinforcement Learning using IKFMPAM-WD

Here, we explain the proposed reinforcement learning method using IKFMPAM-WD[6].

2.1. Outline

In the proposed method, the actor in the actor-critic[2] is realized by the IKFMPAM-WD. In this research, the Input/Output Layer in the IKFMPAM-WD is divided into two parts corresponding to the state \( s \) and the action \( a \), and the actions for the states are memorized.

In this method, the critic receives the states which are obtained from the environment, the state is estimated and the value function is updated. Moreover, the critic outputs the Temporal Difference (TD) error to the actor. The IKFMPAM-WD which behaves as the actor (we call this “actor network”) is trained based on the TD error, and selects the action from the state of environment. Figure 1 shows the flow of the proposed method.

2.2. Actor Network

In the proposed method, the actor in the Actor-Critic[2] is realized by the IKFMPAM-WD.

2.2.1. Dynamics

In the actor network, when the state \( s \) is given to the Input/Output Layer, the corresponding action \( a \) is recalled. In the proposed method, as the state, (1) current and previous states (observations) and previous action and (2) current state (observation) are used. Moreover, the other action is also selected randomly (random selection), and the most desirable action from the recalled actions and the action selected in the random selection is chosen as the action finally.
When the actor network uses the current state $s_1(t)$ and the previous state and action $s_2(t)$, the state $s(t)$ is given by
\begin{equation}
 s(t) = (s_1(t) - s_2(t))^T. \tag{1}
\end{equation}

When the actor network uses the current state $s_1(t)$, the state $s(t)$ is given by
\begin{equation}
 s(t) = (s_1(t) - 1 \cdots - 1)^T. \tag{2}
\end{equation}

When the pattern $X(t)$ is given to the network, the output of the neuron $i$ in the Map Layer at the time $t$ $x_{i}^{map}(t)$ is given by
\begin{equation}
 x_{i}^{map} = \begin{cases} 
 1, & (i = r) \\
 0, & \text{(otherwise)}.
\end{cases} \tag{3}
\end{equation}

In the recall process, the input which does not receive the pattern is set to $-1$, and the winner neuron $r$ is selected randomly from the neurons which satisfy
\begin{equation}
 \frac{1}{N^{in}} \sum_{k \in X_{i}^{ror-1}} g(X_{k}(t) - W_{ik}) \geq \theta^{map} \tag{4}
\end{equation}

\begin{equation}
 g(b) = \begin{cases} 
 1, & (|b| > \theta^d) \\
 0, & \text{(otherwise)}. \tag{5}
\end{cases}
\end{equation}

where $N^{in}$ is the number of neurons which receive the input that not equal $-1$, $C$ is the set of neurons in the Input/Output Layer which receive the input that not equal $-1$, and $\theta^{map}$ is the threshold of the neuron in the Map Layer. And the input vector is given by
\begin{equation}
 X(t) = (s(t) - 1 \cdots - 1)^T \tag{6}
\end{equation}

The output of the neuron $k$ in the Input/Output Layer at the time $t$. $x_{k}^{in}(t)$ is given by
\begin{equation}
 x_{k}^{in}(t) = \begin{cases} 
 1, & 0.5 \leq W_{zk} \\
 0, & 0 \leq W_{zk} < 0.5 \\
 -1, & W_{zk} < 0. \tag{7}
\end{cases}
\end{equation}

\subsection{2.2.2. Learning}

The actor network is trained based on the TD error from the critic and the eligibility.

\begin{center}
\begin{figure}
\centering
\includegraphics[width=0.5\textwidth]{figure1.png}
\caption{Flow of Proposed Method.}
\end{figure}
\end{center}

\textbf{(1) Learning based on Reward}

In the learning based on the reward, the pair of the state $s(t)$ and the selected action $a(t)$ is memorized. The learning vector $X^{tr}(t)$ is given by
\begin{equation}
 X^{tr}(t) = (s(t) \ a(t))^T \tag{8}
\end{equation}

If the action which is recalled in the actor network only from the current state (observation) is selected, $s(t)$ in Eq.(2) is used. In the other cases, $s(t)$ in Eq.(1) is used.

\textbf{(a) When State and Action are not Stored}

If the pair of the state and the action are not stored and the positive reward is given, that pattern is trained as a new pattern. If the input vector is generated by Eqs.(2) and (8), the learning vector $X^{tr}(t)$ is generated again by Eqs.(1) and (8).

\textbf{(b) When State and Action are Stored}

If the pair of the state and the action are stored and the positive reward is given, the area corresponding to the pattern is expanded. If the pair of the state and the action are stored and the negative reward is given, the area corresponding to the pattern is reduced. Here, the area corresponding to the pattern is the area composed of the neurons which satisfy $d(X^{tr}(t), W) > \theta^d$. When the learning vector $X^{tr}(t)$ is generated by Eqs.(2) and (8), all corresponding areas are updated or reduced.

\textbf{(b-1) When Positive Reward is Given}

If the positive reward is given, the area corresponding to the learning vector $X^{tr}(t)$ is expanded. When the learning vector is generated by Eqs.(1) and (8), the size of the area whose center is the neuron $z$ is updated as follows:
\begin{equation}
 a_{z}^{new} = \begin{cases} 
 a_{z}^{old} + \Delta a_{z}^+, & a_{z}^{old} + \Delta a_{z}^+ \leq a_{max} \\
 a_{z}^{old}, & \text{otherwise} \tag{9}
\end{cases}
\end{equation}
\begin{equation}
 b_{z}^{new} = \begin{cases} 
 b_{z}^{old} + \Delta b_{z}^+, & b_{z}^{old} + \Delta b_{z}^+ \leq b_{max} \\
 b_{z}^{old}, & \text{otherwise} \tag{10}
\end{cases}
\end{equation}

where $\Delta a_{z}^+$ is the increment of $a_z$, $\Delta b_{z}^+$ is the increment of $b_z$, $a_{max}$ is the maximum of $a_z$, and $b_{max}$ is the maximum of $b_z$.

When the learning vector is generated by Eqs.(2) and (8), the size of the area whose center is the neuron $z$ is updated as follows:
\begin{equation}
 a_{z}^{new} = \begin{cases} 
 a_{z}^{old} + \Delta a_{z}^+, & a_{z}^{old} + \Delta a_{z}^+ \leq a_{max} \\
 a_{z}^{old}, & \text{otherwise} \tag{11}
\end{cases}
\end{equation}
\begin{equation}
 b_{z}^{new} = \begin{cases} 
 b_{z}^{old} + \Delta b_{z}^+, & b_{z}^{old} + \Delta b_{z}^+ \leq b_{max} \\
 b_{z}^{old}, & \text{otherwise} \tag{12}
\end{cases}
\end{equation}

where $\Delta a_{z}^+ (\Delta a_{z}^+ \leq \Delta a_{z})$ is the increment of $a_z$, and $\Delta b_{z}^+ (\Delta b_{z}^+ \leq \Delta b_{z})$ is the increment of $b_z$. 

- 186 -
When the area size is updated, the connection weights are updated as follows:

\[ W_i(t + 1) = \begin{cases} \mathbf{W}_i(t), & d_{zi} \leq D_{zi} \\ \mathbf{W}_i(t), & \text{otherwise} \end{cases} \]  
(13)

where \( d_{zi} \) is the distance between the neuron \( i \) and the neuron \( z \).

(b-2) When Negative Reward is Given

If the negative reward is given, the area corresponding to the learning vector \( X^s(t) \) is expanded. When the learning vector is generated by Eqs. (1) and (8), the size of the area whose center is the neuron \( z \) is updated as follows:

\[ a_z^{(new)} = \begin{cases} 0, & a_z^{(old)} - \Delta a_1 < 0 \\ a_z^{(old)} - \Delta a_1, & \text{otherwise} \end{cases} \]  
(14)

\[ b_z^{(new)} = \begin{cases} 0, & b_z^{(old)} - \Delta b_1 < 0 \\ b_z^{(old)} - \Delta b_1, & \text{otherwise} \end{cases} \]  
(15)

where \( \Delta a_1 \) is the decrement of \( a_z \), and \( \Delta b_1 \) is the decrement of \( b_z \).

When the learning vector is generated by Eqs. (2) and (8), the size of the area whose center is the neuron \( z \) is updated as follows:

\[ a_z^{(new)} = \begin{cases} 0, & a_z^{(old)} - \Delta a_2 < 0 \\ a_z^{(old)} - \Delta a_2, & \text{otherwise} \end{cases} \]  
(16)

\[ b_z^{(new)} = \begin{cases} 0, & b_z^{(old)} - \Delta b_2 < 0 \\ b_z^{(old)} - \Delta b_2, & \text{otherwise} \end{cases} \]  
(17)

where \( \Delta a_2 \) is the decrement of \( a_z \), and \( \Delta b_2 \) is the decrement of \( b_z \).

If \( a_z \) and \( b_z \) become to 0, the area whose center is \( z \) disappears and the neuron \( z \) is unlocked.

When the area size is updated, the connection weights are updated as follows:

\[ W_i(t + 1) = \begin{cases} R, & P_i^{(after)} < D_{zi} \leq D_i^{(before)} \\ W_i(t), & \text{otherwise} \end{cases} \]  
(18)

where \( R \) is small random value.

(2) Learning based on Eligibility

In the learning based on the eligibility, the areas corresponding to the state \( s_1 \) which satisfy

\[ e(s_1) < \theta_e \]  
(19)

are reduced. Here, \( e(s_1) \) is the eligibility for the state \( s_1 \) and \( \theta_e \) is the threshold for the eligibility.

The area size is updated as follows:

\[ a_z^{(new)} = \begin{cases} 0, & a_z^{(old)} - \Delta a_1 < 0 \\ a_z^{(old)} - \Delta a_1, & \text{otherwise} \end{cases} \]  
(20)

\[ b_z^{(new)} = \begin{cases} 0, & b_z^{(old)} - \Delta b_1 < 0 \\ b_z^{(old)} - \Delta b_1, & \text{otherwise} \end{cases} \]  
(21)

where \( \Delta a_1 \) is the decrement of \( a_z \), and \( \Delta b_1 \) is the decrement of \( b_z \).

When the area size is updated, the connection weights are updated by Eq.(18).

2.3. Reinforcement Learning using IKFMPAM-WD

The flow of the proposed reinforcement learning method using IKFMPAM-WD is as follows:

(1) The initial values of weights in the actor network are chosen randomly.

(2) The agent observes the environment \( s(t) \), and the actor \( a(t) \) is selected by the actor network or the random selection.

(3) The state \( s(t) \) transits to the \( s(t+1) \) by action \( a(t) \).

(4) The critic receives the reward \( r(s(t+1)) \) from the environment \( s(t+1) \), and outputs the TD error \( \delta \) to the actor.

\[ \delta = r(s(t+1)) + \gamma V(s(t+1)) - V(s(t)) \]  
(22)

where \( \gamma (0 < \gamma < 1) \) is the decay parameter, and \( V(s(t)) \) is the value function for the state \( s(t) \).

(5) The eligibility \( e(s) \) is updated.

\[ e(s) = \begin{cases} \gamma \lambda e(s) & \text{(if } s \neq s(t+1)) \\ \gamma \lambda e(s) + 1 & \text{(if } s = s(t+1)) \end{cases} \]  
(23)

where \( \gamma (0 < \gamma < 1) \) is the decay parameter, and \( \lambda \) is the trace decay parameter.

(6) All values for states \( V(s) \) are updated based on the eligibility \( e(s) \) \((s \in S)\).

\[ V(s) = V(s) + \xi \delta e(s) \]  
(24)

where \( \xi (0 < \xi \leq 1) \) is the learning rate.

(7) The connection weights in the actor network are updated based on the TD error and the eligibility.

(8) Back to (2).

3. Computer Experiment Results

Here, we show the computer experiment results to demonstrate the effectiveness of the proposed method.
3.1. Path-Finding Problem

We applied the proposed method to the path-finding problem. In this experiment, a agent moves from the start point (S) to the goal point (G). The agent can observe the states of three cells in the lattice, and can move forward/left/right. As the positive reward, we gave 3 when the agent arrives at the goal and 2 when the agent moves. And as the negative reward, we gave −1 when the agent hits against the wall.

Figure 2 shows the transition of number of steps from the start to the goal. In Fig.2, the trained routes (arrows) are also shown. Figure 3 shows an example of the trained relation between the state and the action. And Fig.4 shows the area size transition in the same trail. As shown in this figure, some areas are expanded and the other areas are reduced or disappeared.

3.2. Learning in Other Environment

Here, the network which was learned in the Map 1 or 2 was used in the Map 3. Figure 5 shows the transition of number of steps from the start to the goal in the Map 3 using the network trained in the Map 1 or 2. As shown in this figure, the proposed method can use the knowledge which was trained in the similar environment.

Figure 4: Transition of Area Size.

Figure 5: Transition of Steps (Map 3).

4. Conclusions

In this paper, we have proposed the reinforcement learning method using IKFMPAM-WD. The proposed method is based on the actor-critic method, and the actor is realized by the IKFMPAM-WD. We carried out a series of computer experiments, and confirmed the effectiveness of the proposed method in path-finding problem.

References
Abstract—Glial cells have known to exist in the brain. They watch brain’s state by signal transmitting each other. Thus, many researchers have taken notice to relationships of glial cells and neurons.

In this paper, we propose a Multi-Layer Perceptron (MLP) having neuro-glia network. We give biological features of glial cells to the standard MLP. We consider that we can obtain good performance by glial cells and neurons’ having the interrelation. By computer simulations for solving Two Spiral Problem (TSP), we confirm that the proposed MLP having neuro-glia network gains better performance than the conventional MLP.

2. Multi-Layer Perceptron with Glial Network

In this section, we explain a concept of MLP having neuro-glia network.

2.1. Multi-Layer Perceptron

MLP is the most famous feed forward neural network. This network is used for pattern recognition, pattern classification, and other tasks. MLP has some layers, it has mainly input layer, hidden layer, and output layer. Generally, it is known that MLP can solve a more difficult task if the number of layer or neuron is increased. We consider MLP which is composed of four layers (one input layer, two hidden layers and one output layer), and MLP has the glial network in the second layer of the hidden layer. The proposed MLP having neuro-glia network structure (connected 2-20-40-1) is shown in Fig. 1.

2.2. Neuron Updating Rule

The updating rule of neurons in the input layer, the first hidden layer and the output layer is described by Eq. (1) which is conventional updating rule.

\[ x_i(t+1) = f \left( \sum_{j=1}^{n} w_{ij}(t)x_j(t) - \theta_i(t) \right) \] (1)
In the MLP having neuro-glia network, random oscillation is added to neurons in the second hidden layer. This neuron’s updating rule is following as Eq. (2).

\[ x_i(t + 1) = f \left( \sum_{j=1}^{n} w_{ij}(t)x_j(t) - \theta_i(t) + \alpha \Psi_i(t) \right), \tag{2} \]

where \( x \) : input or output, \( w \) : weight parameter, \( \theta \) : threshold, \( \Psi \) : oscillation of glial network, \( \alpha \) : amplitude of noise and \( f \) : output function. And we use sigmoid for the output function as Eq. (3).

\[ f(\alpha) = \frac{1}{1 + e^{-\alpha}} \tag{3} \]

### 2.3. Generating Oscillation by MLP with Glial Network

In the biological neural network, it is known that the glial cells affect to the neighbor neurons over a wide range by propagating in the network [6]. We considered that the glial network gives good influence to learning of neural networks. We proposed glial network connected to MLP [4] as shown Eq. (4).

\[ \Psi_i(t) = \sum_{k=n}^{m} \beta^k \psi_{i+k}(t - |k|), \tag{4} \]

where \( \psi \) : uniformed random noise, \( \beta \) : attenuation parameter and \( k \) : the propagating range in the glial network.

### 2.4. Generating Oscillation by MLP Having Neuro-Glia Network

In the glial network has function of Eq. (4). Moreover, we notice glial cell’s function which cool out exciting neurons [3]. In order to realize phenomena, we add random oscillation to neurons by using Eq. (5).

\[ \Psi_i(t) = \sum_{k=n}^{m} \beta^k \psi_{i+k}(t - |k|) \{ 0.5 - O_{i+k}(t - 1) \}, \tag{5} \]

\( O \) is the second hidden layer’s neuron output. Glial cells are watching connecting neuron’s output. They decide next output’s amplitude by \( k \) range existing glial cells’ output and before output of connecting neuron’s. Random oscillation produced by the uniform random function propagates in the glial network and watching neuron’s output as shown in Fig. 2.

### 3. Oscillation of Glial Network

In this section, we show that the conceptual glial network. We show three kinds of generating oscillation by glial cells.

#### 3.1. Generating by Glial Cells

Glial cells generate uniformed random oscillation, thus all glial cells have different oscillation. We shows that neighborhood glial cells generate oscillation in the Fig. 3. We mark glial output of each time with dot and we plot moving average lines. In this figure, each moving average line has different curve. We consider that two glial cells do not have a relationship in this case.

![Figure 3: Uniformed random oscillation of two glial cells.](image)

#### 3.2. Generating by MLP with Glial Network

Figure 4 is a glial cells’ output when generating oscillations pass in the glial network. The oscillations of Fig. 4 become biased oscillation by passing in the glial network. Moreover, moving average line of each changing oscillation become very similar curve. Because neighbor glial cells are influenced each other by passing in the glial network.

![Figure 4: Propagating random oscillation in the glial network.](image)

#### 3.3. Generating by MLP Having Neuro-Glia Network

Figure 5 show that two kinds of oscillation using to MLP having neuro-glia network. These oscillations are similar to oscillation of Fig. 4. However, these vibrating positions...
are different in Fig. 5, because amplitude of oscillation is influenced by the second hidden layer neuron in the MLP having neuro-glia network. We consider that using MLP having neuro-glia network can give to each neuron unit for the adequate oscillation.

\[ E = \frac{\sum \text{False}}{N}, \]  

where \( N \) denotes the number of learning point.

### 4.1. Learning 98 Points

We prepare 98 data of two spirals. The number of learning points is fixed as 500000. We use propagating range \( m = 5 \), attenuation parameter \( \beta = 0.8 \) and learning coefficient \( \eta = 0.05 \).

Table 1 shows simulation result of each MLP learn to 98 points. “Err = 0%” means the number of times which has memorized all the points. “Err < 10%” means the number of False is lower 10%. “AVG of Err” means each MLP learn the average of an error when each MLP finishes learning.

From this table, we can see that the MLP having neuro-glia network shows the highest result of all conditions. In this result, we consider that MLP MLP having neuro-glia network effective in the learning of two-spiral.

### 4.1.1. Learning 98 Points

We prepare 98 data of two spirals. The number of learning points is fixed as 500000. We use propagating range \( m = 5 \), attenuation parameter \( \beta = 0.8 \) and learning coefficient \( \eta = 0.05 \).

Table 1 shows simulation result of each MLP learn to 98 points. “Err = 0%” means the number of times which has memorized all the points. “Err < 10%” means the number of False is lower 10%. “AVG of Err” means each MLP learn the average of an error when each MLP finishes learning.

From this table, we can see that the MLP having neuro-glia network shows the highest result of all conditions. In this result, we consider that MLP MLP having neuro-glia network effective in the learning of two-spiral.

### 4.1.2. Learning 130 Points

We prepare 130 data of two spirals. The number of learning points is fixed as 500000. We use propagating range \( m = 5 \), attenuation parameter \( \beta = 0.8 \) and learning coefficient \( \eta = 0.05 \).

Table 1 shows simulation result of each MLP learn to 130 points. “Err = 0%” means the number of times which has memorized all the points. “Err < 10%” means the number of False is lower 10%. “AVG of Err” means each MLP learn the average of an error when each MLP finishes learning.

From this table, we can see that the MLP having neuro-glia network shows the highest result of all conditions. In this result, we consider that MLP MLP having neuro-glia network effective in the learning of two-spiral.
range $m = 5$, attenuation parameter $\beta = 0.8$ and learning coefficient $\eta = 0.05$.

Table 2 shows result of each MLP learn to 130 points. MLP having neuro-glia network shows the highest percentage for all conditions. We can get a good result in the 98 points learning. However, we can see that clearly MLP having neuro-glia network is competitive solving high nonlinear problem in this result. We consider that this performance which finds nearly optimal solution is one of important characteristics for solving high nonlinear problems.

Table 2: Performance of learning to 130 points by using each MLP

<table>
<thead>
<tr>
<th></th>
<th>Err = 0%</th>
<th>Err &lt; 10%</th>
<th>AVG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neuro-Glia</td>
<td>27%</td>
<td>60%</td>
<td>13%</td>
</tr>
<tr>
<td>Glial Network</td>
<td>3.3%</td>
<td>30%</td>
<td>24%</td>
</tr>
<tr>
<td>Random Noise</td>
<td>0.0%</td>
<td>20%</td>
<td>26%</td>
</tr>
<tr>
<td>Conventional</td>
<td>6.7%</td>
<td>17%</td>
<td>29%</td>
</tr>
</tbody>
</table>

4.2. Classification of Two Spirals

Figure 8 is typical result of learning 130 points. MLP with glial network, MLP adding random noise and conventional MLP are fall into the local minimum at about 250000 learning point. However MLP having neuro-glia network escape out from the local minima. In this figure, we can see learning convergence of MLP MLP having neuro-glia network is very fast.

Figure 8: Error curve by four MLP networks for 130 points.

5. Conclusions

In our study, we have proposed MLP having neuro-glia network. This network gave random oscillations to the second hidden layer’s neuron. The amplitude of oscillation changes by neuron’s output and this random oscillation propagates to other neurons. We confirmed that MLP having neuro-glia network gains better performance than the glial network connected to MLP, MLP adding random noise and the conventional for solving TSP.

References

Canonical Particle Swarm Optimization System

Kenya Jin’no, Takuya Shindo

Electrical & Electronics Engineering,
Nippon Institute of Technology
Miyashiro, Minami-Saitama, Saitama, 345-8501, Japan
Email: jinno@nit.ac.jp

Abstract—A particle swarm optimization (PSO) system is one of the powerful systems for solving global optimization problems. The PSO algorithm can search an optimal value of a given evaluation function quickly compared with other proposed meta-heuristics algorithms. The conventional PSO system contains some random factors, therefore, the dynamics of the system can be regarded as stochastic dynamics. In order to analyze the dynamics rigorously, some papers pay attention to deterministic PSO systems which does not contain any stochastic factors. According to these results, the eigenvalues of the system impinge on the dynamics of the particles. Depending on the parameter, the searching ability of the deterministic PSO is decreased. In order to overcome this, we propose a canonical deterministic PSO which can control its eigenvalues easily, and can improve the searching ability. We will confirm relation between the eigenvalues and the searching ability of the optimal value from some numerical experiments.

1. Introduction

Searching for an optimal value of a given evaluation function of various problems is very important in engineering fields. In order to solve such optimization problems speedily, various heuristic optimization algorithms have been proposed. Particle swarm optimization (PSO), which was originally proposed by J. Kennedy and R. Eberhart [1],[2], is one such heuristic algorithm. The PSO algorithm is a useful tool for optimization problems[3]-[6].

The original PSO is described as

\[ v_{j}^{t+1} = wv_{j}^{t} + c_{1}r_{1}(p_{best_{j}}^{t} - x_{j}^{t}) + c_{2}r_{2}(g_{best}^{t} - x_{j}^{t}) \]  \( (1) \)

\[ x_{j}^{t+1} = x_{j}^{t} + v_{j}^{t+1} \]  \( (2) \)

where \( w \geq 0 \) is an inertia weight coefficient, \( c_{1} \geq 0 \), and \( c_{2} \geq 0 \) are acceleration coefficients, and \( r_{1} \in [0,1] \), and \( r_{2} \in [0,1] \) are two separately generated uniformly distributed random numbers in the range \([0,1] \). \( x_{j} \in \mathbb{R}^{N} \) denotes the location of the \( j \)-th particle on the \( t \)-th iteration in the \( N \)-dimensional space, and \( v_{j} \in \mathbb{R}^{N} \) denotes a velocity vector of the \( j \)-th particle on the \( t \)-th iteration. \( p_{best_{j}} \in \mathbb{R}^{N} \) means the location that gives the best value of the evaluation function of the \( j \)-th particle on the \( t \)-th iteration. \( g_{best} \in \mathbb{R}^{N} \) means the location which gives the best value of the evaluation function on the \( t \)-th iteration in the swarm.

The particles in the swarm fly through the \( N \)-dimensional space according with Eqs. (1) and (2). Each particle shares information of a current optimal value of the evaluation function and its corresponding location of the best particle. Also, each particle memorizes its record of the best evaluation value and its best location. On the basis of such information, the moving direction and velocity are calculated by Eq. (1). Namely, all particles will move toward a coordinate that gives the current best value of the evaluation function.

In such PSO system, the parameters play very important role to the searching ability. Therefore, many researchers study about adequate parameters selecting[7]. The searching ability of such PSO depends on the inertia weight coefficient, and the acceleration coefficients. Since the acceleration coefficients are multiplied by a random number, the system can be regarded as a stochastic system. The rigorous analysis of such stochastic system is difficult. In order to analyze the dynamics of such PSO, M. Clerc, and J. Kennedy proposed a simple deterministic PSO system, and analyzed its dynamics theoretically[8]. The simple deterministic PSO system does not contain stochastic factors, namely, the random coefficients have been omitted from the original PSO system. The analysis of such a deterministic PSO is very important for determining the effective parameters of the standard PSO[8]-[9]. We proposed a deterministic PSO and analyzed the searching ability of the optimal value of the given benchmark functions[10]. According to the results, the dynamics depends on the eigenvalues of the system[10]. The eigenvalues depend on the inertia weight coefficient, and the acceleration coefficients. In order to control the eigenvalues simply, we propose a canonical deterministic PSO.

2. Canonical deterministic PSO

The simplicity acceleration coefficients of the deterministic PSO system can be described as

\[ p_{j}^{t} = \gamma p_{best_{j}}^{t} + (1 - \gamma)g_{best}^{t} \]  \( (3) \)

\[ \gamma = \frac{c_{1}}{c_{1} + c_{2}} \]  \( (4) \)
where $p_j'$ can be regarded as a desired fixed point.

The parameter $\gamma$ controls the mixture rate of the local best and the global best.

Since each dimension variable of the particle is independent, we can consider one dimensional case without loss of generality. Therefore, we consider one dimensional system hereafter. For one dimensional deterministic PSO can be transformed into the following matrix form:

$$
\begin{bmatrix}
  v_{j+1} \\
  y_{j+1}
\end{bmatrix} =
\begin{bmatrix}
w & -c \\
  w & 1 - c
\end{bmatrix}
\begin{bmatrix}
v_j \\
y_j
\end{bmatrix}
$$

(5)

where $y_j = x_j - p_j'$ and $c = c_1 + c_2$.

Note that this system does not contain stochastic factors, therefore, this system can be regarded as a deterministic system.

The dynamics of the deterministic PSO is governed by the eigenvalues of the matrix in Eq. (5). The eigenvalue $\lambda$ is given as

$$
\lambda = \frac{1 - c + w}{2} \pm \sqrt{\frac{(1 - c + w)^2 - 4w}{2}}.
$$

(6)

If the eigenvalue $\lambda$ is complex conjugate number, the system exhibits remarkable searching ability. Since the system is discrete-time system, the damping factor $\Delta$ and the rotation angle $\theta$ on each iteration can be derived by its complex eigenvalues as

$$
\Delta = \sqrt{w}
$$

(7)

$$
\theta = \arctan\frac{\sqrt{4w - (1 - c + w)^2}}{1 - c + w}
$$

(8)

The trajectory in the phase space exhibits spiral motion as shown in Fig. 1 whose desired fixed point is the origin is assumed. In the cases of Fig. 1(c), the range of the variable $x$ becomes wide, and the system can not search around the origin.

Based on the eigenvalue $\lambda$ which is expressed in Eq. (6), we can calculate a translation matrix $P$. By using this translation matrix $P$, we derive the following matrix.

$$
\begin{bmatrix}
  \delta & -\omega \\
  \omega & \delta
\end{bmatrix} = P^{-1}
\begin{bmatrix}
w & -c \\
  w & 1 - c
\end{bmatrix} P
$$

(9)

where

$$
P = \sqrt{\frac{2}{4w - (1 - c + w)^2}}
\begin{bmatrix}
1 & -\frac{w - c - 1}{w} \frac{\sqrt{4w - (1 - c + w)^2}}{2c}
\end{bmatrix}
$$

Here, we consider the following coordinate transformation.

$$
\begin{bmatrix}
v' \\
y'
\end{bmatrix} = P^{-1}
\begin{bmatrix}
v \\
y
\end{bmatrix}
$$

(10)

By using this coordinate transformation of Eq. (10), we can derive the following canonical form:

$$
\begin{bmatrix}
v'_{j+1} \\
y'_{j+1}
\end{bmatrix} =
\begin{bmatrix}
\delta & -\omega \\
\omega & \delta
\end{bmatrix}
\begin{bmatrix}
v'_j \\
y'_j
\end{bmatrix}
$$

(11)

where $\tilde{y}'_j = x'_j - \tilde{p}'_j$.

The new coordinate $(\tilde{y}, \tilde{v})$ can be regarded as a normalized coordinate. Each dimension component in $\tilde{v}$ and $\tilde{y}$ is independent, therefore The behavior of the system is governed by the eigenvalues of the canonical form of Eq. (11). The eigenvalues $\tilde{\lambda}$ of the system is derived as

$$
\tilde{\lambda} = \delta + j\omega
$$

(12)

The system of (11) is a discrete-time system. For the system to become stable, the eigenvalues must exist within the unit circle on the complex plane. Therefore, the system is said to be stable if and only if the following condition is satisfied.

$$
\delta^2 + \omega^2 < 1
$$

(13)

If the parameters satisfy above condition, the eigenvalues exist within the unit circle. The particle converges to a fixed
point $p_j$. In generally, the fixed point $p_j$ is varied with time steps. Therefore, the trajectory exhibits a complex motion. The damping factor $\Delta$ is derived as

$$\Delta = \sqrt{\delta^2 + \omega^2}$$ (14)

Note that if the parameters satisfy the condition (13), the damping factor $\Delta$ satisfy the following.

$$|\Delta| < 1$$ (15)

The rotational angle $\theta$ on each iteration is given as

$$\theta = \arctan \frac{\omega}{\delta}$$ (16)

By using the damping factor and the rotation angle, Eq. (11) can be transformed into the following.

$$\begin{bmatrix} v_{i+1}^x \\ v_{i+1}^y \end{bmatrix} = \Delta \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} v_i^x \\ v_i^y \end{bmatrix}$$ (17)

The trajectory of the canonical deterministic PSO as shown in Fig. 2 whose desired fixed point is the origin is assumed. Comparing the case of Fig. 1(c), the trajectory of Fig. 2(c) does not expand. We think this property gives an effective influence for searching. In the following section, we confirm this fact by some numerical experiments by using some benchmark functions.

3. Simulation

In this section, we investigate the relation between the eigenvalues and the searching ability, we carry out some numerical simulations. We use two well-known benchmark problems. Each objective function consists of 10-dimensional variables. For each simulation, the population size of the swarm is 10.

The parameter $\gamma$ controls the mixture rate between the local best and the global best. $\gamma = 1.0$ denotes that the system uses only the local best information, and $\gamma = 0.0$ means that it uses only the global best information. The previous simulation results indicate the information of the global best is important[11]. We apply $\gamma = 0.0$ for the numerical simulation hereafter.

We confirm the relation between the rotation angle and the convergence property. First, we observe the convergence property when the rotation angle $\theta$ is varied from 0-degree to 180-degree in a period 10-degree. The simulation results are shown in Fig. 3. The vertical axis denotes the mean error from the optimal value with the searched value, the horizontal axis denotes the rotation angle $\theta$. Each curve corresponds to each damping factor; $\Delta = 0.95, 0.90, 0.70, 0.50, 0.30$. Each data is the average of the experimental results with ten trials.

These results indicate that the characteristic of the convergence properties are depended on the damping factor. As the damping factor changes to large, the characteristic of the convergence property is changed. According to the numerical simulation, when the damping factor sets as $\Delta = 0.95$, the canonical deterministic PSO exhibits the most effective performance.

The result of the deterministic PSO is shown in Fig. 4 to compare with the result of the canonical deterministic PSO. In the case of the deterministic PSO, the decline of the performance is observed around the 180 degree in the rotation angle comparing with the case of the canonical deterministic PSO. Therefore, we can say the proposed canonical deterministic PSO exhibits better performance.

4. Conclusions

In this article, we analyzed the convergence performance of the canonical deterministic PSO system. The canonical deterministic PSO system does not contain the stochastic factor. We confirm the relation between the eigenvalue and the convergence property by using the damping factor and the rotation angles. The results suggest these parameters have the optimal value. On the basis of this result, we will construct an effective stochastic PSO system.

References

Figure 2: The trajectory of the canonical deterministic PSO in the phase space $\tilde{y} - \tilde{v}$.


Particle Swarm Optimization with Novel Concept of Complex Network

Haruna Matsushita†, Yoshifumi Nishio‡ and Toshimichi Saito†

†Dept. of Electrical and Electronics Engineering, Hosei University
3–7–2 Kajimo-cho, Koganei, Tokyo 184–8584, Japan
‡Dept. of Electrical and Electronics Engineering, Tokushima University
2–1 Minami-Josanjima, Tokushima, 774–8506 Japan
Email: †[haruna.matsushita.pd, tsaito]@k.hosei.ac.jp, ‡nishio@ee.tokushima-u.ac.jp

Abstract—The paper proposes a novel concept of a complex network for the Particle Swarm Optimization (PSO); Independent-minded PSO (IPSO). Particles of the standard PSO always fly toward its own past best position (pbest) and the best position among the swarm (gbest). On the other hand, in the proposed IPSO, whether each particle and the swarm affected each other is stochastically decided according to a fixed parameter called “Cooperativeness”. We confirm behavior of IPSO and its effectiveness by applying it to various benchmarks.

1. Introduction

Particle Swarm Optimization (PSO) [1] is an algorithm to simulate the movement of flocks of birds. Due to the simple concept, easy implementation and quick convergence, PSO has attracted much attention and is used to wide applications in different fields in recent years. Each particle position is updated according to its personal best position (pbest) and the best particle position among the whole swarm (gbest). In other words, all the particles are fully-connected and always influence each other.

Meanwhile, in the real world, we are spending our life influencing each other in the human community, and it is important not just to depend on other people, but also to have own sense of independence.

On the other hand, various topological neighborhoods for PSO have been considered by researches [2]–[7]. In these search, each particle shares its best position among neighboring particles on the network. In other words, it is an application of the network topology to the particle swarm. Thus, investigations of the suitable network for PSO, especially using complex networks such as small-world network [8], have attracted attention in these years [9], [10].

In this study, we propose a novel application of the complex network to PSO; an Independent-minded Particle Swarm Optimization (IPSO). The most important feature of IPSO is that it is decided stochastically that each particle depends on gbest or becomes independent from the swarm and moves depending only on pbest. In other words, the particles are not always connected each other, and they act with self-reliance.

2. Independent-minded Particle Swarm Optimizer (IPSO)

In the algorithm of the PSO, multiple potential solutions called “particles” coexist. At each time step, each particle flies toward its own past best position (pbest) and the best position among all particles (gbest). In other words, they always influence each other. In this study, we propose the novel concept of the complex network; the Independent-minded PSO (IPSO). The particles of IPSO have independence, thus, it is decided stochastically whether they are connected to others at every step. In other words, they are not always affected by gbest and their pbest does not always affect the swarm.

Each particle has two informations; position and velocity. The position vector of each particle i and its velocity vector are represented by \( X_i = (x_{i1}, \ldots, x_{id}, \ldots, x_{iD}) \) and \( V_i = (v_{i1}, \ldots, v_{id}, \ldots, v_{iD}) \), respectively, where \( (d = 1, 2, \ldots, D), (i = 1, 2, \ldots, M) \).

(Step1) (Initialization) Let a generation step \( t = 0 \). Randomly initialize the particle position \( X_i (x_{id} \in [x_{min}, x_{max}],) \), initialize its velocity \( V_i \) to zero, and initialize \( P_i = (p_{i1}, p_{i2}, \ldots, p_{iD}) \) with a copy of \( X_i \). Evaluate the objective function \( f(X_i) \) for each particle \( i \) and find \( P_g \) with the best function value among all the particles.

(Step2) Decide whether each particle \( i \) is connected to the others according to rand, which is a random number \( (\in (0, 1)) \) for the particle \( i \). If \( \text{rand} \leq C \), the particle \( i \) is connected to other particles. If not, the particle \( i \) is isolated from the swarm, then, it and others does not affect each other. \( C \) is a constant cooperativeness coefficient which is the independence probability of the particles.
Table 1: Four Test Functions.

<table>
<thead>
<tr>
<th>Function name</th>
<th>Test Function</th>
<th>Initialization Space</th>
<th>Criterion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sphere func.;</td>
<td>$f_1(x) = \sum_{d=1}^{D} x_d^2$</td>
<td>$x \in [-5.12, 5.12]^D$,</td>
<td>0.01</td>
</tr>
<tr>
<td>4th De Jong’s func.;</td>
<td>$f_2(x) = \sum_{d=1}^{D} d x_d^3$</td>
<td>$x \in [-1.28, 1.28]^D$,</td>
<td>0.01</td>
</tr>
<tr>
<td>Ackley’s func.;</td>
<td>$f_3(x) = \sum_{d=1}^{D-1} \left(20 + e - 20e^{-0.2\sqrt{\frac{1}{50}(x_d^2 + x_{d+1}^2)}} - e^{\frac{1}{5}(\cos(2\pi x_d) + \cos(2\pi x_{d+1}))}\right)$,</td>
<td>$x \in [-30, 30]^D$,</td>
<td>1.0</td>
</tr>
<tr>
<td>Stretched V sine wave func.;</td>
<td>$f_4(x) = \sum_{d=1}^{D} (x_d^2 + x_{d+1}^2)^{0.25} \left(1 + \sin^2(50(x_d^2 + x_{d+1}^{1/4}))\right)$,</td>
<td>$x \in [-10, 10]^D$</td>
<td>10</td>
</tr>
</tbody>
</table>

Figure 1: Test functions with two variables. First and second variables are on the x- and y-axis, respectively, and z-axis shows its function value. (a) Sphere function. (b) 4th De Jong’s function. (c) Ackley’s function. (d) Stretched V sine wave function.

(Step3) Evaluate the fitness $f(X_i)$ for each particle $i$. Update the personal best position (pbest) as $P_i = X_i$ if $f(X_i) < f(P_i)$. 
(Step4) Let $P_g$ represents the best position with the best pbest among particles being connected to others (gbest). Update $gbest P_g = (p_{g1}, p_{g2}, \ldots, p_{gD})$ according to

$$g = \arg \min_i f(P_i), \quad \text{rand}_1 \leq C. \quad (1)$$

In other words, even if the $f(P_i)$ is the minimum pbest among all the particles, gbest is not updated if $i$ is not connected to others.
(Step5) Update $V_i$ and $X_i$ of each particle $i$ according to

$$V_i(t+1) =
\begin{cases}
wV_i(t) + c_1r_1(P_i - X_i(t)) + c_2r_2(P_g - X_i(t)), & \text{rand}_1 \leq C \quad (2) \\
wV_i(t) + c_1r_1(P_i - X_i(t)), & \text{rand}_1 > C
\end{cases}
$$

$$X_i(t+1) = X_i(t) + V_i(t+1), \quad (3)$$

where $w$ is the inertia weight determining how much of the previous velocity of the particle is preserved. $c_1$ and $c_2$ are two positive acceleration coefficients, generally $c_1 = c_2$. $r_1$ and $r_2$ are $d$-dimensional uniform random number vectors from $U(0, 1)$. These equations mean that whether each particle is affected by gbest is decided at random with the cooperativeness $C$. When $C = 0$, all the particles move depending only on own pbest, and when $C = 1$, the algorithm is completely the same as the standard PSO.

(Step6) Let $t = t + 1$ and go back to (Step2).

3. Simulation

In order to evaluate the performance of IPSO, we use 4 benchmark optimization problems summarized in Table 1. $f_1$ and $f_2$ are unimodal functions, and $f_3$ and $f_4$ are multimodal functions with numerous local minima. The optimum solution $X^*$ of all the functions are $[0, 0, \ldots, 0]$, and its optimum value $f(X^*)$ is 0. All the functions have $D$ variables, in this study, $D = 30$. The landscape maps of benchmark functions with two variables are shown in Fig. 1.

We compare IPSO with three PSOs; the standard PSO (PSO), PSO whose two acceleration coefficients are different (PSO2) and IPSO which use either gbest or pbest (IPSO2). Features of each algorithm are follows:

PSO: This is the standard PSO and is completely same as IPSO with $C = 1$. The velocities of all the particles are updated depending on its pbest and gbest at every generation.

PSO2: This is the standard PSO with different acceleration coefficients. All the particles are updated depending on its pbest and gbest at every generation, however the acceler-
... in about every 10 generations. From these results, we can say that quick communication to the swarm is more im-

The maximum generation are set to 3000. For all PSOs, the population size \( M \) is set to 36, and the parameters are set as \( w = 0.7 \) and \( c_1 = c_2 = 1.6 \). In order to investigate the behavior of PSO2, IPSO and IPSO2, we carry out simulations using different cooperativeness \( C \) from 0 to 1.0. The maximum generation are set to 3000 for all the benchmarks, and the results are evaluated in an achievement rate of the criterion attainment over 100 trials.

### 3.1. Experimental Results

Figure 2 shows respective mean results and their achievement rate over 100 runs in different cooperativeness \( C \). Note that the standard PSO used \( C = 1.0 \) for all the simulations. The best mean result among different \( C \), and its value of \( C \) and achievement rate [%] are listed in Table 2.

As shown in Figs. 2(a) and (b), in IPSO for the unimodal functions, the case that all the particles are connected each other at every generations was the most effective, namely \( C = 1.0 \). In other words, the standard PSO was the most suitable to unimodal functions. Then, let us consider the importances of \( gbest \) and \( pbest \) for the unimodal functions from the results of PSO2 and IPSO2. In the results of PSO2, the larger acceleration rate of \( gbest \) was able to obtain better result, especially, it obtained effective results when the acceleration rate of \( gbest \) was about from 90 to 100 percent of that of \( pbest \) (\( C = 0.9–1.0 \)). Moreover, for IPSO2, the performance was effective when the particles moved depending on \( gbest \) in almost generations and on \( pbest \) in about 10 generations. From these results, we can say that quick communication to the swarm is more im-

![Figure 2: Simulation results of three PSOs using different C to four benchmarks. (a) Sphere function. (b) 4th De Jong’s function. (c) Ackley’s function. (d) Stretched V sine wave function.](image-url)
Table 2: Comparison results of four PSOs on 4 test functions.

<table>
<thead>
<tr>
<th>$D$</th>
<th>$f$</th>
<th>PSO</th>
<th>PSO2</th>
<th>IPSO</th>
<th>IPSO2</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>$f_1$</td>
<td>Best avg.</td>
<td>4.73e-51</td>
<td>4.73e-51</td>
<td>4.73e-51</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-</td>
<td>1.0</td>
<td>1.0</td>
<td>0.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Achievement</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td></td>
<td>$f_2$</td>
<td>Best avg.</td>
<td>9.08e-76</td>
<td>5.07e-99</td>
<td>9.08e-76</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-</td>
<td>0.9</td>
<td>1.0</td>
<td>0.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Achievement</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td></td>
<td>$f_3$</td>
<td>Best avg.</td>
<td>16.87</td>
<td>16.87</td>
<td>0.98</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-</td>
<td>1.0</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Achievement</td>
<td>2%</td>
<td>2%</td>
<td>85%</td>
</tr>
<tr>
<td></td>
<td>$f_4$</td>
<td>Best avg.</td>
<td>18.6</td>
<td>18.68</td>
<td>6.92</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-</td>
<td>1.0</td>
<td>0.06</td>
<td>0.04</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Achievement</td>
<td>7%</td>
<td>7%</td>
<td>86%</td>
</tr>
</tbody>
</table>

4. Conclusions

In this study, we have proposed the Independent-minded Particle Swarm Optimization (IPSO). The particles of IPSO act with its sense of independence. Whether each particle is affected by the swarm and itself also affects the swarm is decided stochastically according to the fixed parameter “Cooperativeness”. We have applied IPSO to various benchmark functions containing unimodal and multimodal functions. From results, we have confirmed that the fully-connected IPSO, namely the standard PSO, was suitable for the unimodal functions because it can quickly transmit each particle information to the whole swarm. On the other hand, for the multimodal functions, IPSO, whose particles were little affected by the swarm, significantly improved the fully-connected PSO. This is because PSO with $C < 1.0$ brings diversity to the swarm and the particles can easily get out from the local optima. From these results, we can say that instead of connecting to all the particles, it is better that the particles value own information and are sometimes affected by others. In addition, we have confirmed a possibility of an appropriate value of $C$ depending on the kinds of optimization problems. Our future problem is to investigate the association between the performance of IPSO and its parameters such as the cooperativeness $C$ and the population size $M$.

References

Application of Particle Swarm Optimizers to Finding Desired Parameters of Switched Dynamical Systems

Haruna Matsushita† and Toshimichi Saito†

†Dept. of Electrical and Electronics Engineering, Hosei University
3–7–2 Kajino-cho, Koganei, Tokyo 184–8584, Japan

Abstract—This paper studies a novel application of particle swarm optimizers to finding desired parameters for multi-objective problems in switched dynamical systems. We consider a simplified model of photovoltaic systems such that the input is a single solar cell and is converted to the output via a boost converter. We define realizable parameters with large average power as a desired parameter set. From simulation results, the efficiency of the proposed algorithm is confirmed.

1. Introduction

The Particle Swarm Optimization (PSO) [1]–[2] is an algorithm to simulate the movement of flocks of birds. Each particle of swarm tries to find a better solution according to its personal best position and the swarm best position. The many real/potential applications have been proposed, including design of artificial neural networks, digital filters, power systems and power converters [3]–[6]. It should be noted that the PSO does not require differentiability of the objective function that is a compulsory item in the gradient methods. The PSO is well suited for application to circuits with discontinuous switching that are widely used in a variety of switching power converters.

This paper proposes an application of the PSO to analysis of switched dynamical systems (SDS). The SDS can exhibit rich bifurcation phenomena [7]–[8] and relates to many engineering systems including power converters. In this study, we consider an example of the SDS which is a simplified model of photovoltaic (PV) systems such that the input is a single solar cell and is converted to the output via a boost converter [9]. Since the maximum power point (MPP) of the PV system depends on the operating terminal voltage and current, the maximum power point tracker has been studied as a key technique [10]–[11]. Our SDS includes the solar cell input modeled by a current-controlled voltage source (CCVS) having piecewise linear (PWL) characteristics that can be regarded as a simplified version of the existing smooth model [12]. A switching rule is a variant of peak-current-controlled switching and can cause various bifurcation phenomena. It has been investigated that as a parameter varies, a stable periodic orbit (SPO) is changed into an unstable periodic orbit (UPO) via period-doubling bifurcation and another SPO appears. Furthermore, the UPO can have larger average power than the SPO and can have the MPP.

This paper proposes an application of the PSO to multi-objective problems (MOP) and applies the proposed algorithm to find desired parameters for the MOP in the SDS. The MOP is described by the hybrid fitness functions consisting of the functions evaluating the validity of parameters and criteria. We define a parameter, which is a period-doubling bifurcation set and is the MPP, as a desired parameter which is realizable with large average power. From simulation results, we confirm that the PSO for the MOP can easily find the desired parameter although a numerical calculation needs huge calculation amount.

2. Particle Swarm Optimizer for Multi-objective problems

We propose the application of the PSO to the MOP. Let us consider a positive definite multi-objective function of parameters \( \vec{a} \equiv (a_1, a_2, \cdots, a_D) \):

\[
F_j(\vec{a}) \geq 0, \quad j = 1, 2, \cdots, M, \tag{1}
\]

where \( M = 1 \) and \( M \geq 2 \) correspond to an uni-objective problem and a MOP, respectively. Although usual optimization algorithms try to find the minimum value of \( F_j \), we introduce the inequalities

\[
0 \leq F_j(\vec{a}) < C_j, \tag{2}
\]

where \( C_j \) is the \( j \)-th criterion. Our problem is finding parameter values that satisfy Eq. (2). This flexibility can help to search a suitable solution and can give better result than MOP without the criterion.

In the algorithm of PSO, multiple potential solutions called “particles” coexist. Each particle has two informations: position and velocity. The position vector of \( i \)-th particle at discrete time \( n \) and its velocity vector are represented by \( \vec{a}_i(n) \equiv (a_{i1}, a_{i2}, \cdots, a_{iD}) \in \mathbb{R}^D \) and \( \vec{v}_i(n) \equiv (v_{i1}, v_{i2}, \cdots, v_{iD}) \in \mathbb{R}^D \), respectively, where \( i = 1, 2, \cdots, N \). The position corresponds to the parameter \( \vec{a} \) in Eq. (2). Each particle moves toward the personal best position \( \vec{a}_{pb} (pbest) \), which is the past best position of \( i \)-th particle, and the global best position \( \vec{a}_{gb} (gbest) \) which is the best \( pbest \) among all the particles. The \( \vec{a}_g \) is the potential solution at time \( n \).

Step 1 (Initialization): Let a discrete generation step \( n = 0 \). Randomly initialize the particle position \( \vec{a}_i(0) \) in the search...
space $D_s \subset \mathbb{R}^D$, and initialize other variables; velocity $\vec{v}_i(n) = 0$, $\vec{a}_p = \vec{a}_i(n)$ and $\vec{a}_g = \vec{a}_1(n)$.

Step 2 (Evaluation): Terminate the algorithm if

$$0 \leq F_j(\vec{a}_g) < C_j \quad \text{for } j = 1, 2, \cdots, M.$$

(3)

If not, go to Step 3.

Step 3 (Updating): Update $\vec{v}_i$ and $\vec{a}_i$ of each particle $i$;

$$\vec{v}_i(n + 1) = w\vec{v}_i(n) + \vec{r}_1 \rho_1(\vec{a}_p - \vec{a}_i(n)) + \vec{r}_2 \rho_2(\vec{a}_g - \vec{a}_i(n)), \quad (4)$$

$$\vec{a}_i(n + 1) = \vec{a}_i(n) + \vec{v}_i(n + 1),$$

where $w$ is the inertia weight determining how much of the previous velocity of the particle is preserved. $\vec{r}_1$ and $\vec{r}_2$ are $D$-dimensional uniform random number vectors from $U(0,1)$. $\rho_1$ and $\rho_2$ are two positive acceleration coefficients.

Step 4 (Hybrid fitness): Renew $p_{best}$ if the fitness is improved or satisfies the criteria. Let $\vec{a}_p = \vec{a}_i(n + 1)$ if

$$F_j(\vec{a}_i(n + 1)) < F_j(\vec{a}_p) \text{ OR } F_j(\vec{a}_i(n + 1)) < C_j, \quad (5)$$

is satisfied for all the objective functions. Renew $\vec{v}_{best}$ if $\vec{a}_g = \vec{a}_i(n + 1)$.

Step 5 Let $n = n + 1$, return to Step 2 and repeat until the maximum time limit $n_{max}$.

3. Multi-Objective Optimization for Circuit model of the boost converter with a solar cell.

3.1. Circuit model

![Figure 1: Circuit model of the boost converter with a solar cell](image)

In order to evaluate the performance of the proposed algorithm, we consider detection of the MPP of the SDS which is the simplified model of the PV system. Figure 1 shows the SDS where the 2-segment PWL CCVS models the solar cell input [9]. The dimensionless circuit equation is described by

$$\frac{dx}{dt} = \begin{cases} 
gy(x), & \text{for State 1} \\
gy(x) - q, & \text{for State 2}, \end{cases} \quad (7)$$

$$\text{Figure 2: Typical orbits and instantaneous power } p_{in}(\text{average power } P_A) \text{ for } \alpha = 0.5, \beta = 9.0, q = 1.6, X_e = 0.7 \text{ and } \gamma = 0.884. \text{ (a) SPO with period 2 and (b) } p_{in} \text{ of SPO } (P_A = 0.87). \text{ (c) UPO with period 1 and (d) } p_{in} \text{ of UPO } (P_A = 0.91).$$

$$\text{SW Rule:}\quad \text{State 2 } \rightarrow \text{ State 1: when } x = X_e > 0. \quad \text{State 1 } \rightarrow \text{ State 2: at } \tau = n \text{ and } x > X_e. \quad (9)$$

The dimensionless variables and parameters are defined by

$$\tau = \frac{i}{I_p}, x = \frac{i}{I_p}, \quad y(x) = \frac{V_i(I_p x)}{V_p}, \quad \alpha = \frac{r_s I_p}{V_p}, \quad \beta = \frac{r_h I_p}{V_p}, \quad q = \frac{V_o}{V_p}, \quad \gamma = \frac{T V_p}{I_p X_e}, \quad X_e = J - I_p \frac{J}{I_p}. \quad (10)$$

The dimensionless 5 parameters can be classified into two categories: $(\alpha, \beta, q)$, which characterizes “solar cell and load”, and $(\gamma, X_e)$ which characterizes “switching control”. The piece-wise exact solution is given by

State 1:

$$x(\tau) = (x_0 - x_1) e^{-\gamma(\tau - \tau_0)} + x_1, \quad \text{for } x \leq 1$$

$$x(\tau) = (x_0 - x_2) e^{-\gamma(\tau - \tau_1)} + x_2, \quad \text{for } x > 1$$

State 2:

$$x(\tau) = (x_0 - x_3) e^{-\gamma(\tau - \tau_3)} + x_3, \quad \text{for } x \leq 1$$

$$x(\tau) = (x_0 - x_4) e^{-\gamma(\tau - \tau_4)} + x_4, \quad \text{for } x > 1$$

where $(\tau_0, x_0)$ denotes an initial condition, $x_1 = 1 + 1/\alpha$, $x_2 = 1 + 1/\beta$, $x_3 = q/\alpha - 1 - 1/\alpha$ and $x_4 = q/\beta - 1 - 1/\beta$. Using Eq. (11), we can calculate waveforms precisely. Figures 2(a) and (c) show typical examples; the SPO with period 2 and the UPO with period 1 for the same parameter values as the SPO.

In order to consider the power characteristics, we define the dimensionless instantaneous and average powers:

$$p_{in}(\tau) = \frac{i(t) V_i(t)}{I_p V_p} \cdot \frac{1}{N_p} \int_0^{N_p} p_{in}(\tau) d\tau \quad (12)$$
where $N_p = T_p/T$ is the dimensionless period of the SPO or the UPO for dimensionless time $\tau$. Figures 2(b) and (d) show the instantaneous power $p_m(\tau)$ corresponding to Figs. 2(a) and (c), respectively. The UPO as Fig. 2(d) has lower peak, the $p_n$ has shallower valley and the $P_A$ is larger than that of the SPO in Fig. 2(b). Although the UPO is not observable, it has larger power than the SPO for the same parameter. Such a UPO can have the MPP for the parameter $\gamma$.

In order to analyze and power characteristics, we derive the phase map. Let $\tau_n$ denotes the $n$-th switching time at the lower threshold $X_-$, and let $\theta_d$ be a border time such that a trajectory started from $(\theta_d, X_-)$ reaches $(1, 1)$. We can describe a one-dimensional map explicitly:

$$ F(\tau_n) = \begin{cases} f_1(f_2(f_3(f_4(\tau_n)))) & \text{for } 0 \leq \tau_n < \theta_d \\ g_1(g_2(\tau_n)) & \text{for } \theta_d \leq \tau_n < 1 \end{cases} \quad (13) $$

where

$$ \tau_{n+1} = f_1(\tau_s), \quad \tau_s = f_2(x_s), \quad x_s = f_3(\tau_s), \quad \tau_s = f_4(\tau_n) $$

Let a phase variable $\theta_n = \tau_n \mod 1$. The phase map $f$ from the unit interval $I \equiv [0, 1)$ to itself;

$$ \theta_{n+1} = f(\theta_n) \equiv F(\theta_n) \mod 1, \text{ for } \theta_n \in I. \quad (14) $$

As shown in Fig. 3, this phase map forms a convex curve. A point $p$ is said to be a $k$-periodic point if $p = f^k(p)$ and $p = f^l(p)$ for $1 \leq l < k$ where $f^l(x_p) = f(f^{l-1}(p))$ and $f^0(p) \equiv p$. A 1-periodic point is referred to as a fixed point. A periodic point $p$ is said to be unstable and stable for initial value if $|Df^k(p)| > 1$ and $|Df^k(p)| < 1$, respectively, where $Df^k(p)$ is the slope of $f^k$ at $p$. The stable and unstable periodic points correspond to the SPO and UPO of the SDS, respectively.

Figure 4 shows the average power $P_A$ of a fundamental periodic orbit (FPO) which corresponds to the fixed point $p_1$ in Fig. 3. As $\gamma$ reaches the first period doubling bifurcation set $D_1$ and decreases, the FPO is changed from SPO to UPO and the $P_A$ has the peak (i.e., MPP) at $\gamma = 0.884$, namely the MPP for $\gamma$ ($\partial P_A/\partial \gamma = 0$). The $M$ is one-to-one on the $\gamma$ versus $X_-$ plane and gives the ridge of the $P_A$ characteristics shown as Fig. 5.

![Figure 3: Typical phase maps. $\alpha = 0.5$, $\beta = 0.9$, $q = 1.6$ and $X_- = 0.7$. (a) Stable fixed point $p_1$ for $\gamma = 1.0$. (b) Stable 2-periodic points for $\gamma = 0.884$. $p_1$ is unstable.](image)

![Figure 4: Average power characteristics for $\alpha = 0.5$, $\beta = 9.0$, $q = 1.6$ and $X_- = 0.7$. FPO shows characteristics of the stable/unstable FPO. The maximum average power $P_A = 0.909$ is obtained when $\gamma = 0.884$.](image)

![Figure 5: Parameter sets of bifurcation and MPP for $\alpha = 0.5$, $\beta = 0.9$ and $q = 1.6$. (a) $D_1$: the period doubling bifurcation set. $M$: ridge of the average power of FPO. (b) The average power $P_A$ corresponding to each parameter set.](image)

### 3.2. Multi-Objective function

We define multi-objective functions for finding desired parameter set with MPP of the SDS. Since bifurcation analysis in the 5-dimensional parameter space corresponding to particle position $\vec{a}_i$ in the PSO is extremely hard, we focus on 2-dimensional parameters $\vec{a} \equiv (a_1, a_2) \equiv (\gamma, X_-)$ which control the switching. The search space $D_s$ is defined as $0 < a_1 \leq 2$ and $0.3 < a_2 \leq 0.9$. For convenience, we consider the cross-point of the parameter set $M$ (the MPP for $\gamma$) and the period doubling bifurcation set $D_1$ as the target parameter of the PSO; it is a border of stability of the FPO with MPP for $\gamma$. Note that $P_A$ increases as $\gamma$ decreases and as $X_-$ increases. However, since $\gamma$ depends on the clock period $T$ and device speed as Eq. (10), the circuit with small $\gamma$ is unrealizable.

We then define two objective functions. The first one is...
about the MPP for γ. Whether  is the MPP is obtained by calculating the slope on present position as

\[
F_1(\bar{a}) = \frac{\partial P_a(\bar{a})}{\partial \gamma} = \frac{\partial P_a(\bar{a})}{\partial \alpha_1} \quad (15)
\]

positions.

\[
F_1(\bar{a}) = 0 \text{ means that } \bar{a} \text{ generates the maximum average power on } \gamma, \text{ namely } \alpha_1.
\]

The second objective function evaluates whether  is the period doubling bifurcation set according to

\[
F_2(\bar{a}) = |Df(p_1)| - 1. \quad (16)
\]

positions.

\[
F_2(\bar{a}) = 0 \text{ means that } \bar{a} \text{ is the period doubling bifurcation set } D_1 \text{ shown in Fig. 5, in other words, the system with } \bar{a}
\]

\[
\text{AND} \quad \left( F_1(\bar{a}_p) < F_1(\bar{a}_g) \text{ OR } F_1(\bar{a}_p) < C_1 \right) \quad (17)
\]

The particle \( i \) with the smallest index is chosen. The terminating condition Eq. (3) is given by

\[
F_1(\bar{a}_g) < C_1 \text{ AND } F_2(\bar{a}_g) < C_2, \quad (19)
\]

4. Numerical experiments

For numerical experiments, we use the following parameters:

\[
N = 30, \quad w = 0.7, \quad \rho_1 = \rho_2 = 1.5, \quad n_{\text{max}} = 1500.
\]

\[
C_1 = 2.5e^{-4}, \quad C_2 = 5.0e^{-4}.
\]

Figure 6 shows typical changes of the average power \( P_A \) of \textit{gbest} and typical fitness functions in the searching process. The obtained result is the average power \( P_A = 0.8182 \) with \( \gamma = 1.6463 \) and \( \lambda_c = 0.4823 \). From Fig. 6(b), we can observe that the two fitness functions converged on the criteria \( C_1 \) and \( C_2 \), not only decreasing, but also increasing. This effect was caused by the two criteria in Eqs. (17) and (18). Each decrease of the two fitness helps other increase, and this effect leads the \( P_A \) to the maximum average power with the realizable parameters. Figure 5 shows the parameter \( \bar{a}_g \) obtained by the simulation. From this figure, we can say that the proposed algorithm has found the intersection of \( M \) and \( D_1 \), namely the desired parameter.

5. Conclusions

This study has proposed the application of the PSO to finding the desired parameters of the SDS. In order to find the desired parameters, we have defined the MOP which evaluates the average power of the system and whether the parameters are realizable. Performing basic numerical experiments, we have confirmed the algorithm efficiency.

References

Dynamical Noise injection to Chaotic Dynamics for Solving Combinatorial Optimization Problems

Takayuki Suzuki\(^1\), Takafumi Matsuura\(^2\) and Tohru Ikeguchi\(^3\)

\(^1\)Graduate School of Science and Engineering, Saitama University
\(^2\)Department of Management Science, Tokyo University of Science
\(^3\)Saitama University Brain Science Institute

Email: \(^1\)suzuki@nls.ics.saitama-u.ac.jp, \(^2\)matsuura@ms.kagu.tus.ac.jp, \(^3\)tohru@mail.saitama-u.ac.jp

Abstract—To solve the quadratic assignment problems (QAPs), two types of chaotic search methods have already been proposed. In one method, mutual connection chaotic neural network (CNN), Hopfield-type CNN method is used, and a firing pattern of the CNN represents a solution of the QAP. For another method, execution of local search algorithm is control by the chaotic dynamics. In both methods, chaotic dynamics works to avoid local minima. To improve performances of the these methods, we have already proposed new methods which combine chaotic dynamics and dynamical noise. As a result, when small amount of dynamical noise is added to the CNN, the solving performance is improved. However, we have not clarified yet why the small amount of dynamical noise is effective to find good solutions and how to change the searching states of the chaotic neural network by the dynamical noise. In this paper, to clarify the reason, we analyze the internal states of the neurons.

1. Introduction

In the real world, various combinatorial optimization problems exist, for example, VLSI design, scheduling problem, routing problem, facility layout problem, and so on. It is important to obtain optimal solutions of these problems because operation costs can be reduced. These kind of problems can be formulated as a quadratic assignment problem (QAP) [1]. The QAP is described as follows: when two \(N \times N\) matrices, a distance matrix \(D\) and a flow matrix \(C\) are given, find a permutation \(p\) which minimizes a value of the following objective function \(F(p)\):

\[
F(p) = \sum_{i=1}^{N} \sum_{j=1}^{N} d_{ij}c_{p(i)p(j)},
\]

where \(d_{ij}\) is the \((i, j)\)th element of \(D\), \(p(i)\) is the \(i\)th element of \(p\), \(c_{p(i)p(j)}\) is the \((p(i), p(j))\)th element of \(C\), and \(N\) is the size of the problem. The QAP belongs to a class of NP-hard. Thus, it is required to develop effective approximate algorithms for finding near optimal solutions in a reasonable time frame.

As an approximate algorithm, a method which uses the mutual connection neural networks, or the Hopfield-Tank neural network (HNN), has already been proposed [2]. In this method, a firing pattern of HNN represents a solution of the QAP. If we decide good synaptic weights of HNN for solving the QAP, we can obtain a good solution by descent down-hill dynamics of HNN. However, this method cannot always get good performance because the states of the HNN get stuck at local minima.

To avoid local minima, a method which injects chaotic dynamical noise into HNN for solving combinatorial optimization problems has been proposed [10, 11]. Using fluctuation of chaotic time series as dynamical noise, the state can escape from local minima. As another method for avoiding local minima, a method which uses chaotic neural network (CNN) [3] has also been proposed [4, 5]. In the method, chaotic dynamics of CNN works to avoid the local minima effectively.

To realize more effective algorithm, we have already proposed a new method which uses both chaotic dynamics and dynamical noise for avoiding local minima [7, 8]. As a result, when the small amount of noise is added to CNN, the proposed method shows good performance. However, as the amount of noise increases, the performance of the proposed method becomes worse gradually. Namely, the small amount of noise leads to effective search.

As another approach for solving QAP, 2-opt algorithm driven chaotic dynamics has been proposed [6]. This method shows higher performance than the method which used Hopfield-type CNN. Alternatively we combined this algorithm with dynamical noise. As a result [9], when small amount dynamical noise is added to the CNN, this method also shows good performance.

Although, the methods which use chaotic dynamics and dynamical noise shows good performance [9, 4, 5], we have not clarified yet why the small amount of dynamical noise is effective to find good solutions and how to change the searching states of the CNN by the dynamical noise. If the reasons are identified, the algorithm can be improved further. Thus, it is important to analyze influence of the dynamical noise to the chaotic dynamics. In this paper, we investigate how to change value of internal state of chaotic neurons depending on the amount of dynamical noise. From a result, temporal average and variance of
the internal states of the takes almost same value when we added small amount of dynamical noise in the Hopfield-type CNN method. However, when the large amount of dynamical noise is added to the CNN, the temporal average and variance take quite different value. On the other hand, in the method which 2-opt algorithm driven by chaotic dynamics, the temporal average and the variance of the neurons increase monotonously as the amount of noise increases.

2. Method using both chaotic dynamics and dynamical noise

2.1. Hopfield-type Chaotic Neural Network with dynamical noise

As an approximate algorithm for solving the QAP, a method which uses mutual connection chaotic neural network (CNN) [3] has already been proposed [4, 5]. The CNN model is constructed by chaotic neurons [3]. This neural network model can qualitatively reproduce a chaotic dynamics observed in real neural membrane. To solve an N-size QAP, N × N chaotic neurons are prepared in the method. The (i, m)th neuron corresponds an assignment of the ith facility and the mth city.

An internal state of the (i, m)th chaotic neuron of CNN is defined as follows:

\[ y_{im}(t + 1) = ky_{im}(t) + \sum_{j=1}^{N} \sum_{n=1}^{N} w_{im,jn} f(y_{jn}(t)) - \alpha f(y_{im}(t)) + \theta_{im}(1 - k), \]

where \( k \) is a decay parameter, \( \alpha \) is a strength parameter of a refractory effect, and \( f \) is an output function. Synaptic weights between the (i, m)th neuron and the (j, n)th neuron \( w_{im,jn} \) and thresholds of the (i, m)th neuron \( \theta_{im} \) are defined as follows:

\[ w_{im,jn} = -2(A - \delta_{jm})\delta_{ij} + B\delta_{im}(1 - \delta_{ij}) + \frac{d_{im}c_{mn}}{q}, \]

\[ \theta_{im} = -(A + B), \]

where \( A \) and \( B \) are positive constants, \( \delta_{ij} \) is Kronecker’s delta, and \( q \) is a normalization parameter. As an output function, a sigmoidal function is used:

\[ f(y) = \frac{1}{1 + \exp(-\gamma y)}, \]

where \( \epsilon \) is a gradient parameter of the sigmoidal function. In the method, the chaotic dynamics works to avoid the local minimum problem.

To improve the performance of the method, we have already proposed a new method which injects dynamical noise into the CNN [7, 8]. Thus, a term of the dynamical noise is added to Eq.2. An internal state of the (i, m)th neuron with dynamical noise is defined as follows:

\[ y_{im}(t + 1) = ky_{im}(t) + \sum_{j=1}^{N} \sum_{n=1}^{N} w_{im,jn} f(y_{jn}(t)) - \alpha f(y_{im}(t)) + \theta_{im}(1 - k) + \lambda z_{im}(t), \]

where \( \lambda \) is a weight of dynamical noise and \( z_{im}(t) \) is a sequence of dynamical noise added to the internal state of the (i, m)th neuron at time \( t \). In the method, both the chaotic dynamics and the dynamical noise are used to avoid local minima.

A single iteration is defined as an update of all neurons asynchronously. Then, the CNN generates solutions for updating each neuron asynchronously. However, we cannot always obtain feasible solutions from outputs of the neurons because an output of the chaotic neuron takes an analog value. Then, we use the firing decision method [4] which can always generate a feasible solution for QAP. The procedure of the method is described as follows:

1. Choose an index \((i, m)\) whose internal state \( y_{im} \) takes the maximum value among all the neurons. Then, set the \((i, m)\)th neuron as to firing state, and let \( x_{im} = 1 \).

2. Set other neurons in the \(i\)th row and the \(m\)th column to a resting state, and let \( x_{ik} = 0(k \neq m) \) and \( x_{mi} = 0(l \neq i) \). Then, exclude neurons which have already been selected in Steps 1 and 2.

3. Repeat Steps 1 and 2 until all states of neurons are decided.

2.2. 2-opt Algorithm Driven by Chaotic Dynamics with Dynamical Noise

As another approach for solving QAP, a method in which chaotic dynamics drives the 2-opt algorithm has been proposed [6]. Although the 2-opt algorithm is one of the simplest local search methods, this algorithm does not obtain good solutions because of local minimum problem. To avoid local minima, 2-opt algorithm driven by chaotic dynamics has been proposed [6]. As a result, this method shows better performance than the method which uses mutual connection chaotic neural network.

Then, to improve the performance of this method, we have already proposed a method which combined 2-opt algorithm driven by chaotic dynamics and the dynamical noise [9]. The dynamics of the \((i, j)\)th chaotic neuron is described as follows:

\[ \xi_{ij}(t + 1) = \beta \Delta_{ij}(t), \]

\[ \xi_{ij}(t + 1) = k \xi_{ij}(t) - \alpha x_{ij}(t) + (1 - k)\theta, \]

where \( \xi_{ij}(t) \) is a gain effect of the \((i, j)\)th neuron, \( \Delta_{ij}(t) \) is a difference between value of the current objective function and that of a new objective function when the \(i\)th and the \(j\)th elements in permutation \( p \) are exchanged by the 2-opt algorithm. \( \xi_{ij}(t) \) is a refractory effect of the \((i, j)\)th neuron.
3. Experimental results

3.1. Performance with Respect to Weight of Noise

To evaluate the performance of the proposed methods [7, 8], we use the benchmark problem from QAPLIB[13]. Parameters of the Hopfield-type CNN method (Eq.(6)) are decided as follows:

\[ A = 0.34, \quad B = 0.34, \quad k = 0.87, \quad \alpha = 1.01, \quad \epsilon = 0.02, \quad q = 1100(\text{Had}20) \quad \text{and} \quad q = 100000(\text{Tai}20a). \]

Parameters of the 2-opt driven chaotic dynamical method (Eq.(8)) are decided as follows:

\[ k = 0.5, \quad \alpha = 1.0, \quad \theta = 1.0, \quad \beta = 0.08(\text{Had}20) \quad \text{and} \quad \beta = 0.0002(\text{Tai}20a). \]

The weight of dynamical noise \( \gamma \) is set to several values. Then, we use white Gaussian noise whose average is zero and variance is unity. The proposed method is applied for 2,000 iterations.

Figure 1 shows results of the proposed methods. In Fig.1, the results are expressed by percentages of average gaps between obtained solutions and the optimal solutions for 30 trials. From Fig.1(a), if an amount of dynamical noise is large (weight of noise \( \lambda \) or \( \gamma \) takes small value), the proposed method shows higher performance. However, it is almost same performance for Had20, when we use 2-opt algorithm driven by chaotic dynamics, because it has already obtained good solutions (Fig.1(b)).

3.2. Temporal Average and Variance of Internal States of the Neural Network

To analyze the influence of the dynamical noise to the CNN, we investigate the value of the internal state of the chaotic neurons. We calculate the temporal average and the temporal variance of the chaotic neuron. The temporal average \( \overline{y}_{im} \) and temporal variance \( \sigma^2_{im} \) of the internal states of the \((i, m)\)th neuron are calculated as follows:

\[
\overline{y}_{im} = \frac{1}{T} \sum_{t=1}^{T} y_{im}(t),
\]

\[
\sigma^2_{im} = \frac{1}{T} \sum_{t=1}^{T} (y_{im}(t) - \overline{y}_{im}).
\]

Figure 2 shows relationships between average of \( \overline{y}_{im} \) and \( \sigma^2_{im} \) and the weight of dynamical noise \( \lambda \) for Hopfield-type CNN with dynamical noise method. Figure 3 shows relationships between average of \( \overline{y}_{im} \) and \( \sigma^2_{im} \) and the weight of dynamical noise \( \gamma \) for the method in which 2-opt algorithm is driven by chaotic dynamics with dynamical noise.

From Fig. 2, if an amount of dynamical noise is large (\( \lambda > 0.007 \)), the temporal average becomes small and the temporal variance becomes larger value. Then, the solving performance becomes worse sharply for large amount of noise (Fig. 1). It is consider that the dynamics of chaotic search is broken by large amount of noise. However, if small amount of dynamical noise (\( \lambda < 0.007 \)) is added to the CNN, both of temporal average and variance slightly increase, and the performances are better than original CNN (\( \lambda = 0.0 \)). For the 2-opt algorithm driven by chaotic dynamics with dynamical noise, both of temporal average and temporal variance monotonically increase as the amount of noise increases (Fig. 3). The effects of dynamical noise to the internal states of the CNN are different from two methods.

4. Conclusions

In this paper, to analyze the influence of the dynamical noise added to the CNN, we examine the temporal average and variance of the internal states of the CNN. As a result, when we add an appropriate amount of dynamical noise to the CNN, the temporal average and temporal variance takes almost same value in the Hopfield-type CNN method. However, in the case of large amount of dynamical noise, the temporal average and the temporal variance take quite different from value of the CNN without noise. On the other hand, when we add the dynamical noise to the method in which chaotic dynamics drives the 2-opt algorithm, the temporal average and the variance of the neurons monotonically increase as the amount of noise increases.

In the Hopfield-type CNN, a firing pattern of CNN represents a solution of the QAP. However, feasible solutions cannot be always obtained from outputs of the neurons because an output of the chaotic neuron takes an analog value. To generate a feasible solution, we use the firing decision method [4]. Then, in the feature work, it needs to investigate relationships between solution decided by the firing decision method [4] and amount of the noise.

References

Figure 1: Percentages of average gaps between obtained solutions and the optimal solutions for 30 trials.

Figure 2: Average of $y_{im}$ and average of $\sigma_{im}$ when we change the weight of dynamical noise $\gamma$ by using Hopfield-type CNN with dynamical noise method.

Figure 3: Average of $y_{im}$ and average of $\sigma_{im}$ when we change the weight of dynamical noise $\gamma$ by using 2-opt algorithm driven by chaotic dynamics with dynamical noise.


Facial Expression Recognition
Using a Simplified Head Model and RBF Networks

Koichi Takahashi †, Hironobu Fukai ‡ and Yasue Mitsukura †

†Tokyo University of Agriculture and Technology
2-24-16 Naka-cho, Koganei-shi, Tokyo, Japan
‡Ritsumeikan University
1-1-1 Noji-higashi, Kusatsu-shi, Shiga, Japan
Email: 50010401217@st.tuat.ac.jp

Abstract—In this paper, we propose a facial expression estimation method using radial basis function (RBF) networks. The advantage of an RBF network is that only the arbitrary area of each expression is able to be learned as a classifier. Therefore, we consider whether RBF network is suitable for learning one arbitrary partial expression. The facial expression recognition experiments we conducted demonstrate that the new method estimates the expressions more accurately than the previous method using the Mahalanobis distance.

1. Introduction

In the field of human-computer interaction and computer vision, facial expression estimation is a fundamental challenge. Facial expression estimation is expected to be applied to a variety of fields such as expression mirroring for web chat technology and psychological profiling.

Recently, many facial expression recognition methods have been presented [1]–[3]. Methods for image sequences using a Support Vector Machine (SVM) are proposed [1], [2]. The SVM has the advantage that one can classify the learned expressions accurately. In these methods, high recognition accuracy is achieved using the experimental results. Pose-invariant method proposed in [3] tracks the user’s head pose and estimates the state of facial expressions simultaneously by using particle filtering.

In order to apply facial expression recognition to our lives, the method should distinguish whether the expressions are already learned or not yet learned because human facial expressions vary widely given a person’s emotional state. Classification using SVM is only able to describe the boundary between the expression classes. However, this classifier is not able to recognize whether the input data belong to the class or lies outside of all classes. Of course, by using a Dynamic Bayesian Network, the recognition results will be unsuitable if unknown expression is input.

In an attempt to solve these problems, we have already presented a simplified head model [4]. The simplified head model is based on the architecture in the reference [5] and is able to track a user’s head position, pose and facial deformation using particle filtering [6]. Although the simplified head model can represent a user’s arbitrary facial actions, we have only employed the Mahalanobis distance in the facial deformation feature space to classify the expressions. Perhaps there are more effective methods to be utilized to determine facial expressions. In this paper, we propose a facial expression recognition method using RBF networks. By application of RBF networks for facial expression classification, it is expected that only the arbitrary area of each expression will be able to be learned.

The importance of this study is to demonstrate the effectiveness of RBF networks for facial expression recognition and to propose a method for learning arbitrary expressions. In this paper, we present the details of the proposed method and confirm that the proposed method works as thought.

2. Methods

2.1. Simplified Head Model

In this study, the simplified head model [4] is utilized for the head state tracking. Here, the head state refers to the head position, pose and facial deformation.

The strategy of the simplified head model method is shown in Figure 1. The simplified head model is generated from only one baseline image by approximating the coordinates of a user’s facial parts by a cylinder. The baseline image is the user’s frontal facial image which describes a neutral face. The simplified head model uses 8 feature points: inner and outer corners of both eyes, inner corners of both eyebrows and the outer corners of the mouth. In order to define these feature points, we selected them manually for now. At this time, baseline templates \( T \) are also created based on the feature points from the baseline image.

The simplified head model is a deformable model which has 3 degrees of freedom \( d_{mx}, d_{my}, d_{my} \). The variables indicate the horizontal elastic movement of the mouth, the up-down movement of the mouth and the up-down movement of the eyebrows, respectively. Therefore, the simplified head model can represent the user’s facial action, and facial expressions are recognized by analyzing of these deformation parameters when the user displays corresponding expressions.
Here, we can assume that the head state tracking problem is determining the best match for the simplified head model in each frame image. By using the simplified head model, we can extract each template of each feature point \( T' \) by using the Affine transform and the nearest neighbor algorithm. Hence, if the baseline templates \( T \) and extracted templates \( T' \) match by applying template matching, the simplified head model is placed according to the required parameters. Therefore, 3 dimensional position \( x, y, z \), facial deformation parameter \( d_{mx}, d_{my}, d_{mz} \), and facial deformation parameter \( d_{ex}, d_{ey} \) have to be tracked. As a result, this becomes the 9 state tracking problem. For the tracking method, particle filtering [6] can be employed by using the similarity of template matching as a likelihood function.

Although template matching has a drawback of high computing costs in general, the simplified head model utilizes local template matching only around the feature points. Therefore, this strategy leads to low computing costs.

2.2. RBF Network for Expression Classification

In this study, we regard the facial expression recognition problem as accurate classification of the positive, negative and neutral expressions. Here, positive and negative expressions indicate a smiling face and a disgusted face, respectively.

Up to now, we have conducted facial deformation measuring experiments where the subjects display each facial expression using the simplified head model in the reference [4]. These experiments were conducted on five subjects aged 22 to 24 consisting of four males and one female. The results are shown in Figure 4(a). In this figure, each scale of axis is determined based on the distance between the center of the eyes which is set at 80. Each expression has 4103, 4352 and 4742 data points. These results indicate that each expression is densely distributed, where the positive state varies toward the \( d_{mx} \) and \( d_{my} \) axes, and the negative state varies toward the \( d_{ex} \) axis.

For this facial expression recognition method we employ the RBF network instead of the Mahalanobis distance used in the previous method. The RBF network is utilized for the interpretation problem in general. A schematic of the RBF network with \( n \) inputs and a scalar output is described in Figure 2. This network implements a mapping \( f_r: \mathbb{R}^n \rightarrow \mathbb{R} \) according to

\[
f_r(x) = \sum_{i=1}^{m} a_i \phi(||x - c_i||)
\]

where \( \phi(\cdot) \) is an RBF, \( a_i \), \( 1 \leq i \leq m \) are the weights, \( c_i \in \mathbb{R}^n \), \( 1 \leq i \leq m \) are RBF centers and \( m \) is the number of centers. In this study, the Gaussian function

\[
\phi(r) = \exp\left(-\frac{r^2}{\sigma^2}\right)
\]

where \( \sigma \) is a real constant, is defined as RBF.

The challenging in this method for facial expression classifying is determining which output has the highest value between the three RBF networks corresponding to each facial expression. It is therefore required that the RBF network outputs close to 1 when the corresponding expression is displayed, and the other networks output close to 0. At this point, the considerable problem is how to select centers \( c_i \) and define constant \( \sigma \). In practice, the centers are normally chosen from the data set \( \{x(t)\}_{t=1}^{N} \). Chen [7] presented a method where the set of \( M \) candidate regressors \( \{x'(t)\}_{t=1}^{N} \) is first selected randomly. Subsequently, the centers are selected based on the orthogonal least squares (OLS) algorithm.

Let \( \{x(t)\}_{t=1}^{N} \) is the set of \( N \) data points corresponding to the arbitrary expression. We can regard that learning an RBF network for each expression is the least squares problem as follows:

\[
y = Pa
\]

where \( y \in \mathbb{R}^N \) is a desired output vector whose elements are all 1, \( a \in \mathbb{R}^M \) is a weight vector whose \( i \)th element is \( a_i \) and \( P = [p_1 \ p_2 \ \cdots \ p_M] \in \mathbb{R}^{N \times M} \) is a matrix whose element \( p_{ji} \) is defined as

\[
p_{ji} = \phi(||x(j) - x'(i)||).
\]

Here, the matrix \( P \) can be decomposed into

\[
P = WB
\]

where \( W \in \mathbb{R}^{N \times M} \) is a matrix with orthogonal columns \( w_i \), and \( B \in \mathbb{R}^{M \times M} \) is an upper triangular matrix.
Step 1 \((k = 1)\)
\[\begin{align*}
w_1^{(i)} &= p_i, \\
r_1^{(i)} &= \|w_1^{(i)}p_i\|_2^2 / N(w_1^{(i)}P_1w_1^{(i)}) \\
i_1 &= \arg\max_{1 \leq i \leq M} r_1^{(i)} \\
w_1 &= w_1^{(i_1)}
\end{align*}\]

Step \(k \ (k = 2, 3, \ldots, M)\)
\[\begin{align*}
b_j^{(i)} &= w_j^T p_i / (w_j^T w_j), \quad j = 1, \ldots, k - 1 \\
w_k^{(i)} &= p_i - \sum_{j=1}^{k-1} b_j^{(i)} w_j \\
r_k^{(i)} &= \|w_k^{(i)}p_i\|_2^2 / N(w_k^{(i)}P_1w_k^{(i)}) \\
i_k &= \arg\max_{1 \leq i \leq M, i \neq i_1, \ldots, i_{k-1}} r_k^{(i)} \\
w_k &= w_k^{(i_k)}
\end{align*}\]

Figure 3: RBF network learning algorithm.

Chen’s method computes each column of \(W\) and orthogonalizes \(P\) simultaneously using the OLS method based on the Gram-Schmidt scheme. In the case where the elements of desired output vector \(y\) are all 1, Chen’s algorithm is described as shown in Figure 3. \(\| \cdot \|_1\) denotes an \(L^1\) norm. We can consider that \(r_k^{(i)}\) explains the contribution ratio of the candidate regressor \(x'(t)\) at step \(k\). The regressor with the highest contribution ratio is therefore selected in each step. The procedure continues while
\[1 - \sum_{j=1}^{k} r_j^{(i)} < \rho\] (6)
where \(0 < \rho < 1\) is a chosen tolerance.

3. Experimental Results and Discussions

Fortunately, each facial expression distribution is not very complicated. It is therefore important that the tolerance should be set low, but overfitting has to be avoided. In order to evaluate resulting RBF networks, we utilize the \(a_{\text{max}}\) and \(a_{\text{min}}\) which is the maximum and minimum element of the weight vector \(a\), respectively. If \(a_{\text{max}}\) is well over 1 or \(a_{\text{min}}\) is a negative value, we can consider that overfitting has occurred. For this reason, we conducted the learning 10000 times per expression due to the fact that \(a\) depends on the randomly selected regressors \(|x'(t)|_{i=1}^{M}\), and selected the result with the least \(a_{\text{max}}\) and nonnegative \(a_{\text{min}}\). Moreover, we decided \(\sigma\) such that the minimum results \(\sigma\) without overfitting. \(\rho\) is set to 0.02 due to the fact that overfitting happens frequently when \(\rho\) is less than one. Figure 4(b) shows the RBF network learning results for each expression, with \(\sigma\) corresponding to the expression positive, negative, and neutral at 3.5, 3.0, and 2.0, respectively. In addition, the number of the center points for each expression is 14, 13, and 11, respectively.

Table 1: Recognition Results.

<table>
<thead>
<tr>
<th>Input</th>
<th>Positive[%]</th>
<th>Negative[%]</th>
<th>Neutral[%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive</td>
<td>100.0(99.1)</td>
<td>0.0(0.0)</td>
<td>0.0(0.0)</td>
</tr>
<tr>
<td>Negative</td>
<td>0.0(0.0)</td>
<td>98.4(99.1)</td>
<td>1.6(0.0)</td>
</tr>
<tr>
<td>Neutral</td>
<td>0.2(0.8)</td>
<td>0.8(4.5)</td>
<td>98.9(94.7)</td>
</tr>
</tbody>
</table>

For facial expression recognition, if the RBF network outputs over 0.1 and greater than other outputs, the expression can be regarded as the corresponding expression. At this point, neutral is given preference since the false estimations from neutral into other expressions should be prevented. Figure 5 describes the results of the expression recognition using a test movie. In this movie, we defined Frames 1 to 433, 686 to 763, and 978 to 1078 as the neutral, frames 446 to 672 as the positive, and the frames 773 to 971 as the negative. The average results of 10 experiments are shown in Table 1. The values shown in parentheses denote the results using the previous classifier, the Mahalanobis distance. The total recognition accuracy shows 99.1\% which is better than the 96.5\% shown by the previous method. Consequently, we were able to confirm that the RBF networks effectively represent the expressions.

4. Conclusions

In this study, we adopted RBF networks for a simplified head model and utilized them for facial expression recognition. According to the experimental results, the effectiveness of our approach is confirmed, and the precision of recognition is increased. In the future, we are going to modify the system to recognize more kinds of expressions.

References


(a) The distribution of each facial expression

(b) The position of each unit

Figure 4: The learning results of RBF network.

Figure 5: Simulation results using the proposed method.


Facial Feature Animation and Its Artistic Representation

Yang Yang†‡, Yuanqi Su†, Yuehu Liu† and Yoshifumi Nishio‡

†Institute of Artificial Intelligence and Robotics
Xi’an Jiaotong University, Xi’an 710049, China
‡Dept. of Electrical and Electronic Engineering
Tokushima University, Tokushima 770-8506, Japan
Email: {yyang, yqsu, liuyh}@aiar.xjtu.edu.cn, nishio@ee.tokushima-u.ac.jp

Abstract—Facial animation and artistic representation are active research topics in the multimedia industry. This paper addresses facial object with geometric feature to generate expressions using the artistic representation. Firstly, we propose an animation method to transfer expressions from a source face to a target face automatically, which is effective when only a single face with neutral expression is available as the novel target. Then, we provide a facial artistic representation framework and exhibit its producing results by facial cartoon and sketches. Combining above work by building the corresponding map between the facial animation and representation features, we generate expressions for the target face vividly.

1. Introduction

Facial feature animation and its artistic representation has played an important role in the human’s lives with increasing requirements in the entertainment industry and multimedia communication. Recently, the movie Avatar made a huge success by using the related techniques. It took us to a spectacular world by exhibiting real actors’ performance onto fanciful characters in the movie. Meanwhile, more and more communication tools in the daily life, like the MSN and mobile phone, show their potential applications for the personalized interface. To generate the artistic facial animation will be welcomed by every user. In this paper, we firstly animate facial features by transferring expressions from a source performer’s face to target faces, and then exhibit the animation results using the artistic representation.

The commonly used animation technique in the movie industry needs animators to repeatedly define facial animation parameters for every face. In addition, it often requires complex computing and manual adjusting because of the correlations among these parameters. Some blendshape methods [1]-[3] were proposed based on key examples of the source and target faces. They simplified the animation process just to estimate and transfer the blending weights. However, it is also a tedious work to prepare key examples for every face. In this paper, our animation method directly transfers and adjusts motion vectors of the source face to target faces, which can animate the novel target face effectively.

Some impressive works have been developed to produce facial artistic representations. PicToon [4] utilized a statistic model with non-parametric sampling to create cartoon from the given face image. It can generate the facial sketches and specific animations from templates. The MSN cartoon [5] was an online application published by Microsoft. Human’s interactions are required by the system to select and adjust the personalized cartoon shape and accessories. A facial sketch generation method was proposed by Xu et al. [6], which gave a hierarchical facial representation for rendering facial sketches. In our early work about a face cartoon producer [7], a cartoon rendering method was proposed based on the automatic template feature fitting and adaptation. In this paper, we extend the producer to generate artistic animation results by the corresponding map built between the facial animation and representation features.

2. System Overview

The system overview is shown in Fig. 1. When input a target face image with neutral expression, a set of feature points are extracted by the ASM [8] method. We can render an artistic representation for this target face by the artistic face producer. For generating expression animation sequences, source expressions are transferred to target facial features firstly, and then the whole animation results are represented with the same style as the artistic target face. In the facial expression database, examples are labeled off-line on various expressions of different people. In the artistic face database, rendering templates are drawn by artists. In the following sections, we describe each function block in the facial feature animation and representation in more detail.

3. Expression Transfer for Facial Feature Animation

Human’s faces share a common topological structure and exhibit a similar movement when making the same expression. When just one single face is available as the target input, Blanz et al. [9] gave an effective expression transfer method by the vector space operations. Their method is based on the assumption that the 3D displacements of surface points are the same for all individuals, which is just
suited to very similar faces. We propose a two steps expression transfer method. Inspired by Blanz’s work [9], we could firstly acquire a coarse animation result using the direct motion vector’s transferring. Although that may generate an unreasonable face, it has mapped the source face’s movement on the target face roughly. And then we adjust this coarse animation result according to the personality of the target face.

3.1. Global Movement Transferring

Facial feature is represented by a set of feature points’ coordinates \( C = (x_1, y_1, x_2, y_2, ..., x_N, y_N) \), where \( N \) is the points’ number. We suppose that the first frame in source expression sequence is always with the neutral expression. Motion vectors are calculated by the source neutral face \( C_{SN} \) and expressional face \( C_{SE} \), and transferred to the target facial feature \( C_{TN} \) directly as:

\[
C_{TE} = C_{TN} + C_{SE} - C_{SN}. \tag{1}
\]

Without the magnitude and direction adjusting, the transfer result \( C_{TE} \) is easily to cause an unreasonable face. To improve it, we project \( C_{TE} \) in the facial space and reconstruct it by principal component analysis (PCA) model:

\[
C'_{TE} = \hat{C} + Pb, \tag{2}
\]

where, \( \hat{C} \) is the mean shape of training examples and \( P \) contains unit eigenvectors of the covariance matrix. \( b \) is the principal component for a given \( C_{TE} \). The reconstructed \( C'_{TE} \) is the most similar reasonable face as \( C_{TE} \) for the reason of the PCA.

One problem is that the PCA model is learned based on various faces in the database, which is hard to maintain the personality especially from a coarse input. Moreover, the motion vectors of the source face also need to be adapted for the target face. In the following part, since a globally reasonable facial structure could be guaranteed by the PCA, we discuss the local feature adjustment for each facial component.

3.2. Local Personalized Adjustment

Xiong et al. [10] proposed a facial component model using neighbor reconstruction method for personalized expression synthesis. In the facial parameter space, they used neighbor examples to reconstruct target face, and transferred reconstruction weights for generating target expressional faces. However, every expression of the source face should be known in advance by the animator, since they needed re-preparing corresponding expression’s examples. Here, we use the neighbor reconstruction as a constraint to formulate the personalized adjustment as an optimization problem.

We construct the facial expression database by a set of example pairs. Every example pair contains one neutral face and one expressional face belonged to the same person. All facial features are separated into seven local subsets according to facial components, including left and right eyes, left and right eyebrows, nose, mouth and face contour.

Firstly, each feature subset of target face finds the neighbor examples by the k-nearest neighbor algorithm from neutral faces in the example pairs.

And then, we calculate the reconstruct weights for local component based on two principles: one is the reconstructed similarity with target neutral face; the other is to match with the current expression. Using the global movement transferring result \( C'_{TE} \) as the current expressional face, the optimization function is built as follows:

\[
\min \left( \| C'_{TE} - \sum_{i=1}^{K} \alpha_i^c S_{iN}^c \|_2^2 + \lambda \| C'_{TN} - \sum_{i=1}^{K} \alpha_i^c S_{iN}^c \|_2^2 \right)
\]

s.t. \( \alpha_i^c \geq 0 \), \( \sum_{i=1}^{K} \alpha_i^c = 1 \). \tag{3}

We use \( c \in (1, 2, ..., 7) \) as the index for seven facial components to represent facial feature subset. \( S_{iN}^c \) and \( S_{iE}^c \) are the neutral and expressional facial feature subset belonged to \( i \)th example pair in the \( K \) neighbors. Parameter \( \lambda \) balances two terms in Eq. (3).

This is a quadratic problem with linear constraints, where the objective function is positive semidefinite. We
use fmincon function in Matlab to solve Eq. (3) and acquire a set of optimized weights $\alpha_c^i$.

Finally, the fine animation result $C_{T,E}^{\text{rec}}$ is acquired by reconstructing from neighbor expressional examples using weights $\alpha_c^i$:

$$C_{T,E}^{\text{rec}} = \sum_{i=1}^{K} \alpha_c^i S_{T,E}^i.$$  \hfill (4)

4. Artistic Facial Representation

Our prior work [7] proposed a face cartoon producer for automatically rendering cartoon based on extracted features from static input face image. This producer can be easily extended to a multiple styles artistic face producer by just changing the rendering templates. Liu et al. [11] presented a three-layer framework for cartoon facial animation, but did not discuss the animation method for the sequence generation. In this section, we briefly introduce the framework of artistic face producer and then give the details about the animation sequence generation.

4.1. Artistic Face Producer

Given an input face image and extracted feature points, the process for generating artistic presentation has the following steps.

**Template selection:** A group of rendering templates are drawn by artists. All these templates are parameterized with rendering vector’s information, rendering rules, facial geometric features, and so on. For the target face, the most similar template is selected by the facial features’ comparison.

**Model adjusting:** The selected template is similar to the target face, but still needs to be adapted for exactly fitting the target facial feature by the geometric deformation algorithm.

**Artistic Rendering:** The producer combines the painting entities with the picked out rendering rules and spatial arrangements of the template, and outputs feature representation with artistic style in a vector way.

With above steps, facial features could be artistically presented correspondingly. Two classes of artistic face rendering results, facial cartoon and sketches, are shown in Fig. 2.

4.2. Animation Sequence Generation

The rendering style should be consistent to all the frames in the expression sequence for one target face. Facial features in the animation sequence have been acquired in the section 3. Here, we build the corresponding map between the facial animation and representation features. Then the rendering vectors of the artistic target face are warped to each animation result correspondingly.

Suppose the facial feature in artistic target face is represented as $(u, v)$, and facial feature of an arbitrary frame in the animation sequence is $(x, y)$. To build correspondence mapping between two set of features, we use thin-plate splines (TPS) [12] in the following form for 2D points.

$$f(x, y) = d_1(x, y) + \sum_{i=1}^{N} \omega_i U(||(x, y)^T - (u, v)^T||)$$

$$f(x, y) = d_2(x, y) + \sum_{i=1}^{N} \omega_i U(||(x, y)^T - (u, v)^T||).$$  \hfill (5)

Here, $U(r) = r^2 \ln r$ is the kernel function in the TPS. $d$ represents the affine transformation, and $\omega_i$ represents the non-affine deformation. All the coefficients are solved by a minimized energy function as defined in [12].

The reason for choosing TPS as the mapping function is its suitability in the facial deformation. There is a smoothness measure term $\beta \int \sum_i (\frac{\partial f_i}{\partial x})^2 + 2(\frac{\partial f_i}{\partial y})^2 + (\frac{\partial f_i}{\partial v})^2) dx dy$ in the minimized energy function for the TPS fitting, which is in charged by the parameter $\beta$. We can control the smoothness of the transformation by different $\beta$ flexibly. During people making expressions, rigid transformation dominates the eyebrow’s motions, so that we choose a bigger $\beta$ to animate eyebrow; but for the mouth, a smaller $\beta$ is used for its variable shape.

At the end, we repeat the third step in the artistic face producer to render the entity in each frame for generating the final animation sequence. Some animation results are artistically presented in Fig. 3, in which the source expressions are collected from the Cohn-Kanade database [13].

5. Conclusion

In this paper, we presented a facial feature animation and artistic representation method, which can generate expression animation results with artistic style. Our method is suited for the novel input face image. No complex model parameters need to be adjusted, which saves a lot of manual work for the application. We extended our artistic face producer by the corresponding map built between the facial animation and representation features. The animation results were presented vividly.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant No. 60775017 and the National Basic Research Program of China under Grant No. 2007CB311005.

References


Figure 2: Facial artistic representation.

Figure 3: Animation results with facial sketches representation.


An Embedding and Detection Method of Invisible Calibration Pattern for Print-Type Data Hiding

Hironori Takimoto†, Seiki Yoshimori, Yasue Mitsukura and Minoru Fukumi‡

†Okayama Prefectural University, 111, Kuboki, Soja, Okayama, 719-1197, JAPAN
‡Nippon Bunri University, 1727, Ichigı, Oita, Oita, 870-0397, JAPAN
Tokyo University of Agriculture and Technology, 2-24-16, Nakamachi, Koganei, Tokyo, 184-8588, JAPAN
The University of Tokushima, 1-1, Minami-Josanjima, Tokushima, Tokushima, 770-8506, JAPAN
Email: takimoto@c.oka-pu.ac.jp, yoshimori@nbu.ac.jp, mitsue@cc.tuat.ac.jp, fukumi@is.tokushima-u.ac.jp

Abstract—In the print-type steganographic system and watermark, a calibration pattern is arranged around contents where invisible data is embedded, as plural feature points between an original image and the scanned image for normalization of the scanned image. In this paper, we propose an arrangement and detection method of an invisible calibration pattern based on the characteristics of human visual perception. The most important part of human visual perception in the proposed method is the spectral luminous efficiency characteristic and the chromatic spatial frequency characteristic. We embed the calibration pattern in an original image by adding high frequency component to blue intensity in a limited region. It is suggest that the proposed method protect page layout and artwork of original contents.

1. Introduction

Recently year, a print-type steganographic system and a watermarking technique had been widely studied by the improvement of scanning devices such as a digital scanner and a digital camera [1]-[6]. A basic concept of these methods is to embed invisible data into printed images and documents. Those printed contents are distributed widely, and it is used for security and internet services. In the digital watermarks system, embedding data is used for copyrights protection. On the other hand, print-type steganographic technique is one of data hiding method, does not reduce image quality. Moreover, it does not interfere with page layout and artwork.

Generally, when a printed image is converted from analog data to digital data by using the scanning device, the scanned image is influenced by geometrical transform, light source, noise and so on. Therefore, in order to normalize the scanned image, it is necessary to detect plural feature points which are one to one corresponding points between an original image and the scanned image. In the conventional method, many kind of calibration pattern have been proposed to represent these feature points. To take an example of typical calibration pattern, visible calibration patterns were set to four corners of contents. Nakamura set a black border to the surroundings of contents [4]. In [5], visible calibration patterns are arranged around contents where invisible data is embedded. On the other hand, shape and background color of contents is restricted to simplify extraction of region where invisible data is embedded [6]. Hence, it is clear that conventional methods interfere with page layout and artwork of contents.

In this paper, we propose an arrangement and detection method of an invisible calibration pattern based on characteristics of human visual perception. The most important part of human visual perception in the proposed method is the spectral luminous efficiency characteristic and the chromatic spatial frequency characteristic. The calibration pattern is embedded to blue intensity in an original image by adding high frequency component. It is difficult to perceive the embedded calibration pattern by the effects of the human visual perception characteristic. In addition, background in the surrounding of contents is not restricted to uniform color. It is suggest that the proposed calibration pattern protects page layout and artwork.

2. The characteristics of visual perception

A human recognize electromagnetic waves whose wavelength is 360 - 800 [nm] as rays of light. Then, human perceives color when signal is transmitted from a visual cell to a brain. Under bright light condition, a cone operates actively compared with a rod cell. Sensitivity of blue light in the human visual characteristic is lower than other color. By the way, when human see a periodic stimulation pattern, the spatial frequency characteristic in the human visual characteristic is defined to the threshold of contrast. In other words, it is frequency characteristic of sensitivity which is defined by the inverse of the threshold. Generally, it is difficult for human to perceive pattern which changes cyclically at high frequency. The example of chromatic
and brightness spatial frequency characteristics of the visual system is shown in Fig. 1 [7]. For instance, when human see a periodic stimulation pattern which was constructed by two colors, there is a possibility that human perceive other color. In the high frequency domain of the pattern such as Fig. 2, it is perceived as uniform gray by influence of the spatial frequency characteristic.

3. Proposed Method

3.1. Pattern template

In this paper, we propose a new digital template to set the invisible calibration pattern. The 2-dimensional template is created based on Eq. 1, and the detail of the template is shown in Fig. 3.

\[ \text{template}(x, y) = \alpha \cos \frac{\pi}{\beta} \sqrt{x^2 + y^2} * e^\gamma \sqrt{x^2 + y^2} \]  \hspace{1cm} (1)

The cyclical structure of this template is defined concentrically on the basis of center of template. In Eq. 1, the parameter \( \alpha \) means the maximum amplitude of template, the parameter \( \beta \) means a period. The parameter \( \gamma \) means damping. The origin of template is the central part of template. In the origin, the template has the maximum amplitude.

3.2. Embedding of calibration pattern

In order to embed the invisible calibration pattern in an original image, coefficients of defined template are added to blue component of each pixel. The reason why the blue component is changed is that sensitivity of blue light is lower than other light in the human visual characteristic. In order to embed the calibration pattern in point \((x, y)\), the blue component in the range of \((-\frac{l^t}{2} < i, j < \frac{l^t}{2})\) is operated based on the following equation.

\[ T_B(x + i, y + j) = I_B(x + i, y + j) + \text{template}(i, j) \]  \hspace{1cm} (2)

The parameter \( f_s \) means width of the proposed template. The \( I_B(x, y) \) is blue intensity of point \((x,y)\) in the original image. The \( T_B(x, y) \) is blue intensity of embedded image. Generally, the BMP image is defined as 24 [bit] based on RGB color image, and each intensity value is shown by 8 [bit]. If the template which has high amplitude is used for the embedding of calibration pattern, there is a possibility that the blue component of calibration pattern in the embedding image is not expressible by 8 [bit]. For this reason, if the pixel not expressible by 8 [bit] is conformed when the proposed template was embedded in the original image, the maximum excess value is operated based on the following equations. The parameters \( i \) and \( j \) ranges from \((-\frac{l^t}{2} < i, j < \frac{l^t}{2})\).

\[ E_u(x, y) = \max_{-\frac{l^t}{2} \leq i, j \leq \frac{l^t}{2}} \{ T_B(x + i, y + j) - 255 \} \]  \hspace{1cm} (3)

\[ E_l(x, y) = \max_{-\frac{l^t}{2} \leq i, j \leq \frac{l^t}{2}} \{ 0 - T_B(x + i, y + j) \} \]  \hspace{1cm} (4)

\[ T_B(x+i, y+j) = \begin{cases} T_B(x+i, y+j) - E_u(x, y) & (E_u > 0) \\ T_B(x+i, y+j) + E_l(x, y) & (E_l > 0) \end{cases} \]  \hspace{1cm} (5)

However, if the calibration pattern was embedded in region which has blue intensity with variation, it is difficult to detect the pattern embedded in the normal background image, since an embedded pattern had lost defined shape by influence of original background.
To avoid decrease of detection accuracy, we propose the improvement of embedding method. The procedure includes the following step: firstly, the calibration pattern is temporarily embedded in point \((x, y)\) of the original image. Then, the Normalized Cross Correlation (NCC) on target pixel \((x, y)\) is calculated. If \(\text{NCC}(x, y)\) is less than defined threshold \(T\), the blue intensity of the original image is smoothed to avoid decrease of detection accuracy. The blue component in the range of \((x - \frac{L}{2} < a < x + \frac{L}{2}, y - \frac{L}{2} < b < y + \frac{L}{2})\) on the basis of point \((x, y)\) is operated based on the following equation.

\[
I_B(a, b) = \frac{1}{w^2} \sum_{j=-\frac{w}{2}}^{\frac{w}{2}} \sum_{i=-\frac{w}{2}}^{\frac{w}{2}} I_B(a + i, b + j) \tag{6}
\]

In this equation, the parameter \(w\) means the range of smoothing filter. Finally, to ascertain whether the similarity \(\text{NCC}(x, y)\) is higher than the threshold, the calibration pattern is temporarily embedded. If similarity is less again, the smoothing is carried out several times. As a result of this smoothing, it is confirmed that image quality does not deteriorate by influence of human image perception.

### 3.3. Detection of Calibration Pattern

Conventionally, the NCC is used for a detection method of proposed calibration pattern \[8\]. The NCC is robust for linear changing of brightness and additive noise. However, the detection using the NCC requires a lot of time.

Thus, in order for the detection of calibration pattern to achieve high-speed and high accuracy processing, the detector composed of three kinds of brief weak classifiers is proposed. The new similarity \(R(x, y)\) composed of some brief weak classifiers is shown in Eq. (7).

\[
R(x, y) = \frac{1}{w^2} \sum_{j=-\frac{w}{2}}^{\frac{w}{2}} \sum_{i=-\frac{w}{2}}^{\frac{w}{2}} \left( D_1(i) D_2(i) \right) (a - 1) + \sum_{k=0}^{\frac{w}{2}} D_3(k) \tag{7}
\]

\[
w_i = \text{template}(-\frac{f_8}{2} + i, 0), \quad v_i = T_B(x - \frac{f_8}{2} + i, y) \tag{8}
\]

The target pixel for the calculation is restricted in the direction of one-dimensional. The weak classifier \(D_1(i)\) estimate the direction of gradient of pixel value, which is represented by:

\[
D_1(i) = \begin{cases} 
1 & \text{If the sign of } W_i \text{ and } V_i \text{ is the same} \\
0 & \text{otherwise}
\end{cases} \tag{9}
\]

where \(W_i\) means \(w_{i+1} - w_i\), and \(V_i\) means \(v_{i+1} - v_i\). The weak classifier \(D_2(i)\) estimate the difference pixel value between target pixel and neighboring region.

\[
D_2(i) = \begin{cases} 
1 & \text{If } |V_i| < M \\
0 & \text{otherwise}
\end{cases} \tag{10}
\]

The threshold \(M\) is defined from shape of the embedded pattern. The weak classifier \(D_3(k)\) judge the direction of damping by using the extreme value.

\[
D_3(k) = \begin{cases} 
1 & \text{If } u_k < u_{k+1} \quad (k < \frac{w}{2}) \\
0 & \text{or } u_k > u_{k+1} \quad (k > \frac{w}{2}) \\
0 & \text{otherwise}
\end{cases} \tag{11}
\]

In the Eq. (7), the parameter \(w\) is same as the number of extreme value of proposed template. The parameter \(u_k\) means absolute value of difference between each extreme value.

### 4. Evaluation

#### 4.1. Experimental Condition

We use 25 images including a landscape, a portrait and an illustration for evaluation. The size of original images is \(256 \times 256\) [pix.], and these are 24 [bit] RGB color. The blank space of about 50 pixels is provided to the surrounding of original image. The test data for the simulation is composed of a uniform background image and a normal background image. In the uniform background image, a color based on two colors which were selected at random is painted on the blank space. The sample of these images is shown in Fig. 4. The calibration patterns are embedded in four regions shown by green line. Experimentally, the embedded pattern has each parameter \(\alpha = 40, \beta = 2.0, \gamma = -0.1, \text{w} = 5, T = 0.25\). In addition, the size of calibration pattern is set to 31 [pix].

Moreover, we use two type of scanning devices, the digital scanner (CANON F5400) and digital camera (CANON PowerShot S3 IS). The resolution for scanning is 150 [dpi] and 600[dpi], respectively.

#### 4.2. Evaluation of detection

In this simulation, we evaluate the combination of brief weak classifiers for stable detection of calibration.

![The uniform image](image1.png) ![The normal image](image2.png)

Figure 4: The sample of test image
pattern. The results of pattern detection are shown in Tab. 1. As the case 1, the similarity is composed of only weak classifiers (\(D_1, D_2\)) for estimation of the periodicity. As the case 2, the similarity includes all weak classifiers. The similarity \(R(x, y)\) is calculated from horizontal and vertical pixels on the basis of target pixel. For the purposes of comparison of proposed similarity with conventional method, we show the detection accuracy using the NCC. From this result, the detection rate in the proposed similarity is lower than the conventional method.

To evaluate the performance of execution time, we implemented it with the support of the Intel OpenCV library using a PC with Microsoft Windows XP SP3. The hardware platform for the experiment is a PC equipped with an Intel Core 2 Duo 2.8GHz CPU and 2GB RAM. As a result, execution time of case 2 is 11.8 [ms], execution time of the NCC is 223.8 [ms].

In order to improve the detection accuracy, the region for calculation of similarity is expanded. The extended regions consist of a diagonal lines and four lines adjoining lines defined previously. As a result, detection rate in the uniform background is 98[\%], and the normal background is 83[\%]. On the other hand, execution time is 50.8 [ms]. Therefore, it is confirmed that the detection accuracy and execution time of proposed method is sufficient compared with conventional method.

### 4.3. Evaluation of invisibility

In order to show the evaluation of invisibility, we performed the subjectivity experiment based on the single-stimulus methods. In this experiment, one calibration pattern is arranged to the normal background image at random. On the other words, the proposed pattern is not embedded in three other corners. We show some test images to 9 test subjects. The test subject answers to the question where pattern was embedded. As one measure of the performance of invisibility, we require detection accuracy and time for questions to get answered. The parameters of arranged pattern are \(\alpha=40, \beta=(2.0, 3.0), \gamma=-0.1\). The resolution for printing is 150 [dpi]. The distance between printed paper and test subject’s eye is restricted to 30 [cm].

As a result of subjective experiment, detection rate is 16.6[\%], and response time is 8.3[s]. On the other hand, when the periodic parameter is \(\beta = 3.0\), each result are 22.2[\%] and 7.1[s]. From these results, invisibility of proposed pattern is high.

### 5. Conclusions

In this paper, we proposed the embedding and detection method of invisible calibration pattern based on the human visual perception. In order to show the effectiveness of proposed method, we revealed the relation between pattern parameters and detection accuracy. In addition, we performed evaluation of visibility in the proposed method by subjective experiment.

### References


Age Estimation using Kernel Regression Analysis

Hironobu Fukai†, Hironori Takimoto‡, Yasue Mitsukura* and Minoru Fukumi**

†Department of Media Technology, Ritsumeikan University
1-1-1 Nojihigashi, Kusatsu, Shiga, 525-0058 Japan
Email: fukai@fc.ritsumei.ac.jp

‡Department of Communication Engineering, Okayama Prefectural University
111 kuboki, souja-shi, 719-1197 okayama
Email: takimoto@c.oka-pu.ac.jp

* Graduate School of Bio-Applications & Systems Engineering, Tokyo University of Agriculture & Technology
2-24-16 Naka-cho, Koganei-shi, Tokyo, 184-8588 Japan
Email: mitsu_e@cc.tuat.ac.jp

** Institute of Technology and Science, The University of Tokushima
2-1 Minami-josanjima, Tokushima, 770 Japan
Email: fukumi@is.tokushima-u.ac.jp

Abstract—In this paper, we propose age estimation using kernel regression analysis. Age estimation is one of the most difficult problems of facial recognition research areas. However, if age can be estimated by computer, it is possible to apply it to various fields, for example, marketing, communication person and machine, and so on. Then, there are many efforts, but practical use research is few. Furthermore, a necessary technology changes greatly also under the environment respectively, for example, use ID photo, use images of takes before laptop PC, use images that takes surveillance camera. We focus on images from surveillance camera, and we aim to propose a generalized age estimation method.

Therefore, we propose age estimation method using wrinkle and pigmented spot information that can be extracted age feature invariably even if appearance changes. To extract pigmented spot and wrinkle information, we use \( \epsilon \)-filter. Moreover, the kernel regression analysis is used for age estimation method. To evaluate effectiveness of the proposed, we simulate age estimation by using actual face image. As a result, age estimation error value is about 6.65 years old. This result is high accuracy as a generalized method.

1. INTRODUCTION

In our life, we estimate the person’s age roughly by using our experience. Furthermore, we meet the person a lot in daily life and we can take a smooth and flexible response routinely by estimating age. For example, if we saw the elder person, we behave politely. Therefore, it is considered that age is one of the most important characteristics of a person. However, it is easy for us to estimate our age roughly, but it’s difficult for the computer to do it. Therefore, age estimation methods based on images of face have been widely studied [1]-[10].

In a study on the change in the physical shape of the face with age, Todd et al. [1, 2] indicate that the contour of the skull can be approximated by a cardioid transform. Yamaguchi et al. [3] confirm that the differences between the features of an adult’s face and a child’s face include the length of the face and the ratio of each part. Age estimation by computer has also been performed. Kanno et al. [4] show that a male can be identified by neural networks representing four ages (12 years, 15 years, 18 years, and 22 years). Kwon and Lobo [5] reported that the proposed method has been implemented to classify input images into one of three age-groups: babies, young adults, and senior adults by using the placement information and texture information. However, almost of all their studies were based on cranio-facial development method and skin wrinkle analysis. Burt and Perrett [6] studied age perception using averaged faces of people from 25 to 60 years old used a method of focusing on face texture and shape. Ueki et al. [7] reported a method of age-group classification by linear discriminant analysis (LDA). Takimoto et al. [8] proposed a gender and age estimation technique that is not influenced by posture changes by estimating a NN from several features including the face texture and features. We proposed novel age estimation system [9]. In this method, we reported that the proposed method can estimate the apparent-age and gender by frequency feature of a face. Recently, practicable method is proposed. Ibara et al. [10] report age estimation using covariate shift adaptation. It’s robust for illumination variation. However, there are no methods that robust for face rotation and position.

Therefore, we propose an age estimation system that aims for practical use. To extract age feature, we use the \( \epsilon \)-filter. Moreover, the age is estimated continuously by kernel regression analysis. For the purpose of showing the effectiveness of the proposed method, computer simulations are performed using the actual data.
2. PROPOSED METHOD

In this section, we describe the procedure of the proposed method. The proposed method is constructed by 3 processings. The 1st processing is normalization of the facial image. The 2nd processing is age feature extraction. The 3rd processing is age estimation. We explain these 3 processings, respectively.

2.1. Normalization

It is necessary to normalize the face images for age estimation because the face area is different in facial image. Therefore, we detect facial area and skin area, and to normalize facial images.

2.2. Age Feature Extraction

It is known that the facial features are changed by aging. Especially, pigmented spot and skin wrinkle are well known.

2.2.1. Texture Feature

Pigmented spot and wrinkle on skin area are considered as minute change noise. Then, we use the ε-filter for extract this feature [11]. The ε-filter is defined as follows:

\[ y(n) = x(n) + \sum_{k=-N}^{N} a_k F(x(n-k) - x(n)) \]  

\[ |F(x)| \leq \varepsilon : -\infty < x < \infty. \]

There is a difference of the I/O signal below ε when it is a nonlinear function that \( F(x) \) shows in Fig.1. Example result of the ε-filter is shown in Fig.2. We subtract the result image of the ε-filter from original image and we make the histogram. Then, we normalize histogram by skin area. We adopt these histograms as texture features.

2.3. Age Estimation

We estimate age using kernel regression analysis. In this study, we estimate age using texture feature. Then, age(AGE) are estimated by following equation.

\[ AGE(x, w) = \sum_{n=1}^{N} w \phi(x_n) \]

where \( \phi(x_n) = (\phi(x_1), \ldots, \phi(x_D))^T \), and \( \phi(x_n) \) are basis functions. Moreover, parameters \( w \) are determined by minimizing a regularized sum-of-squares error function given by

\[ J(w) = \frac{1}{2} \sum_{n=0}^{N-1} [w^T \phi(x_n) - t_n]^2 + \frac{\lambda}{2} w^T w \]

\( \lambda \) means regularized parameter, and \( \lambda \geq 0 \). \( t_n \) is age values of training data sets. If we set the gradient of \( J(w) \) equal to zero, \( w \) is

\[ w = -\frac{1}{\lambda} \sum_{n=1}^{N} [w^T \phi(x_n) - t_n] \phi(x_n) = \sum_{n=1}^{N} a_n \phi(x_n) = \Phi^T a \]

If we substitute \( w = \Phi^T a \) into \( J(w) \), we obtain

\[ J(a) = \frac{1}{2} a^T \Phi \Phi^T a - \frac{1}{2} \Phi^T a \frac{1}{\lambda} a + \frac{1}{2} a^T \Phi^T a \]

where \( t = (t_1, \ldots, t_N)^T \). If we define the Gram matrix \( K = \Phi \Phi^T \), and \( k_n(x) = \phi(x_n^T, x) \), \( J(w) \) is

\[ J(a) = \frac{1}{2} a^T K K a - \frac{1}{2} a^T K t + \frac{1}{2} a^T K a. \]

Then, we obtain

\[ a = (K + \lambda I_N)^{-1} t. \]

Finally, we obtain the following prediction for a new input \( x \)

\[ AGE(x) = w \phi(x) = a^T \Phi \phi(x) = k(x)^T (K + \lambda I_N)^{-1} t. \]

Furthermore, we use liner kernel, Gaussian kernel and polynomial kernel.

3. COMPUTER SIMULATIONS

3.1. Face Image Database

The face database was provided from the Human and Object Interaction Processing (HOIP) organization in

---

Figure 1: Example of nonlinear function

Figure 2: Result of ε-filter

\[ y(n) = x(n) + \sum_{k=-N}^{N} a_k F(x(n-k) - x(n)) \]
Table 1: Detail of the face image database

<table>
<thead>
<tr>
<th>Size</th>
<th>640x480[pix.]</th>
</tr>
</thead>
<tbody>
<tr>
<td>24 bit color</td>
<td></td>
</tr>
<tr>
<td>Gender</td>
<td>150 images for each</td>
</tr>
<tr>
<td>Age</td>
<td>30 images per 5 years</td>
</tr>
<tr>
<td>Emotion</td>
<td>neutral</td>
</tr>
<tr>
<td>Rotation</td>
<td>H-45~+45, V-15~+15</td>
</tr>
</tbody>
</table>

Figure 3: HOIP database image

Japan [12]. The subject images comprised people with a wide range of ages. The background was made the same for all subjects. Furthermore, facial expression is neutral. Moreover, there are facial images of which it takes a picture from various directions. In this paper, the face database was used with permission from Softopia corporation, Japan. It is prohibited to copy, to use, and to distribute the images without the authorization of the copyright holder.

3.2. Conditions of Age Estimation

In order to show the effectiveness of the proposed method, we perform a simulation. In this paper, we use the actual data that is provided from HOIP organization in JAPAN [12], and we use a sample size of only 113 males, and it is only frontal face image, no-glasses. This is a first step of evaluation of the proposed method. Moreover, we use the $\varepsilon$-filter that window size is $7 \times 7$ and $\varepsilon$ value is 20. In this simulation, we use the leave-one-out cross-validation method.

3.3. Age Estimation Results and Discussions

Table 2,3,4 shows error average of estimated age. In addition, we show the age estimation error of each generation. Table 2 means result of liner kernel, Table 3 shows result of Gaussian kernel and Table 4 is result of polynomial kernel. From these results, age estimation error of 10’s, 20’s, and 60’s are larger than all generation’s average. Moreover, to compare 3 results, good result of average and generation’s average is polynomial kernel result. As a result, age estimation error value is about 6.65 years old. This result is high accuracy as a generalized method.

4. CONCLUSIONS

In this paper, we proposed an age estimation system based on texture feature. Firstly, we detect the face area and skin area, and normalize face image. Next, age feature is extracted by $\varepsilon$-filter. Finally, age is estimated by kernel regression analysis. From result, age estimation error value is about 6.65 years old. It’s good result of age estimation research area. However, there are no simulation of female and multi direction face image. It’s necessary to show the effectiveness of the proposed method.

References


Table 2: Age estimation error using liner kernel

<table>
<thead>
<tr>
<th>All generations</th>
<th>6.85 years</th>
</tr>
</thead>
<tbody>
<tr>
<td>10’s</td>
<td>8.77 years</td>
</tr>
<tr>
<td>20’s</td>
<td>6.37 years</td>
</tr>
<tr>
<td>30’s</td>
<td>4.66 years</td>
</tr>
<tr>
<td>40’s</td>
<td>5.9 years</td>
</tr>
<tr>
<td>50’s</td>
<td>6.3 years</td>
</tr>
<tr>
<td>60’s</td>
<td>10.32 years</td>
</tr>
</tbody>
</table>

Table 3: Age estimation error using Gaussian kernel

<table>
<thead>
<tr>
<th>All generations</th>
<th>6.64 years</th>
</tr>
</thead>
<tbody>
<tr>
<td>10’s</td>
<td>8.6 years</td>
</tr>
<tr>
<td>20’s</td>
<td>6.14 years</td>
</tr>
<tr>
<td>30’s</td>
<td>5.03 years</td>
</tr>
<tr>
<td>40’s</td>
<td>6.22 years</td>
</tr>
<tr>
<td>50’s</td>
<td>5.21 years</td>
</tr>
<tr>
<td>60’s</td>
<td>9.75 years</td>
</tr>
</tbody>
</table>

Table 4: Age estimation error using polynomial kernel

<table>
<thead>
<tr>
<th>All generations</th>
<th>6.65 years</th>
</tr>
</thead>
<tbody>
<tr>
<td>10’s</td>
<td>8.6 years</td>
</tr>
<tr>
<td>20’s</td>
<td>6.23 years</td>
</tr>
<tr>
<td>30’s</td>
<td>4.89 years</td>
</tr>
<tr>
<td>40’s</td>
<td>6.43 years</td>
</tr>
<tr>
<td>50’s</td>
<td>5.76 years</td>
</tr>
<tr>
<td>60’s</td>
<td>8.71 years</td>
</tr>
</tbody>
</table>


A Basic Fuzzy-Estimation Theory for Available Operation of Complicated Large-Scale Network Systems

Kazuo Horiuchi

Abstract—In this paper, we shall describe about a basic fuzzy-estimation theory based on the concept of set-valued operators, suitable for available operation of complicated large-scale network systems. Fundamental conditions for availability of system behaviors of such network systems are clarified in a form of \( \beta \)-level fixed point theorem for system of fuzzy-set-valued operators. Here, the proof of this theorem is accomplished by the concept of Hausdorff’ ball measure of non-compactness introduced into the Banach space.

1. Introduction

In extremely complicated large-scale network systems, precise evaluation and perfect control, and also ideal operation, of overall system behaviors cannot be necessarily expected by using any type of commonplace technologies for maintenance, which might be accomplished by simple measure in usual hierarchical network structures.

In order to effectively evaluate, control and maintain those complicated large-scale networks, as a whole, the author has recommended to introduce some connected-block structure: i.e., whole networks might be separated into several blocks which are carefully self-evaluated, self-controlled and self-maintained by themselves, and so, which are originally self-sustained systems. However, by always carefully watching each other, whenever they observe and detect that some other block is in ill-condition by some accidents, every block can repair and sustain that ill-conditioned block, through inter-block connections, at once. This style of maintenance of the system is sometimes called as locally autonomous, but the author recommends that only the ultimate responsibility on observation and regulation of whole system might be left for headquarter itself, which is organized over all blocks just as United States Government [1].

Here, let us consider Banach spaces \( X_i \) \((i = 1, \cdots, n)\) and \( Y_j \) \((j = 1, \cdots, n)\), and their bounded convex closed subsets \( X_i^{(0)} \) and \( Y_j^{(0)} \), respectively, corresponding to each block, Block \( i \) and Block \( j \) of whole network system. Let us introduce operators \( f_{ij} : X_i \to Y_j \) such that \( f_{ij}(X_i^{(0)}) \subseteq Y_j^{(0)} \) and let \( f_{ij} \) be completely continuous on \( X_i^{(0)} \).

For each block : Block \( i (i = 1, \cdots, n) \), dynamics of system behaviors can be represented originally by simple equations:

\[
x_i = a_i f_i(x_i), \quad (i = 1, \cdots, n),
\]

where \( a_i \) is a continuous operator: \( Y_j^{(0)} \to X_i^{(0)} \). These equations have solutions \( x_i^* \) in every \( X_i^{(0)}(i = 1, \cdots, n) \), according to the well-known Schauder’s type of fixed point theorem. Of course, these solutions represent original values of system behaviors.

\( f_i \) represents the original performance of the \( i \)-th block itself, \( f_{ij} \) represents the operation fed-back through all other blocks \((j \neq i)\) into the original \( i \)-th block, and \( f_{ji} \) represents inter-block connections from all other blocks, in order to repair and sustain the \( i \)-th block performance.

However, the fluctuation imposed on the actual system is nondeterministic rather than deterministic. In this case, even the effect due to a single cause is multi-valued, and the behavior is more naturally represented by a set of points, rather than a single point.

Therefore, it is reasonable to consider some suitable subset of the range of system behavior, in place of single ideal point, as target which the behavior must reach under influence of system control. Now, we can name it as an “available range” of the system behavior. Thus, by the available range, we mean the range of behavior, in which every behavior effectively satisfies good conditions beforehand specified, as a set of ideal behaviors. From such a point of view, the theory for fluctuation imposed on the system should be developed concerning the set-valued operator.

The author has given a series of studies on set-valued operators in functional analysis aspects, and has vigorously applied it to analysis of uncertain fluctuations of network systems [2], [3], [4].

Recently, the author gave a general type of fixed point theorem for the system of set-valued operator equations, in order to treat with extremely complicated large-scale network systems [1], [5], [7], [8].

Namely, let us introduce \( n \) set-valued operators \( G_i : X_i \times \Pi^j Y_j \to \mathcal{F}(X_i) \) (the family of all non-empty closed compact subsets of \( X_i \)) \((i = 1, \cdots, n)\), where \( \Pi^j Y_j \) means the direct product of \( n \) \( Y_j \)'s, for any \( j \in \{1, \cdots, n\} \), and \( \Pi^i Y_i \) means direct product of \( n \) \( Y_i \)'s, for fixed \( i \).
Under some natural conditions, the author presented important fixed point theorems on systems of set-valued operator equations:
\[ x_i \in G_i(x_i; f_{1i}(x_i), \ldots, f_{ni}(x_i); f_{1i}(x_1), \ldots, f_{ni}(x_n)), \quad (i = 1, \ldots, n). \]  
(2)

Proofs of fixed point theorems in ref. [1], [5] were accomplished by natural assumptions, on the other hand, the proof of the same theorem in ref. [7] was accomplished by a refined precise deduction, in weak topology, and the proof in ref. [8] was accomplished by the ball measure concept of non-compactness.

For convenience sake, let us define a direct product space \( Y_i \triangleq \prod^n Y_j \times \prod^n Y_i \) and also let \( Y_i^{(n)} \) be a non-empty bounded closed convex subset of \( Y_i \). Here, let us consider a vector \( v_i = (x_1, x_2, \ldots, x_n) \in V_i \) and an operator \( f_i(v_i) : V_i \to Y_i \) by
\[ f_i(v_i) = (f_{1i}(x_1), \ldots, f_{ni}(x_n); f_{1i}(x_1), \ldots, f_{ni}(x_n)). \]  
(3)

Here, we know that \( y_{ij} \triangleq f_{ij}(x_i) \in Y_j, y_{ij} \triangleq f_{ij}(x_i) \in Y_j \) and \( y_{ij} \triangleq (y_{ij_1}, \ldots, y_{ij_n}; y_{ij_1}, \ldots, y_{ij_n}) \in Y_i \). Therefore, we have a simple representation of the system of set-valued operators (2), as follows:
\[ x_i \in G_i(x_i; f_i(v_i)), \quad (i = 1, \ldots, n). \]  
(4)

On the other hand, the author recently presented a fixed point theorem for a general system of fuzzy-set-valued operator equations [6], under natural assumptions and with the proof similar to ref. [1] and [5].

Besides, the same fixed point theorem was proved precisely in weak topology in ref. [2].

Further, in this report, we will present a refined estimation theory of the fixed point theorem for such a general system of fuzzy-set-valued operator equations, with a more basic proof by the use of the ball measure of non-compactness.

2. Fuzzy Set and Fuzzy-Set-Valued Operator

First of all, let us consider a family of all fuzzy sets originally introduced by Zadeh [10], in a Banach space \( X \) with the norm \( \| \cdot \| \), and let any fuzzy set \( A \) be characterized by a membership function \( \mu_A(x) : X \to [0, 1] \). Now, we can consider an \( \alpha \)-level set \( A_\alpha \) of the fuzzy set \( A \) as \( A_\alpha \triangleq \{ x \in X \mid \mu_A(x) \geq \alpha \} \), for any constant \( \alpha \in (0, 1] \). The fuzzy set \( A \) is called compact, if all \( \alpha \)-level sets are compact for arbitrary \( \alpha \in (0, 1] \).

A fuzzy-set-valued operator \( G \) from \( X \) into \( X \) is defined by \( G : X \to \mathcal{F}(X) \), where \( \mathcal{F}(X) \) is a family of all non-empty , bounded and closed fuzzy sets in \( X \). If a point \( x \in X \) is mapped to a fuzzy set \( G(x) \), the membership function of \( G(x) \) at the point \( x \in X \) is represented by \( \mu_{G(x)}(x) \).

For convenience, let us introduce a useful notation: for an arbitrarily specified constant \( \beta \in (0, 1] \), a point \( x \) belongs to the \( \beta \)-level set \( A_\beta \) of the fuzzy set \( A \): \( x \in A_\beta \triangleq \{ x \in X \mid \mu_A(x) \geq \beta \} \) is denoted by \( x \in A_\beta \)

Here, let us introduce a new concept of \( \beta \)-level fixed point: for the fuzzy set \( G(x) \), if there exists a point \( x^* \) such that \( x^* \in A_\beta \), then \( x^* \) is called \( \beta \)-level fixed point of the fuzzy-set-valued operator \( G \).

Now, let us remember that we have introduced a new metric into the space of fuzzy sets [11, 12].

**Definition 1** Let us consider a Banach space \( X \). For any fixed constant \( \beta \in (0, 1] \), the \( \beta \)-level metric \( \rho_\beta \) between a point \( x \in X \) and a fuzzy set \( A \) is defined as follows:
\[ \rho_\beta(x, A) \triangleq \inf_{\beta \leq 1} d_{\alpha}(x, A), \]  
(5)

where
\[ d_{\alpha}(x, A) \triangleq \begin{cases} \inf_{y \in A_\alpha} \| x - y \| & \text{if } \alpha \leq \alpha_A, \\ \inf_{y \in A_{\alpha_A}} \| x - y \| & \text{if } \alpha > \alpha_A. \end{cases} \]  
(6)

Here, \( \alpha_A \triangleq \sup_{x \in X} \mu_A(x) \). And also, for any fixed constant \( \beta \in (0, 1] \), by means of the Hausdorff metric \( d_H \), the \( \beta \)-level metric \( d_H \) between two fuzzy sets \( A \) and \( B \) is introduced as follows:
\[ d_H(A, B) \triangleq \sup_{\beta \leq 1} D_{\alpha}(A, B), \]  
(7)

where \( D_{\alpha} \) is defined as
\[ D_{\alpha}(A, B) \triangleq \begin{cases} \inf_{y \in A_{\alpha}, y \in B} \| x - y \| & \text{if } \alpha < \alpha_A, \\ \inf_{y \in A_{\alpha_A}, y \in B_{\alpha_A}} \| x - y \| & \text{if } \alpha \geq \alpha_A. \end{cases} \]  
(8)

Here, \( \alpha_B \triangleq \sup_{x \in X} \mu_B(x) \) and the Hausdorff metric \( d_H \) between two sets \( S_1 \) and \( S_2 \) is defined by
\[ d_H(S_1, S_2) \triangleq \max\{ \sup_{x \in S_1} d(x_1, S_2), \sup_{y \in S_2} d(x_1, S_2) \}, \]  
where \( d(x, S) \triangleq \inf\{ \| x - y \| \mid y \in S \} \) is the distance between a point \( x \) and a set \( S \).

In order to give a new methodology for the discussion more sophisticated than the one by usual set-valued operators, the author presented mathematical theories based on the concept of \( \beta \)-level fixed point, by establishing fixed point theorems for \( \beta \)-level fuzzy-set-valued nonlinear operators which describe detailed characteristics of such fuzzy-set-valued nonlinear operator equations, for every level \( \beta \in (0, 1] \) [11, 12].
3. System of Fuzzy-Set-Valued Operator Equations

Now, let us introduce a more fine estimation theory for available operation of large-scale system of set-valued operators (2) and (4), by introducing \( \beta \)-level fuzzy estimation.

Originally, these sets are crisp. However, in order to introduce more fine estimation into these resultant fluctuations, here we can reconsider anew these sets \( G_i \) as fuzzy sets. Then, let us replace the above described crisp sets \( G_i(x_i; f_i(v_i)) \) by fuzzy sets with same notations, accompanied with suitable membership functions \( \mu_G(\xi_i), \xi_i \in X_i \), which should be properly introduced corresponding to conscious planning for the fine evaluation of resultant fluctuations themselves.

In order to realize a more precise analysis, let us introduce different values of \( \beta \) as \( \beta_i, (i = 1, \ldots, n) \), consciously selected corresponding to every block : Block \( i \).

Now, for any fixed constant \( \beta_i \in (0, 1] \), we can introduce a system of \( \beta_i \)-level fuzzy-set-valued nonlinear operator equations:

\[
x_i \in \beta_i \cdot G_i(x_i; f_i(v_i)), \quad (i = 1, \ldots, n).
\]  

If there exists a set of \( \beta_i \)-level fixed points \( x_i^* \) in \( X_i^{(0)} \) \((i = 1, \ldots, n)\), which satisfy the system of \( \beta_i \)-level fuzzy-set-valued operator equations (9), each \( x_i^* \) can be considered as a \( \beta_i \)-level likelihood behavior of Block \( i \), \((i = 1, \ldots, n)\). Here, this \( \beta_i \)-level likelihood behavior \( x_i^* \) can be found in a closed domain in which the membership function \( \mu_{G_i(x_i^*; f_i^*(v_i))}(\xi_i) \) has value larger than or equal to \( \beta_i \).

4. Fixed Point Theorem For System of \( \beta_i \)-level Fuzzy-Set-Valued Operators

Here, we will present a fixed point theorem for such a general system of \( \beta_i \)-level fuzzy-set-valued operator equations.

Now, let us introduce a series of assumptions:

**Assumption 1** Let the operator \( f_j : X_j^{(0)} \rightarrow f_j(x_j^{(0)}) \subset Y_j \) be completely continuous (continuous and compact).

**Assumption 2** Let the fuzzy-set-valued operator \( G_i : X_i^{(0)} \times Y_i \rightarrow \mathcal{F}(X_i) \) (a family of all non-empty compact subsets of \( X_i \)) satisfies the following Lipschitz condition with respect to the \( \beta_i \)-level metric \( \mathcal{H}_{\beta_i} \): that is, there are two kinds of constants \( 0 < k_i < 1 \) and \( h_i > 0 \) such that for any \( x_i^{(1)}, x_i^{(2)} \in X_i \), for any \( y_i^{(1)}, y_i^{(2)} \in Y_i \), \( G_i \) satisfies the inequality:

\[
\mathcal{H}_{\beta_i} \left( G_i(x_i^{(1)}, y_i^{(1)}), G_i(x_i^{(2)}, y_i^{(2)}) \right) \leq k_i \cdot \| x_i^{(1)} - x_i^{(2)} \| + h_i \cdot \| y_i^{(1)} - y_i^{(2)} \|. \tag{10}
\]

Now, we know that for any \( x_i \in X_i^{(0)} \) and \( f_i(v_i) \in Y_i \),

\[
G_i^{(0)}(x_i; f_i(v_i)) \triangleq G_i(x_i; f_i(v_i)) \subset X_i^{(0)} \neq \emptyset, \quad \text{and, moreover,}
\]

there exist projection points \( z_i^* \in X_i^{(0)} \) of arbitrary point \( x_i \in X_i^{(0)} \) upon the set \( G_i^{(0)}(x_i; f_i(v_i)) \) such that

\[
\| z_i^* - x_i \| = \min \left\{ \| z_i - x_i \| : z_i \in G_i^{(0)}(x_i; f_i(v_i)) \right\}, \tag{11}
\]

where \( G_i^{(0)} \triangleq \{ \xi \in X_i \mid \mu_G(\xi) \geq \beta_i \} \).

Then, we have the final result:

**Theorem 1** [Fixed Point Theorem] The system of \( \beta_i \)-level fuzzy-set-valued operator equations

\[
x_i \in \beta_i \cdot G_i^{(0)}(x_i; f_i(v_i^*)), \quad (i = 1, \ldots, n) \tag{12}
\]

has at least one fixed point \( x_i^* \in X_i^{(0)} \).

5. The Proof of the Fixed Point Theorem

In order to prove the fixed point theorem; Theorem 1, on a basic aspect of mathematical foundation, let us introduce Hausdorff’s ball measure of non-compactness \( \chi \), as follows: for the non-compactness of bounded subset \( S \) of real Banach space, Hausdorff’s ball measure \( \chi(S) \) is defined by\cite{13}

\[
\chi(S) \triangleq \inf \{ \varepsilon \geq 0 \mid S \text{ can be covered with a finite number of balls of radii smaller than } \varepsilon \}. \tag{13}
\]

Here, \( \chi(S) = 0 \) means that the closure of \( S \) is compact. Let us consider an arbitrary point \( x_i \in X_i \), and the corresponding arbitrary point \( y_i \in Y_i \). So, let us consider a bounded closed convex subset \( X_i^{(0)} \subset X_i \), and the corresponding bounded closed convex subset \( Y_i^{(0)} \subset Y_i \). Then, we have a lemma:

**Lemma 1** For an arbitrary point \( x_i \in X_i^{(0)} \), we have

\[
\chi(G_i^{(0)}(x_i; f_i(V_i^{(0)}))) = 0, \tag{14}
\]

where,

\[
G_i^{(0)}(x_i; f_i(V_i^{(0)})) \triangleq \bigcup_{v_i \in V_i^{(0)}} G_i^{(0)}(x_i; f_i(v_i)), \tag{15}
\]

Eq. (14) means that the convex closure of \( G_i^{(0)}(x_i; f_i(V_i^{(0)})) \)
belongs to \( \mathcal{F}(X_i) \).

Next, let us define:

\[
G_i^{(0)}(X_i^{(0)}; f_i(V_i^{(0)})) \triangleq \bigcup_{x_i \in X_i^{(0)}; v_i \in V_i^{(0)}} G_i^{(0)}(x_i; f_i(v_i)). \tag{16}
\]

Then, we have a lemma:

**Lemma 2**

\[
\chi(G_i^{(0)}(X_i^{(0)}; f_i(V_i^{(0)}))) \leq k_i \cdot \chi(X_i^{(0)}). \tag{17}
\]
Next, let us introduce some family of non-empty, convex, compact sets, invariant to the set-valued operator $G_i^{(0)}(x_i; f_i(v_i))$. For such a purpose, we shall prepare the following lemma:

**Lemma 3** Let $X^{(0)}$ be a bounded, closed and convex subset of the real Banach space $X$, and let the set-valued operator $F : X \to \mathcal{F}(X)$ be $k$-Lipschitz with respect to the non-compactness measure $\chi$, with a Lipschitz constant $k : (0 < k < 1)$: i.e., for any convex subset $A \subset X^{(0)}$, there exists a constant $k(0 < k < 1)$ such that

$$\chi(F(A)) \leq k \cdot \chi(A), \quad (0 < k < 1). \quad (18)$$

If, let us introduce a sequence $\{W_m\} (m = 0, 1, 2, \cdots)$ by the successive procedure such that $W_0 = X^{(0)}$, and $W_m \overset{\text{con}}{=} \text{conv} : F(W_{m-1}), (m = 1, 2, \cdots)$, then we have in turn

$$X^{(0)} = W_0 \supset W_1 \supset W_2 \supset \cdots \quad (19)$$

and

$$\chi(W_m) \leq k^m \cdot \chi(W_0). \quad (20)$$

When $m \to \infty$, $\chi(W_m) \to 0$, and if we put $W_\infty \overset{\text{def}}{=} \bigcap_{m=0}^{\infty} W_m$, then, $W_\infty$ is a non-empty, convex compact set, invariant to the set-valued operator $F$ such that

$$F(W_\infty) \subset W_\infty. \quad (21)$$

From this flow of deduction, now, we can refer the well-known Ky Fan’s fixed point theorem on upper semi-continuous set-valued operators, as follows:

**Lemma 4 (Ky Fan [14])** In a locally-convex topological linear space $X$, let $V$ be its non-empty convex compact subset. Let a set-valued operator $H : V \to \mathcal{H}(V)$ be a family of non-empty closed convex subsets of upper semi-continuous. Then, there exists a fixed point $x^*$ such that $x^* \in H(x^*) \subset V$.

As a result, by Ky Fan’s fixed point theorem, there exist fixed points $x_i \in \beta_i \cdot G_i^{(0)}(x_i; f_i(v_i))$ in all subsets $W_m$ with $V_m^{(0)}(i = 1, \cdots, n)$.

### 6. Concluding Remarks

Thus, the fluctuation analysis of this type of large-scale network systems, undergone by undesirable uncertain fluctuations, can be successfully accomplished at arbitrary fine-level of estimation, by immediate application of the here-presented fixed point theorem for system of $\beta_i$-level fuzzy-set-valued nonlinear operators, with consciously selected different value of parameter $\beta_i$, for every Block $i$.

In this paper, the fixed point theorem was proved basically by using the concept of Hausdorff’s ball measure of non-compactness.
Potential Games Based Coverage Control with Voronoi Partition

Saori Teraoka, Toshimitsu Ushio, Takafumi Kanazawa, and Naoki Hayashi

Graduate School of Engineering Science, Osaka University
1-3 Machikaneyama, Toyonaka, Osaka 560-8531 JAPAN
Email: [teraoa,na-hayashi]@hopf.sys.es.osaka-u.ac.jp and [ushio,kanazawa]@sys.es.osaka-u.ac.jp

Abstract—This paper presents sensor coverage control to cover a whole mission space and to maximize a sensing performance to detect targets. Suppose that each point in the mission space is covered by the nearest sensor. Then, a sensing area of each sensor is represented as a Voronoi partition. We introduce an objective function which represents the sensing performance based on the Voronoi partition and formulate the sensor coverage problem as an optimization problem. By introducing a barycentric coordinate over the mission space, we show that the sensor coverage problem can be transformed into a potential game. In potential games, local maximizers of a potential function are stable equilibrium points of the corresponding replicator dynamics. We propose distributed sensor coverage control based on the replicator dynamics to find the local maximizers of the objective function. Moreover, by simulation, we investigate the relation between a value function and stable equilibrium points of the replicator dynamics.

1. Introduction

Because of recent development of sensor network technologies and electronic devices, a group of autonomous sensors with computing, communication, and mobility capabilities is expected to perform a variety of distributed sensing tasks such as surveillance and environmental monitoring [1]. In such a sensor network, each sensor communicates with its neighbor sensors locally and decides its optimal placement based on the local information. So, the coverage control requires a decentralized control method for the optimal placement of a large number of sensors to achieve a high sensing performance [2, 3]. We consider a sensor coverage problem which places mobile sensors in the mission space to maximize an objective function. To solve such a maximization problem, Martínez et al. applied motion coordination such as swarming behaviors in biological groups [3].

On the other hand, the concept of potential games was introduced by Monderer and Shapley [4]. In potential games, all information about payoffs that is relevant to agents’ incentives can be captured in a scalar-valued function which is called a potential function. Therefore, the profitable strategy revisions increase a value of the potential function. Marden et al. proposed a control method based on potential games to maximize an objective function [5]. Hayashi et al. extended it to a power-aware sensor coverage problem [6].

In this paper, we consider a sensor coverage problem to cover the whole mission space and to maximize the sensing performance to detect targets in the mission space. We assume that each point in the mission space is covered by the nearest sensor. Then, a sensing area of each sensor is represented as a Voronoi partition. The objective function to maximize the sensing performance is considered as a potential function in a potential game. Then, we show that the sensor coverage problem can be represented as a potential game by introducing a barycentric coordinate over the mission space. Sandholm showed that all local maximizers of a potential function in potential games are stable equilibrium points of the corresponding replicator dynamics [7]. Based on the result, we use replicator dynamics to find a suboptimal position of each sensor.

The remainder of this paper is organized as follows. In section 2, we review the problem setting of the sensor coverage problem. We introduce a barycentric coordinate over the mission space and rewrite it as an optimization problem based on the barycentric coordinate in section 3. Section 4 shows that the sensor coverage problem can be transformed into a potential game where the barycentric coordinate corresponds to a mixed strategy. We also propose distributed optimal sensor coverage control based on replicator dynamics. In section 5, we show some simulations. Finally, we conclude this paper in section 6.

2. Sensor Coverage Problem

In this paper, we consider a sensor coverage problem to cover a whole mission space and maximize sensing performance. We model the mission space of $n$ mobile sensors as a convex polytope $Q \subset \mathbb{R}^2$ with $m$ vertices

$$v_1 = (v_{11}, v_{12})^T, v_2 = (v_{21}, v_{22})^T, \ldots, v_m = (v_{m1}, v_{m2})^T.$$  

Shown in Fig. 1 is an example of the convex polytope $Q$ with 5 vertices. Let $I = \{1, \ldots, n\}$ be a set of $n$ mobile sensors. Suppose that a position of each sensor $i$ is represented as $r_i = (r_{i1}, r_{i2})^T \in Q$. A value function $\phi: Q \rightarrow [0, \infty)$ is integrable and represents the relative importance of each point in the mission space $Q$. We assume that all sensors have the same performance. For sensor $i$ at the point $r_i \in Q$, its ability to detect a target originating at a point $q = (q_1, q_2)^T \in Q$ degrades with a distance $\|q - r_i\|$. This ability is measured by a performance function $h: [0, \infty) \rightarrow \mathbb{R}$ which is assumed to be nonincreas-
Figure 1: Example of a convex mission space $Q$ where each $v_i$ is a vertex ($i = 1, \ldots, 5$).

ing and piecewise-differentiable. With the value function $\phi$ and the performance function $h$, we introduce an objective function $H$ by

$$H(r) := \int_Q \max_{\|q - r\|} h(\|q - r\|) \phi(q) dq,$$  \hspace{2cm} (1)

where $r = (r^1, \ldots, r^n)$.

We assume that each point in the mission space is covered by the nearest sensor. Then, a sensing area of each sensor is represented as a Voronoi partition. Given $Q \subset \mathbb{R}^2$ and $r = (r^1, \ldots, r^n)$ of $n$ distinct points, the Voronoi partition of $Q$ generated by $r$ is the collection of sets $[v_1(r), \ldots, v_n(r)]$ defined by $v_i(r) := \{q \in Q \mid \|q - r^i\| \leq \|q - r^j\| \text{ for all } i \neq j, j \in I\}$ [3]. We refer to $v_i(r)$ as the Voronoi cell of $r^i$. Then, since the function $h(\|q - r\|)$ is monotonically non-increasing, the objective function $H$ is rewritten as follows:

$$H(r) = \sum_{i=1}^n \int_{v_i(r)} h(\|q - r^i\|) \phi(q) dq,$$  \hspace{2cm} (2)

where $v_i$ represents a sensing area of sensor $i$ for all $i \in I$.

3. Optimization Problem with a Barycentric Coordinate

All points in a convex polytope are represented by a barycentric coordinate [8]. In this section, we propose a representation of a position of sensor $i$ based on the barycentric coordinate. Let $x^i = (x^i_1, \ldots, x^i_m)^T$ be the barycentric coordinate of sensor $i$, where $x^i_k \geq 0$ and $\sum_{k=1}^m x^i_k = 1$ for all $i \in I$ and $k \in \{1, \ldots, m\}$. All sensors are placed in the mission space $Q$. The position $r^i$ is represented by the barycentric coordinate $x^i = (x^i_1, \ldots, x^i_m)^T$ as follows:

$$r^i(x^i) = \begin{bmatrix} v_{i1} \cdots v_{im} \\ x^i_1 \cdots x^i_m \end{bmatrix} \begin{bmatrix} x^i_1 \\ \vdots \\ x^i_m \end{bmatrix}, \text{ for all } i \in I. \hspace{2cm} (3)$$

where

$$\sum_{k=1}^m x^i_k = 1 \quad \text{for all } i \in I,$$

$$x^i_k \geq 0 \quad \text{for all } i \in I \text{ and } k \in \{1, \ldots, m\}.$$

Thus, the objective function (2) is a function of the barycentric coordinate as follows:

$$H(x) := H(r(x)) = \sum_{i=1}^n \int_{v_i(r(x))} h(\|q - r^i(x')\|) \phi(q) dq, \hspace{2cm} (4)$$

where $x = (x^1, \ldots, x^n)$ is the barycentric coordinate of $n$ sensors. Therefore, we can formulate the sensor coverage problem as the following optimization problem:

maximize $H(x)$
subject to $\sum_{k=1}^m x^i_k = 1 \quad \text{for all } i \in I,$
$x^i_k \geq 0 \quad \text{for all } i \in I \text{ and } k \in \{1, \ldots, m\}.$

For all sets $v \subset Q$, let $M_v$ and $C_v$ be as follows:

$$M_v := \int_v \phi(q) dq, \hspace{2cm} (5)$$
$$C_v := \frac{1}{M_v} \int_v q\phi(q) dq. \hspace{2cm} (6)$$

$C_v$ is called a centroid of a set $v$. If a combination of sensor positions $r$ satisfies Eq.(7), $r$ is a local maximizer of objective function (2).

$$r^i = C_{v_i} \quad \text{for all } i \in I. \hspace{2cm} (7)$$

4. Potential Game for Sensor Coverage Problem

In potential games, local maximizers of a potential function are stable equilibrium points of the corresponding replicator dynamics. Therefore, we formulate the sensor coverage problem as a potential game, and search local maximizers of the objective function $H$.

We consider sensor $i$ as population $i$ in a potential game and a pure strategy $k \in S = \{1, \ldots, m\}$ is to locate a sensor at a vertex $v_k$ of the mission space. Suppose that $S$ is the common set of pure strategies of all populations. Then, the barycentric coordinate $x^i$ of sensor $i$ can be considered as a population state of population $i$. Note that $\sum_{k=1}^m x^i_k = 1$ for all $i \in I$. Moreover, we consider the objective function $H$ as the potential function in the potential game. Therefore, introducing a payoff function $F_k^i(x)$ defined by

$$F_k^i(x) = \frac{\partial H}{\partial x_k}(x) \quad \text{for all } i \in I \text{ and } k \in S, \hspace{2cm} (8)$$

we can transform the sensor coverage problem into an $n$ population potential game, which has $m$ vertices of the mission space $Q$ as the pure strategies, the barycentric coordinate $x^i$ as the population state for population $i$, and the objective function $H(x)$ as the potential function.
For simplicity, we assume that \( h(d) = -d^2 \). Let
\[
J_{r, y} := \int y |q - r|^2 \phi(q) dq.
\]
(9)
Then, we have the following equation:
\[
J_{r, y} = J_{C_{r, y} + M_v } \| r - C_{v} \|^2.
\]
(10)
and the objective function (4) is rewritten as follows:
\[
H(x) = - \sum_{i=1}^{n} J_{C_{v_i}} - \sum_{j=1}^{m} M_v \| r_j - C_{v_j} \|^2.
\]
(11)
Thus, the payoff function \( F_k^i(x) \) is given by
\[
F_k^i(x) = \frac{\partial H}{\partial x_k^i}(x)
\]
\[
= \frac{\partial}{\partial x_k^i} \left(- \sum_{j=1}^{n} M_v \left\| \sum_{l=1}^{m} x_l^j v_l - C_{v_j} \right\|^2 \right)
\]
\[
= -2M_v \left\{ v_k \left( \sum_{l=1}^{m} x_l^j v_l - C_{v_j} \right) + v_k \left( \sum_{l=1}^{m} x_l^j v_l - C_{v_j} \right) \right\}.
\]
(12)
We introduce replicator dynamics to search the local maximizers of the potential function. Suppose that the increase rate of agents with strategy \( k \in S \) is proportional to the difference between the payoff \( F_k^i(x) \) and an average payoff of population \( i \). Then, replicator dynamics is given as follows [9]:
\[
\dot{x}_k^i = x_k^i \left( F_k^i(x) - \sum_{l=1}^{m} x_l^i F_l^i(x) \right).
\]
(13)
By substituting the payoff function (12) for Eq. (13), we have
\[
\dot{x}_k^i = -2M_v x_k^i \left\{ v_k \left( \sum_{l=1}^{m} x_l^j v_l - C_{v_j} \right) + v_k \left( \sum_{l=1}^{m} x_l^j v_l - C_{v_j} \right) \right\}
\]
\[
- \sum_{j=1}^{m} x_k^j \left\{ v_k \left( \sum_{l=1}^{m} x_l^j v_l - C_{v_j} \right) + v_k \left( \sum_{l=1}^{m} x_l^j v_l - C_{v_j} \right) \right\}.
\]
(14)
The stable equilibrium points of replicator dynamics (14) are local maximizers of the potential function (11). Thus, we can search the optimal position of each sensor to maximize the objective function using Eq. (13).

5. Simulation

We consider that 5 sensors cover a square mission space \( Q = [0, 100] \times [0, 100] \) with the following value function \( \phi(q) \):
\[
\phi(q) = \exp(-((q_1 - 80)^2 + (q_2 - 70)^2)/A),
\]
(15)
where \( A \) is a parameter. Note that \( \phi(q) \) is a one-hump function whose top is \((80, 70)\). The smaller \( A \) is, the steeper its top is. As \( A \) goes to the infinity, \( \phi \) converges to \( \phi(q) = 1 \). Shown in Figs. 2(a) and 2(b) are stable equilibrium points of the replicator dynamics when \( \phi(q) = 1 \). For \( A = 12000 \), these equilibrium points are moved as shown in Figs. 2(c) and 2(d). For \( A = 10000 \), however, the two equilibrium points coincide and we observe one Voronoi partition as shown in Fig. 2(e). In other words, a pitchfork bifurcation occurs. For any \( A \) less than 10000, we observe a unique stable equilibrium point which is a globally optimal sensor location. Shown in Fig. 2(f) is the stable equilibrium point for \( A = 4000 \). Moreover, shown in Fig. 3 is a bifurcation diagram for the two equilibrium points where the vertical and the horizontal axes are \( q_1 \) and \( A \), respectively. As \( A \) is decreased, the two equilibrium points are closer and finally coincide. Shown in Fig. 4 is the relation between the parameter \( A \) and values of the objective function at the equilibrium points. As \( A \) is decreased, the value function is steeper and the sensors move to its top to make the value of the objective function higher. Therefore, the value of the objective function increases. From this simulation, it is shown that, if the value function is enough steep, the replicator dynamics has the unique stable equilibrium point which corresponds to the optimal sensor location. Such a bifurcation phenomenon is useful to obtain a globally optimal sensor location for a specified parameter \( A^* \). First, we set \( A \) to be small enough to have the unique stable equilibrium point and obtain it using the replicator dynamics. Next, we increase \( A \) and set several initial points around it. If their trajectories converge to different equilibrium points, we select one which optimizes the objective function. Then, we repeat this procedure until \( A \) becomes the specified value \( A^* \), for which we obtain the optimal sensor location. It is future work to evolve this idea.

6. Conclusion

We discussed an application of a potential game to a sensor coverage problem using a barycentric coordinate. We showed that the optimal position of each sensor to maximize the objective function can be obtained by replicator dynamics. It is our future work to investigate global bifurcation properties of sensor positions obtained by replicator dynamics. It is also our future work to discuss improvements of our approach to search for the global optimal sensor location.

Acknowledgments

This work was partially supported by Global COE (Center of Excellence) Program, Grant-in-Aids for Scientific Research (C), No. 21560464 and Young Scientists (B), No. 21760330, and JSPS Fellows, No. 21-3450, from the MEXT of Japan.
Figure 2: Stable equilibrium points: For $A = 1$ (resp. 12000), there are two equilibrium points (a) and (b) (resp. (c) and (d)) and, for $A = 10000$ (resp. 4000), there is a unique one (e) (resp. (f)).

References

Convergence Analysis of Discrete-Time Multi-Agent Systems Based on Sequential Connectivity

Yao Chen†⋆, Jinhu Lü⋆‡, Daniel W. C. Ho†, and Xinghuo Yu‡

Abstract—It is well known that the jointly connectivity and sequential connectivity are two fundamental concepts in multi-agent systems. We discover that the jointly connectivity is equivalent to sequential connectivity if the communication topology has self-loops at each node. Based on the above result, the consensus of discrete-time multi-agent systems is then given under the condition of jointly connectivity by constructing a sequentially connected sequence. The above proof greatly simplified the former proofs.

1. Introduction

A multi-agent system (MAS) is a system consisted of multiple interacting intelligent agents. Examples include birds flocks, sheep herds, fish schools, online trading, disaster response, multi-robot coordination, economic systems, and so on. Over the past decades, numerous models are proposed to characterize and analyze MAS, including Vicsek model ([11]), Boid model ([12]), linear iteration model ([3], [4], [9], [7]), and so on. Recently, MAS has received an increasing attention from mathematics, physics, engineering sciences, and social communities ([6], [8], [10]).

As one of the simplest MAS models, the linear iteration model updates the state of each agent by using the linear average of all states of its neighbors ([7], [11], [12]). Some known results have indicated that all states of the linear iteration model will converge to some identical value under the condition of jointly connectivity ([4], [7], [5], [9]) or sequential connectivity ([10]). As pointed out in [10], the sequential connectivity is a much more stronger condition than the jointly connectivity and the joint connectivity does not implies the sequential connectivity. In this paper, we will bridge the gap between the sequential connectivity and the jointly connectivity under some suitable conditions.

According to [4], [5], and [9], the linear MAS can converge exponentially under the condition of jointly connectivity. It should be pointed out that the above results are obtained based on the theory of infinite matrices products introduced by Wolfowitz in [3]. From [3], the convergence condition of the products of infinite matrices is that there exists a uniformly lower bound for the nonzero entries of all matrices. However, in this paper, we will prove that the above uniformly lower bound condition is not always necessary for the consensus of the linear MAS.

In [7], Blondel and his colleagues introduced a new approach to prove the consensus of discrete-time MAS without using the matrices products. Unfortunately, the above method cannot be generalized to the case of joint graph with a spanning tree. To overcome the uniformly lower bound condition of matrices products approach in [3] and the limitation of method in [7], a new technique is proposed to prove the consensus of discrete-time multi-agent systems under the same condition of jointly connectivity in this paper, called sequential connectivity approach.

This paper is organized as follows. Section 2 introduces several necessary preliminaries on graph theory and matrix theory. The problem is formulated in Section 3. In Section 4, a novel approach is then given to prove the consensus of discrete-time MAS. Finally, some concluding comments or suggestions are given in Section 5.

2. Preliminaries

A graph \( G = (V, E) \) is composed of two sets \( V \) and \( E \), where \( V = \{1, 2, ..., N\} \) is the set of nodes and \( E \subseteq V \times V \) is the set of edges. There exists a path from node \( i \) to \( j \) if and only if there exist \( k \) different nodes \( \{i_s\} \) with \( 1 \leq s \leq k, i_1 = i, i_k = j \) satisfying \( (i_p, i_{p+1}) \in E \) for any \( 1 \leq p \leq s - 1 \). Denote \( (i, j) \in G \) if there exists a path from node \( i \) to \( j \).

In graph \( G \), if there exists a node \( i_0 \) which has paths to any other nodes, then the graph \( G \) contains a spanning tree with the root \( i_0 \). For the above graph \( G \), if there exist paths from any node \( i \in V \) to any node \( j(\neq i) \in V \), then the graph is called strongly connected.

For graph \( G = (V, E) \) and \( S \subseteq V \). Define

\[ N(S, G) = \{ j \in V : \exists i \in S, (i, j) \in E \} \]

be the set of neighbors of \( S \).

For different graphs \( G_k = (V, E_k) \) \((1 \leq k \leq K)\) with the same set of vertices \( V \), the union of these \( K \) graphs is
defined as
\[
\bigcup_{k=1}^{K} G_k = (V, \bigcup_{k=1}^{K} E_k).
\]

A matrix is called nonnegative if each of its entry is nonnegative. A nonnegative matrix is called stochastic if the sum of each row equals to 1. A matrix is 0-1 if each entry is 0 or 1. For two matrices \(A = (a_{ij})_{m \times n}\) and \(B = (b_{ij})_{m \times n}\), define \(A \circ B = (\max\{a_{ij}, b_{ij}\})_{m \times n}\). At the same time, define
\[
\prod_{i=1}^{K} A_i = A_K A_{K-1} \cdots A_1 \text{ to be the left products of matrices.}
\]

For each non-negative matrix \(A = (a_{ij})_{m \times n}\), a graph \(G = (V, E)\) with \(E \subseteq V \times V\) is given to characterize the structure of \(A\), where \(V = \{1, 2, \ldots, N\}\) and \(a_{ij} > 0\) if \((i, j) \in E\).

Hereafter, two fundamental concepts are given as follows.

Definition 1 [4] A sequence of graphs \(\{G_k\}_{k=1}^{K}\) with vertices set \(V\) is called sequential connectivity if there exist sets \(S_k \subseteq V\) such that
\[
S_{k+1} \subseteq N(S_k, G_k)
\]
hold for any \(1 \leq k \leq K\). \(S_1\) is a singleton, and \(S_{K+1} = V\). An infinite sequence of graphs \(\{G_k\}_{k=1}^{\infty}\) with vertices set \(V\) is called sequential connectivity if there exists an increasing integer sequence \(\{t_k\}_{k=1}^{\infty}\) such that \(\{G_k\}_{t_k}^{t_{k+1}-1}\) is sequential connectivity for any \(k \geq 1\).

Definition 2 [10] A sequence of graphs \(\{G_k\}_{k=1}^{K}\) with vertices set \(V\) is called jointly connectivity if \(\bigcup_{k=1}^{K} G_k\) contains a spanning tree. An infinite sequence of graphs \(\{G_k\}_{k=1}^{\infty}\) with vertices set \(V\) is called jointly connectivity if there exists an increasing integer sequence \(\{t_k\}_{k=1}^{\infty}\) such that \(\{G_k\}_{t_k}^{t_{k+1}-1}\) is jointly connectivity for any \(k \geq 1\).

According to the above definitions of these two kinds of connectivity, one knows that if \(\{G_k\}_{k=1}^{K}\) is sequential connectivity, then it is also jointly connectivity, but not vice versa. It can be seen from the following examples.

Let
\[
A_1 = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix},
\]
\[
A_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix}
\]

It is easy to verify that
\[
S_1 = \{1\}, \quad N(S_1, G(A_1)) = \{2\}
\]
\[
S_2 = \{2\}, \quad N(S_2, G(A_2)) = \{1, 3\}
\]
\[
S_3 = \{1, 3\}, \quad N(S_3, G(A_3)) = \{1, 2, 3\}
\]
\[
S_4 = \{1, 2, 3\}
\]

Thus \(\{G(A_i)\}_{i=1}^{3}\) is sequential connectivity.

Similarly, denote
\[
A_1 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}
\]
\[
A_3 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}
\]

It is easy to verify that
\[
A_1 \circ A_2 \circ A_3 = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}
\]

Hence, \(\{G(A_i)\}_{i=1}^{3}\) is jointly connectivity. Moreover, \(\{G(A_i)\}_{i=1}^{3}\) is not sequential connectivity.

3. Formulation of the Problem

Consider an MAS consisting of \(N\) autonomous agents, let \(V = \{1, 2, \ldots, N\}\) be the set of these \(N\) agents. Let \(x_i(t)\) be the state of agent \(i\) at time \(t\). The updating rule of the above states is described by
\[
x_i(t+1) = \sum_{j=1}^{N} a_{ij}(t) x_j(t),
\]
where \(t \geq 0, a_{ij}(t) \geq 0,\) and \(\sum_{j=1}^{N} a_{ij}(t) = 1\). Denote
\[
A(t) = (a_{ij}(t))_{i,j=1}^{N}.
\]

An interesting question is: What conditions can guarantee the consensus of all states in (1)? That is, \(|x_i(t) - x_j(t)| \to 0\) as \(t \to \infty\) for any \(i, j \in V\).

For the MAS (1), some necessary assumptions are given as follows:

(A1) For some integer \(T > 0\), there exists a sequence of nonnegative integers \(\{t_k\}\) with \(t_1 = 0\) satisfying \(0 < t_{k+1} - t_k \leq T\) for any \(k \geq 1\).

(A2) The graph \(G_k = \bigcup_{t=t_k}^{t_{k+1}-1} G(t)\) contains a spanning tree \(T_k\).

(A3) There exists some \(\alpha \in (0, 1]\) such that \(A(t) \geq \alpha D(t)\) and \(G(\prod_{t=t_k}^{t_{k+1}-1} D(t)) = T_k\) with \(T_k\) defined in (A2).

(A3.1) There exists some \(\alpha \in (0, 1]\) such that \(\inf_{t \geq 0} a_{ij}(t) \geq \alpha\) and \(\inf_{v \leq t \geq 0} a_{ii}(t) \geq \alpha\).

Also, for the MAS (1), denote
\[
m(t) = \min_{v \in V} x_v(t),
\]
\[
M(t) = \max_{v \in V} x_v(t),
\]
\[
\Delta(t) = M(t) - m(t),
\]
Remark 1 Assumption (A3-1) has been widely used in Refs. [4], [5], [7], [9], and [12]. It should be pointed out that this assumption requires that all nonzero entries have a uniformly nonzero lower bound. However, according to (A3), the above uniformly lower bound condition is not necessary for the consensus of MAS (1) under the condition that the weights of those edges corresponding to the spanning tree $T_k$ have a nonzero lower bound.

4. Convergence Analysis Based on Sequential Connectivity

In this section, one discovers that the jointly connectivity is equivalent to sequential connectivity if the communication topology has self-loops at each node. Moreover, the consensus of MAS (1) will be proved under the condition of jointly connectivity by constructing a connective sequence.

Lemma 1 Given a sequence of $N \times N$ non-negative matrices $\{A_i\}_{i=1}^K$ with positive diagonal entries and $K \geq K_N$, where $K_N$ is given by

$$K_N = N \cdot K_{N-1} + 1, \quad K_2 = 1.$$ 

If each $G(A_i)$ contains a spanning tree, then $\{G(A_i)\}_{i=1}^K$ is sequential connectivity.

Proof: Suppose that graph $G$ contains a spanning tree. Delete all redundant edges in $G$ and make the left graph $G'$ be a spanning tree. Thus there exists some node $i$ in $G'$ with out-degree 0 and in-degree 1. Based on the above process, $G$ is called $i$-deletable.

Since each $A_i$ is positive-diagonal, it is easy to verify that for any $S \subseteq \{1, 2, ..., N\}$, one gets

$$S \subseteq N(S, G(A_i))$$

for any $1 \leq i \leq K$.

Use induction, it is obvious that the result of this lemma holds for the case of $N = 2$ and $K \geq 1$.

Assume that the case of $N$ holds. For the case of $N + 1$, given $(N + 1)K_N + 1$ matrices $\{A_i\}$ with dimension $N + 1$. According to the pigeonhole principle, there exists some node $i \in \{1, 2, ..., N + 1\}$ and a subsequence $\{A_{i_j}\} \subseteq \{A_i\}$ with elements not less than $K_N + 1$ such that any graph in $\{G(A_{i_j})\}$ is $i$-deletable.

Without loss of generality, denote $i = N + 1$. From the assumption of case $N$, there exists $S_k \subseteq \{1, 2, ..., N\}$ with $1 \leq k \leq K_N$ satisfying

$$S_{k+1} \subseteq N(S_k, G(A_{i_k})).$$

where $S_1$ is a singleton and $S_{K_N+1} = \{1, 2, ..., N\}$.

Since $G(A_{K_N+1})$ contains a spanning tree and is also $(N + 1)$-deletable, then one has

$S_{K_N+2} = N(\{1, 2, ..., N\}, G(A_{K_N+1})) = \{1, ..., N, N + 1\}.$

Therefore, the case of $N + 1$ holds. Consequently, the above result holds for any $N \geq 2$.

Lemma 2 Given a sequence of $N \times N$ 0-1 matrices $\{D_i\}_{i=1}^K$, where $D_i$ has positive diagonal entries. If $\{G(D_i)\}_{i=1}^K$ is jointly connectivity, then

$$\mathcal{G}(D_KD_{K-1} \cdots D_1)$$

contains a spanning tree.

Proof: According to the assumptions, one has

$$D_KD_{K-1} \cdots D_1 \geq D_K \circ \cdots \circ D_2 \circ D_1.$$ 

The left proof is obvious and hence omitted here.

From Lemmas 1 and 2, the sequential connectivity and jointly connectivity are equivalent if each graph in the sequence has self-loops. Therefore, the following theorem bridges the gap between the above two kinds of connectivity.

Theorem 1 Given an infinite sequence of graphs $\{G_k\}_{k=1}^{\infty}$ with the same set of vertices $V$. If each node in $G_k$ has a self-loop, then the jointly connectivity is equivalent to the sequential connectivity.

Proof: The proof is omitted here since it can be derived easily from Lemmas 1 and 2.

Lemma 3 Given a sequence of $N \times N$ stochastic matrices $\{A_i\}_{i=1}^K$, where $A_i \geq a P_i$ and $P_i$ is a 0-1 matrix. Let $x(t + 1) = A_i x(t)$. If $\{G(P_i)\}_{i=1}^K$ is sequential connectivity, then one obtains

$$\Delta(K + 1) \leq (1 - a^K) \Delta(1).$$

Proof: For the sequence of matrices $\{P_i\}_{i=1}^K$, according to the definition of sequential connectivity, there exists $S_k \subseteq V$ such that

$$S_{k+1} \subseteq N(S_k, G(P_k)).$$

where $S_1$ is a singleton and $S_{K+1} = \{1, 2, ..., N\}$. Denote

$$M_t^* = \max_{i \in S} x_i(t), \quad m_t^* = \min_{i \in S} x_i(t).$$

For $t \geq 1$, if $i \in S_{t+1}$, then one has

$$x_i(t + 1) = \sum_{j=1}^N a_{ij}(t)x_j(t)$$

$$\leq \sum_{j \in S_i} a_{ij}(t)x_j(t) + M(t) \sum_{j \notin S_i} a_{ij}(t)$$

$$\leq \sum_{j \in S_i} a_{ij}(t)x_j(t) + M(t)(1 - \sum_{j \in S_i} a_{ij}(t))$$

$$\leq M_t^* \sum_{j \in S_i} a_{ij}(t) + M(t)(1 - \sum_{j \in S_i} a_{ij}(t)).$$
Since $\sum_{j \in S} a_{ij}(t) \geq \alpha$ and $M(t) \leq M(1)$ for $t \geq 1$, then one has

$$M_{t+1}^* \leq \alpha M_t^* + (1 - \alpha)M(1).$$

By using iteration, one gets

$$M_{K+1}^* \leq \alpha^K M_1^* + (1 - \alpha^K)M(1).$$

Similarly, one deduces

$$m_{K+1}^* \geq \alpha^K m_1^* + (1 - \alpha^K)m(1).$$

According to $S_{K+1} = \{1, 2, ..., N\}$ and $m_1 = M_1^*$, one obtains

$$\Delta(K + 1) \leq (1 - \alpha^K) \Delta(1).$$

\begin{proof}

Denote

$$H_t(k) = A(t_{k+1} - 1) \cdots A(t_k).$$

From (A3), one has $H_t(k) \geq \alpha^T D(t_{k+1} - 1) \cdots D(t_k)$. Moreover, by Lemma 2, $G(H_t(k))$ contains a spanning tree, where each node has a self-loop. According to Lemma 1, $\{G(H_t(i))\}_{i=k_{Kk}}^{Kk} = \{G(H_t(k))\}_{i=k_{Kk}}^{Kk}$ is sequential connectivity. By Lemma 3, one gets

$$\Delta(t_{kKk+1}) \leq (1 - \alpha^{TKk}) \Delta(t_{k(k-1)Kk+1}).$$

Consequently, one obtains

$$\Delta(t_{kKk+1}) \leq (1 - \alpha^{TKk}) \Delta(t_{k(k-1)Kk+1}) \leq (1 - \alpha^{TKk}) \Delta(t_{k(k-1)Kk+1}) \leq \cdots \leq (1 - \alpha^{TKk}) \Delta(t_i).$$

Therefore, $\lim_{k \to \infty} \Delta(t_{kKk+1}) = 0$ and $\lim_{t \to \infty} \Delta(t) = 0$. That is, the MAS (1) can reach consensus.

\end{proof}

5. Concluding remarks

This paper has further investigated the inner relationship between the sequential connectivity and jointly connectivity. That is, the jointly connectivity is equivalent to sequential connectivity if the communication topology has self-loops at each node. Based on the above result, the consensus of discrete-time multi-agent systems is proved under the condition of jointly connectivity by constructing a connective sequence. The above proof greatly simplified the former proofs. Some real-world applications will be further explored in the near future.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grants 60821091 and 60772158, the National Basic Research (973) Program of China under Grant 2007CB310805, the Support of K. C. Wong Education Foundation of Hong Kong, and the ARC Future Fellowships under Grant FT0992226.

References


Equatorial Climate Data Analysis and Forecasting by Singular Spectrum Analysis

Naoki Itoh † and Jürgen Kurths
†Center for Dynamics of Complex Systems
University of Potsdam, D-14476 Potsdam, Germany
Email: naoki.itoh@gmail.com

Abstract—This paper describes the techniques of Singular Spectrum Analysis (SSA) and forecasting by the Linear Recurrent Formulae (LRF), which are applied to monthly precipitation and lake-sediments in Kenya. By the SSA algorithm, information, such as a trend, seasonal periodicities, anomaly cycles, and noise, can be extracted from these data series. And then from the results obtained by the SSA, it is possible to forecast the data which are assumed to be governed by the LRF. The goal in this paper is to investigate the properties of the equatorial climate changes, where the global warming is remarkable.

1. Introduction

Climate problems through the term global warming have received a lot of attention in recent years. Interest in the topics spans not only meteorologist and geologist but also physicist, economist and so on as interdisciplinary subjects.

Data observed in nature generally consist of complicated components such as exogenous and endogenous factors. These data series have successfully been analyzed by a number of statistical tools. If such noisy data are, for instance, analyzed for a forecast, the noise has to be reduced from it. In other words, the underlying deterministic dynamics in the data will be extracted by the noise reduction. In some of the previous research the modelling and forecasting have often been performed by a linear model. However, since most of the actual data series are currently well-known as a nonlinear nature, it is necessary to consider both the linear and nonlinear models for the modelling and forecasting. As one of the ideal methods, Singular Spectrum Analysis (SSA) is powerful and useful, and it is especially applicable for the analysis of time series with complex seasonal components and non-stationarity, i.e., it is not necessary to assume stationarity of the series or normality of the residuals. The technique is defined as a nonparametric technique of the time series analysis including the statistical tools such as the classical analysis, dynamical system, signal processing, and so on. Another advantage of this method is that it can well be applied to small sample sizes.

An early study of the SSA is described in the papers by Broomhead and King [1]. Then, the idea of the SSA is independently developed in several groups in Russia, UK, and USA. Especially, the theoretical and practical foundations of this technique are described in the book by Golyandina, Nekrutkin, and Zhigljavsky in the Russian group (2001) [5].

The purpose of SSA is to decompose the original data series into some components with useful and interpretable information (e.g. a slow trend, oscillatory components, and a structureless noise). Their decomposed components are of substantial importance for time series forecasting by the Linear Recurrent Formulae (LRF).

In this paper the Caterpillar-SSA proposed by Golyandina et al. (2001), is applied to data about precipitation and lake-sediments in Kenya in order to compare present climate changes with a paleoclimate and to forecast the new data point by using the LRF [2–7, 9–16].

2. Singular Spectrum Analysis

SSA aims to decompose the observed data series into some meaningful subseries, which are in general identified as a slowly varying trend, harmonic (periodic and quasi-periodic) components, and noise. These kinds of components show essential properties of observed data.

The algorithm of the method generally falls into two stages, the first stage: decomposition and the second stage: reconstruction, and then each of them has two following separate steps: embedding (step 1) and Singular Value Decomposition (SVD) (step 2) in the first stage, and grouping (step 1) and diagonal averaging (step 2) in the second stage. The reconstructed data series are then used for forecasting.

2.1. First Stage: Decomposition

Embedding of the step 1 in the first stage is to transfer a one-dimensional series \( Y = (y_1, \ldots, y_N) \) into the multidimensional series \( \{X_1 : \cdots : X_K\} \) with vectors \( X_j = (y_j, \ldots, y_{j+L-1})^T \in \mathbb{R}^L \) \((j = 1, \ldots, K)\), where \( K \) and \( L \) which are defined as the integer parameters for the SSA, can be described by \( K = N - L + 1 \), thus the SSA has substantially the single parameter \( L \) of the embedding, called window length, restricted by \( 2 \leq L \leq N/2 \). The matrix which consists of the vectors \( X_j \) is defined as the trajectory matrix, \( X = [X_1 : \cdots : X_K] = (x_{i,j})_{i,j=1}^{L,K} \). Since the trajectory matrix \( X \) is a Hankel matrix, all the elements along the diagonal \( i + j = \text{const} \) are equal [8].
In the step 2, the Singular Value Decomposition (SVD) is applied to the trajectory matrix, which can then be written as \( X = X_1 + \cdots + X_d \), where \( X_i = U_i \sqrt{\lambda_i} V_i^T \) (\( i = 1, \cdots, d \)) defined as a rank-one orthogonal elementary matrix. The collection \( \{ \sqrt{\lambda_i}, U_i, V_i \} \), which are a singular value, empirical orthogonal functions (EOFs), and principal components (PCs), respectively, is called the \( l \)-th eigentriple of the matrix \( X \).\( d \) is the number of non-zero singular values (i.e., \( \sqrt{\lambda_1} \geq \cdots \geq \sqrt{\lambda_d} > 0 \)). The relationship among the terms in the collection can be described by \( V_1 = X_1 U_1 / \sqrt{\lambda_1} \).

The fact that \( \sum_{i=1}^{d} (\sqrt{\lambda_i})^2 \) is equal to the squared Frobenius-Perron norm of the trajectory matrix \( X \), and also \( \sqrt{\lambda_i} \) measures the degree of approximation of the trajectory matrix, that is, it shows a contribution of the elementary matrices to the trajectory matrix.

2.2. Second Stage: Reconstruction

So called **eigentriple grouping** will be performed so that the index set \( \{1, \cdots, d\} \) of the elementary matrices is reformulated into \( m \) disjoint subsets \( I_1, \cdots, I_m \):

\[
X = X_{I_1} + \cdots + X_{I_m} = X_{I_1} + \cdots + X_{I_m},
\]

(1)

with \( X_{I_h} = X_{I_h} + \cdots + X_{I_h}, \ h \in \{1, \cdots, d\}. \) These represented matrices are defined as \( m \) resultant matrices. In order to find a proper parameter \( r \) for the grouping, singular spectrum, \( S = \{ \sqrt{\lambda_1}, \cdots, \sqrt{\lambda_d} \} \) and weighted-correlation (w-correlation), \( \rho^{w} \) are introduced:

\[
\rho^{w}_{ij} = \frac{(Y_{i}^{1}), (Y_{j}^{w})_{w}}{||Y_{i}^{1}||_{w} ||Y_{j}^{w}||_{w}},
\]

(2)

where \( (Y_{i}^{1}), (Y_{j}^{w})_{w} = \sum_{k=1}^{N} w_k Y_{i}^{1}(k) Y_{j}^{w}(k), \ ||Y_{i}^{1}||_{w} = \sqrt{(Y_{i}^{1}), (Y_{i}^{1})_{w}} \) \( (i, j = 1, \cdots, d) \), and \( w \) is defined by \( w_k = \min(k, L, N - k + 1) \). The series \( Y_{i}^{(1)}, Y_{i}^{(w)} \) is available from the elementary matrix by using the diagonal averaging which is defined as follows:

\[
y_{i}^{(1)} = \begin{cases} \sum_{m=1}^{n} y_{i}^{(1)}(m) / n & (1 \leq n < L) \\ \sum_{m=1}^{L} y_{i}^{(1)}(m) / L & (L \leq n < K) \\ \sum_{m=K+1}^{N} y_{i}^{(1)}(m) / (N - K) & (K \leq n \leq N) \end{cases}
\]

(3)

If the reconstructed series \( Y_{i}^{(1)} \) and \( Y_{i}^{(w)} \) are highly correlated, then they can be grouped into a same component. In contrast, if the w-correlation between these two series is quite low or zero, it means that they are well separable into different groups. The series grouped by the result of the w-correlation can be represented as a decomposed form of the initial series, \( Y = Y_{i}^{(1)} + \cdots + Y_{i}^{(w)} \), where the labels of \( I_s \) are equivalent to the \( m \) disjoint subsets in the r.h.s. of eq. (1).

2.3. Forecasting: Linear Recurrent Formulae

The SSA forecasting is in general started with the assumption that the data series is approximately governed by the Linear Recurrent Formulae (LRF) [4–7, 9–13, 15, 16]:

\[
y_{it} = \sum_{k=1}^{d} a_k y_{it+k}, \quad 1 \leq i \leq N - d,
\]

(4)

where \( d \) is the dimension for forecasting and \( a_1, \cdots, a_d \) are defined as coefficients for the LRF. Note that if the original data series \( Y \) satisfies an LRF, there exist at most \( d \) non-zero singular values \( (\sqrt{\lambda_1} \geq \cdots \geq \sqrt{\lambda_d} > 0) \) within a window length \( L \). Therefore, in this forecasting technique it is necessary to prepare at most \( d \) elementary matrices \( X \), in order to reconstruct the series.

The SSA recurrent forecasting algorithm can be explained as follows: Recall the eigenvector \( U \in \mathbb{R}^L \) computed in the SVD step. Let us denote that the vector of the first \( L - 1 \) components of \( U \) as \( U^L \in \mathbb{R}^{L-1} \) and set \( \nu^2 = \pi_1^2 + \cdots + \pi_d^2 < 1 \) as a sum of the square of the last components \( (\pi_i = u_i, \ i = 1, \cdots, r) \) of \( U \). It can be proved that \( y_{it} = a_1 y_{i,t-1} + \cdots + a_{L-1} y_{i,t-L} \) where the coefficients

\[
A = (a_1, \cdots, a_{L-1}) = \left( \frac{1}{1 - \nu^2} \sum_{i=1}^{r} \pi_i U_i^T \right).
\]

(5)

3. Application

The data analyzed and forecasted by the SSA in this study are precipitation and lake-sediments profile in Kenya to investigate the structure of climate in the equatorial zone of East Africa.

3.1. Data: Precipitation and Lake-Sediments

Precipitation shown in the figure 1, had been recorded at the stations of Kenyan three towns, Nakuru, Naivasha, and Narok, with different time length.

![Figure 1: Monthly precipitation in Nakuru, [1904-1991] (left panel), in Naivasha, [1950-1985] (middle panel), and in Narok, [1913-1991] (right panel) from GHCN v2 database.](image-url)

A lake-sediments profile had been taken from the lake Nakuru, which is shown in the figure 2 (top). The bottom one shows the color intensity of the profile. Depth of this profile is ca. 4.6 cm, which corresponds to the time length for ca. 63 years [17].

3.2. Analysis

The first mode component extracted by the SSA is in general a slowly-varying trend. The figure 3 shows that
the larger window length $L$, the slower the trend variation. The component can be shown as a smoothing for different purposes.

![Graph 1](image1.png)

**Figure 2:** Lake-sediments profile from the lake Nakuru (top) and the color intensity of a layer shown on the red line in this profile (bottom). The depth is ca. 4.6 cm (for 63 years). It becomes deeper toward the right. The color intensity is supposed to be linked with rainfall variability [17].

The table lists the periodicity and quasi-periodicity of harmonics components from the precipitation and the lake-sediments.

From the results of precipitation, indeed seasonal cycles are dominant in all the data because such periodicities are shown in the first 3 groups. On the other hand, in other groups of the higher modes, several irregular cycles in an annual sense are shown. Since these results show partly common cycles (10, 15, and 68 months cycles), they may be considered as individual characteristics in this area. Although the rest of the components is in general assumed as noise, it still remains a matter of debate because they are not well interpreted.

The result obtained from the analysis of sediments is that the dominant periodicities are obviously longer than those of precipitation.

### 3.3. Forecast

![Graph 2](image2.png)

**Figure 3:** The trends of precipitation (left panel) and lake-sediments (right panel) in Nakuru for $L = 12$, 36, and 60. The longer $L$, the more slowly-varying is the trend.

![Graph 3](image3.png)

**Table 1:** The groups with oscillation components for $L=60$. The values show monthly periodicities.

<table>
<thead>
<tr>
<th>Group</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nakuru</td>
<td>12</td>
<td>6</td>
<td>4</td>
<td>15</td>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>Naivasha</td>
<td>12</td>
<td>6</td>
<td>13-14</td>
<td>68</td>
<td>10</td>
<td>–</td>
</tr>
<tr>
<td>Narok</td>
<td>12</td>
<td>6</td>
<td>4</td>
<td>68</td>
<td>25</td>
<td>15</td>
</tr>
<tr>
<td>Sediments</td>
<td>46-47</td>
<td>35</td>
<td>27</td>
<td>23</td>
<td>17-19</td>
<td>13-15</td>
</tr>
</tbody>
</table>

The results are depicted in the last 6 points of each data in the figure 4. All of them can approximately be forecasted. Their $L$s are equal to 49 (Nakuru), 98 (Naivasha), 48 (Narok), and 120 (Lake-sediment), respectively. This means that both data complexities consist of the components of similar climate background. On the other hand, the value of $L$ for Naivasha is relatively high. As a prime suspect, it would appear that the amount of data observed in Naivasha is smaller than in the other two towns. $L$ of the sediments data results in an even higher value. This means that these sediments data have already been simplified, i.e., there is a possibility that some information vanished. Therefore, the forecast of the sediments data can be achieved with the small errors.

![Graph 4](image4.png)

**Figure 4:** Forecasts; Nakuru for $\hat{L} = 49$ (left-top panel), Naivasha for $\hat{L} = 98$ (right-top panel), Narok for $\hat{L} = 48$ (left-bottom panel), and Lake-sediment for $\hat{L} = 120$ (right-bottom panel).

### 4. Conclusions

In this study precipitation and lake-sediments profile in Kenya have been analyzed and forecasted by using the

---

- 239 -
The dominant information of these precipitation are some seasonal periodicities, 12 months, 6 months, and 4 months, which mean that cycles such as rainy and dry seasons are regularly repeated in a year. In particular, since there are two rainy seasons in Kenya (in Spring and Autumn), the interval corresponds to a 4 months cycle. As minor properties besides the above cycles, some irregular cycles can be found in the higher modes. These kinds of properties may be considered as cycles not belonging to the seasonal one. From the analysis of the lake-sediments profile, the result is that the dominant cycle is longer than that of precipitation.

In the SSA forecast a complexity of the observed data can be obtained by the parameter $L$. The precipitation in Nakuru and Narok can be forecasted by an almost equivalent optimal parameter $\hat{L}$. The lake-sediments analyzed as a hint about a paleoclimate structure can be forecasted by a relatively large $\hat{L}$, which means that the structure of this data set is simpler than that of the other data sets.

Acknowledgments

The precipitation data from GHCN v2 database (http://www.ncdc.noaa.gov/) used in this study was provided by Norbert Marwan (PIK Potsdam), he and Udo Schwarz (Center for Dynamics of Complex Systems, University of Potsdam) gave me some quite valuable comments and suggestions. We would like to acknowledge the help of them.

References


Directional Spike Propagation by Anisotropic Inhibitory Connections Modulated through STDP in a Recurrent Network

Toshikazu Samura† and Hatsu Hayashi†

†Graduate School of Life Science and Systems Engineering, Kyushu Institute of Technology
2-4 Hibikino, Wakamatsu-ku, Kitakyushu 808-0196, Japan
E-mail: {samura,hayashi}@brain.kyutech.ac.jp

Abstract—It has been supposed that theta oscillations are synchronized across the hippocampus. However, it was reported last year that the theta oscillations propagate along the septotemporal axis in the hippocampal CA1. Although this requires that the hippocampal functions are reconsidered based on the propagation of the theta oscillations, it has not been well understood how the directional propagation is produced in the hippocampus. One of the reasonable mechanisms is that directional spike propagation in the hippocampal CA3 is reflected in the CA1 because the CA3 has rich excitatory recurrent connections. In this paper, we investigated whether the directional propagation emerges in a recurrent network in which the distance of inhibitory connections was anisotropic and weights of excitatory connections from excitatory neurons to inhibitory interneurons were modified through STDP.

1. Introduction

It has been supposed that theta oscillations, which play a crucial role in sequence learning of places, are synchronized across the hippocampus. However, Lubenov and Siapas have recently shown that theta oscillations propagate from the septal side to the temporal side in the hippocampus. Although it has been supposed that theta oscillations propagate along the septotemporal axis in the hippocampal CA1, it has not been well understood how the directional propagation is produced in the hippocampus. One of the reasonable mechanisms is that directional spike propagation in the hippocampal CA3 is reflected in the CA1 because the CA3 has rich excitatory recurrent connections. In this paper, we investigated whether the directional propagation emerges in a recurrent network and found that it was produced certainly when the distance of inhibitory connections was anisotropic and weights of excitatory connections from excitatory neurons to inhibitory interneurons were modified through STDP.

In this paper, we investigated whether the directional spike propagation emerges in a recurrent network in which the distance of recurrent connections is anisotropic. At least excitatory connections between excitatory neurons were modified by a STDP rule in order to cause spike propagation as well as Yoshida and Hayashi model. When the distance of connections was anisotropic and excitatory synapses from excitatory neurons to inhibitory interneurons were not modified by a STDP rule, the present recurrent network caused directional propagation transiently, and then the directional propagation was getting non-directional with the passage of time. However, when the distance of inhibitory connections was anisotropic and excitatory connections from excitatory neurons to inhibitory interneurons were modified by a STDP rule, directional spike propagation was established stably whether the distance of excitatory connections from excitatory neurons is anisotropic or not.

2. Methods

2.1. Recurrent Network

The recurrent network consists of excitatory neurons and inhibitory neurons. Both kinds of neurons are the Izhikevich’s simple model [4]. Equations of the neurons are as follows:

\[ v' = 0.04v^2 + 5v + 140 - u + I, \] (1)

\[ u' = a(bv - u), \] (2)

where \( v \) is the membrane potential and \( u \) is the membrane recovery variable. \( I \) is the sum of weighted inputs from other neurons. \( a \) is the rate of recovery, and \( b \) is the sensitivity of the recovery variable \( u \).

If \( v \geq 30 \), then \( \begin{cases} v \leftarrow c \\ u \leftarrow u + d. \end{cases} \) (3)

If \( v \) is larger than 30, then the neuron fires a spike, and then \( v \) and \( u \) are reset to \( c \) and \( u + d \) respectively. The excitatory neuron was modeled by an intrinsic bursting neuron, so that we set parameters as follows: \( a = 0.02, b = 0.2, c = -55, d = 4 \). We also modeled the inhibitory interneuron as a fast spiking neuron, so that we set parameters as follows: \( a = 0.1, b = 0.2, c = -65, d = 2 \).
10,000 excitatory neurons were placed on 100 × 100 lattice points. 1,250 inhibitory interneurons were placed uniformly among excitatory neurons. Figure 1 shows a part of the structure of the recurrent network. Each inhibitory neuron (▼) was put on an excitatory neuron (▼). Each excitatory neuron was connected to 26 excitatory neurons selected within its projection field (e.g. square (i) in Fig.1) in accordance with the following rule; when a neuron (e.g. A in Fig.1) was connected to another neuron randomly selected in the field (e.g. B in Fig.1), then neuron A was also connected to the neuron on the opposite side of the neuron B (e.g. C in Fig.1). Moreover, the excitatory neuron was connected to 1–7 inhibitory interneurons randomly selected within the projection field. Connections had a delay of 1.4–1.7 ms and the initial excitatory synaptic weights were 40. On the other hand, each inhibitory interneuron was connected to 64 excitatory neurons within its projection field in accordance with the above rule. Inhibitory connections had a delay of 0.9–1.1 ms, inhibitory synaptic weights were −55. Projection fields of neurons near the border were moved inside for keeping the number of neurons within the fields (e.g. square (ii) in Fig.1), and the inhibitory synaptic weights in the field were −82.5.

2.2. Learning Rules

We revised the STDP function proposed by Izhikevich [5]. Synaptic weights between excitatory neurons were updated by the following STDP rule. The magnitude of synaptic change \( \Delta w \) was calculated as follows:

\[
\Delta w = \begin{cases} 
A_e e^{-\Delta t/\tau} & \text{if } \Delta t > 0 \\
A_e e^{\Delta t/\tau} & \text{if } \Delta t \leq 0,
\end{cases}
\]  

(4)

\( \Delta t \) denotes the relative timing between the arrival time of a spike from the pre-neuron and the firing time of the post-neuron. \( A_e \) and \( A_i \) are the maximal potentiation and depression rates respectively. \( \tau \) is the time constant. We set these parameters as follows: \( A_e = 1.0, A_i = -1.2, \tau = 20 \). Each synaptic weight was updated at intervals of 1 sec as follows:

\[
w \leftarrow w + w_{\text{now}} + w_{\text{past}},
\]

(5)

\[
w_{\text{past}} \leftarrow 0.9 \left( w_{\text{now}} + w_{\text{past}} \right),
\]

(6)

where \( w \) is the synaptic weight, and \( w_{\text{now}} \) is the sum of \( \Delta w \) over the last one second. \( w_{\text{past}} \) denotes the sum of past synaptic changes. After the update of synaptic weight by Eq. (5), \( w_{\text{past}} \) was updated by Eq. (6). The current synaptic change \( w_{\text{now}} \), therefore influenced following updates of the synaptic weight. The synaptic weights are limited to the range of 20.0 ≤ \( w \) ≤ 80.0.

3. Results

3.1. Comparison among spike propagations developed in three conditions (control, EA, and IA)

Here, we defined three conditions (control, EA, and IA). The first one is a non-anisotropic condition that is similar to those of Yoshida and Hayashi model as control. Excitatory and inhibitory neurons have connections within a square region in which the distance of connections is the same in vertical and horizontal directions. The second one is an excitatory anisotropic (EA) condition where an excitatory neuron has connections within a rectangle region where the distance of connections is different in vertical and horizontal directions. The third one is an inhibitory anisotropic (IA) condition that an inhibitory neuron has connections within a rectangle region similar to the EA condition. In fact, O-LM inhibitory interneurons in CA3 extend their axons to distant neurons in the septotemporal direction [3]. Table.1 shows the size of the projection field in three conditions.

<table>
<thead>
<tr>
<th>Table 1: Size of the projection fields in three conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control</td>
</tr>
<tr>
<td>EA</td>
</tr>
<tr>
<td>IA</td>
</tr>
</tbody>
</table>

To compare among the developing processes of spike propagation in three conditions, we observed spike propagation in 10 trials in each condition. In each trial, connections between neurons were initialized as shown in Table.1. Here, we applied stimuli (\( I = 200 \)) to 15 neurons within the central 5 × 5 region of the network at intervals of 500 msec. The 15 neurons were randomly chosen every trial. Stimulation was started 5 sec after the beginning of the simulation, then the stimuli were applied to the same neurons to the end. The equations of neurons were updated by the fourth-order Runge-Kutta method every 0.1 msec for 1,000 sec. Synaptic weights between excitatory neurons were modified by the STDP rule, and the synaptic weights from exci-
tatory neurons to inhibitory interneurons were not modified here.

Figure 2 shows typical patterns of spike propagation obtained in each conditions. The patterns are remarkably different within initial 100 sec. In the control condition, spikes propagated radially in 8 trials out of 10 (Fig.2 (a), left). On the other hand, in the EA and IA conditions, the directional spike propagation was caused in a particular direction (Fig.2 (b) and (c), left panels). However, the directional spike propagation changed to the non-directional propagation with the passage of time in all conditions (Fig.2 (a)–(c), right panels). The directional spike propagation was just transient even when the distance of the recurrent connections was anisotropic.

3.2. Directional spike propagation in a network with strong recurrent inhibition

The directional propagation of neuronal activity was transiently produced, and it became non-directional with time, as shown in the previous section. Probably, when integrated EPSPs evoked by firing of nearby excitatory neurons happened to overcome integrated IPSPs evoked by firing of nearby inhibitory interneurons in a particular direction, spike propagated and the excitatory connections were strengthened in the direction through the STDP rule. Although the directional propagation grew transiently, other directional propagations would also grow little by little and the propagation became non-directional finally. Thus, we investigated whether the directional propagation was established stably by stronger inhibition that balanced with strengthened excitation. Here, inhibitory synaptic weights were set as $-70 (-105$ for inhibitory interneurons near the border).

In the control condition, the spike propagation was not produced in all trials. Figure 3 shows typical spike propagation patterns produced in the EA and IA conditions with the stronger inhibition. The directional spike propagation was produced in a particular direction in both conditions (Fig.3 (a), (b)). However, in the EA condition, the directional propagation was produced in only 5 trials out of 10 (Fig.3 (a), left) and propagation was non-directional in other 5 trials (Fig.3 (a), right). In the IA condition, the directional propagation was produced in only 2 trials out of 10 (Fig.3 (b), left) and propagation was not produced in other 8 trials (Fig.3 (b), right).

3.3. Stable directional spike propagation by inhibition modified through a STDP rule

Although the strength of excitatory connections between excitatory neurons was dynamically changed by STDP, strength of inhibitory recurrent connections was fixed in the previous section. It is, therefore, supposed that recurrent inhibition could not properly counteract the strength-
ened recurrent excitation. In this section, excitatory connections from excitatory neurons to inhibitory interneurons were modified through STDP because the inhibitory connections are dynamically modified in the hippocampus [6]. Excitatory and inhibitory synaptic weights were set to 60 and -120 (-180 for inhibitory interneurons near the border) respectively. The weights of excitatory connections were updated within the range of $10.0 \leq w \leq 80.0$. Inhibitory interneurons were connected to 66 excitatory neurons within each projection field. Thirty neurons within the central $15 \times 15$ region were chosen to be stimulated. The other settings were the same as in section 3.1.

Figure 4 shows the typical patterns of spike propagation produced in each condition. In the control and the EA conditions, spike propagation, which was not directional propagation, was produced in 8 trials and 2 trials out of 10 respectively (Fig.4 (a) and (b)). In the other trials, spike propagation was not produced. On the other hand, the directional spike propagation along the vertical axis was produced in the IA condition (Fig.4 (c)). Although, the width of the directional propagation was wider than that of the propagation in Fig.3 (b), the directional propagation shown in Fig.4 (c) was caused in 8 trials out of 10; spike propagation was incomplete in one trial, and no spike propagation was produced in another trial.

![Figure 4](image.png)

**Figure 4:** Typical patterns of spike propagation produced by local stimulation after 1,000 sec from the onset of the simulation. Gray scale of each dot indicates the firing timing of a neuron for 100 ms following each input.

4. Discussions

We investigated whether directional propagation of neuronal activity was produced in a recurrent network, and found that the directional propagation was produced certainly when the distance of inhibitory connections was anisotropic and weights of the excitatory connections from excitatory neurons to inhibitory interneurons were modified through a STDP rule, regardless of the anisotropy of the distance of the recurrent connections from excitatory neurons. It has been reported that inhibitory interneurons, O- LM cells, have axons, which extend along the septotemporal axis in CA3 [3]. This fact is consistent with the IA condition in the present model. It has also been reported that excitatory synaptic connections from excitatory neurons to inhibitory interneurons have synaptic plasticity [6]. Although it has not been elucidated whether the plasticity of the excitatory synaptic connections is dependent on spike-timing, it would be plausible that the synapses are modified through a STDP rule. The present results suggest that anisotropy of the distance of inhibitory connections determines the direction of spike propagation, and the synaptic change in excitatory connections from excitatory neurons to inhibitory neurons ensures that directional propagation is caused in a recurrent network. However, it seems to be paradoxical that spikes propagate in the vertical direction because connections of inhibitory interneurons extend along the vertical axis in the present model and recurrent inhibition is expected to be stronger in the vertical direction than in the horizontal direction. It is an interesting future problem to investigate how recurrent inhibition changes spatiotemporally during the development of the directional propagation and understand the mechanism of the propagation in the direction of connections of inhibitory interneurons.

References


Delay and Release System by utilized Lorenz Attractor

Yoshiyuki Kobori†, Masatoshi Sato† and Mamoru Tanaka†

†Department of Electrical and Electronics Engineering, Sophia University
7-1, Kioi-cho, Chiyoda-ku, Tokyo 102-8554, Japan
Email:nerokiro777@yahoo.co.jp, masatoshi@tlab.ee.sophia.ac.jp, mamoru.tanaka@gmail.com

Abstract—In this paper, we propose a chaotic delay and release system which can change the tone of a sound with time. The chaotic system uses characteristics of chaos waveform and one-pole lowpass filters, and produces a sound with vibrato which is similar to live sound. The output of chaotic system is simulated by using the Csound.

1. Introduction

The chaotic delay and release system is applied to the sound which is converted into electric signals such as electronic musical instruments and electric musical instruments, and the system invents a near sound more naturally. The system is a kind of sound effecter which changes a tone. In general, the sound effecter shows a device that gives the effect of the sound. It is inserted between the device that invents the sound and the speaker, and an original sound is processed[1]. The sounds by stringed instruments such as piano change harmonic sound with time. The changes of the sound volumes are based on the decrease of the vibration with time[2], and this phenomenon is simulated by using a envelope which controls the sound volumes.

The sound is a bright tone at the beginning of waveform which is outputted. The overtone element of the sound has a lot of elements at the beginning of waveform, and attenuates as fast as a high overtone element, and disappears[3,4].

In this proposed system, the filter is utilized in order to express the variation of the sound tone. The sound which is outputted from the oscillator originally has a high overtone element. The attenuation of the overtone element is expressed by lowering the cutoff frequency of the filter which is set high frequency, and being cut from one with a high overtone element gradually. The sound is brought close to the sound of real by lowering the cutoff frequency in chaos. The system proposed method is simulated by using the Csound.

The time change of the overtone element and the filter are shown in Fig. 1. The overtone element attenuates with time, and the inclination of the overtone element grows to attenuate as fast as the high frequency side.

The chaos waveform which is utilized in this proposed system obeys a principle of Lorenz Attractor which has many frequencies. The Lorenz Attractor is one of the nonlinear equations that shows a chaos demeanour. The sound of the natural world has chaos waveform which has characteristics felt warm to us.

The Csound is a programming language that treats the sound, and a text file of two kinds of special forms, orchestra file and a score file, is used as an input[5]. The orchestra file describes the characters of musical instruments, and the score file describes the aging parameter of the score[6]. The Csound executes the instruction group that exists in these files, and outputs the sound. The advantages of the Csound are a lot of the modules and the height of the extendibility by the user[7,8].

Figure 1: Time change and filter of overtone element

2. Algorithm

2.1. Lorenz Attractor

The Lorenz Attractor is defined by

\[ \frac{dx}{dt} = \sigma(y-x), \]
\[ \frac{dy}{dt} = -y - xz + \gamma x, \]
\[ \frac{dz}{dt} = xy - \beta z, \]

where \( \sigma = 10, \gamma = 28, \beta = 8/3. \)

The Lorenz Attractor enters the state of continuous chaos under the condition of Eq. (4). There are three point characteristic of Lorenz Attractor. First, the result is greatly different in the difference of a little initial value (initial value
sensitive). Second, the orbit doesn’t return to the same point (non-periodicity). Third, there is no order, but the orbit settles down in the inside of set.

The output of Lorenz Attractor is assumed to be “kl” which is defined by

\[ k_1 = x + y + z. \]  

(5)

2.2. Algorithm of delay and release system

2.2.1. Input Sound

The frequency spectrum of a piano sound at the beginning of waveform which is outputted is shown in Fig. 2.

The frequency spectrum of a input sound created with the oscillator is shown in Fig. 3. To evaluate a output sound which is utilized the proposed system, the input sound modeled and produced the frequency spectrum of a piano sound.

Several oscillators are used for making the input sound. Each oscillator send a different respectively frequency by the volume of the proper quantity. The method of adding the oscillator is shown in Fig. 4.

2.2.2. Filter

The block diagram of the proposed system is shown in Fig. 5. In "EXPON", a numerical value is changed with the time change, and the output is "kcut". The numerical value is changed from 700hertz to 0hertz in the simulation. "TONE" gives the effect of the low-pass filter to the shape of waves of a input sound from the oscillator. The shape of waves which is outputed from "a1" of "OSCIL" is attenuated from the high frequency by the numeric change that depends on "EXPON" and "Lorenz Attractor". The shape of waves is outputed from "out".

In the simulation, the input sounds are compared by using the filter by "EXPON" or by "EXPON" and "Lorenz Attractor".

3. Simulation results

3.1. Attenuation of volume

The passage of the time of a piano sound and the shape of waves of the volume are shown in Fig. 6. It is shown that the volume is attenuate with the passage of time.

The shape of waves of the time change and the volume which are outputed from the oscillator is shown in Fig. 7. The change of the volume is few.

The shape of waves of sound which is attenuated with the passage of time by "EXPON" is shown in Fig. 8.

The shape of waves of sound which is attenuated with the passage of time by "EXPON" and "Lorenz Attractor" is shown in Fig. 9. To make differences of Fig. 8 and Fig.
9 brought close, the shape of waves of Fig. 8 and Fig. 9 can be brought close to a piano sound by adding not only the filter but also a specific envelope to the input sound.

3.2. Time change of frequency

The volume of the high and low frequency of a piano sound on the time change are shown in Fig. 10. At the beginning of waveform which is outputted, the volume of the high frequency is large, however it attenuates at once. The volume of low frequency attenuates slowly.

The volume of the high and low frequency on the time change when only the volume of the sound is attenuated with the passage of time is shown in Fig. 11. The high frequency and low frequency attenuate similarly respectively. When Fig. 10 and Fig. 11 are compared, it is shown that the time change of each frequency is greatly different.

On the time change, the volume of high and low frequency which is attenuated with passage of time by ”EXPON” is shown in Fig. 12. On the time change, the volume of high and low frequency which is attenuated with the passage of time by ”EXPON” and ”Lorentz Attractor” is shown in Fig. 13.

The ratio corresponding to the passage of the time of each high frequency and each low frequency in Fig. 10 and Fig. 12 and Fig. 13 is shown in Fig. 14. It is shown that the attenuation method using ”Lorentz Attractor” approaches a piano sound more than the attenuation method using only ”EXPON”.

![Figure 6: The shape of waves of piano sound](image1)

![Figure 7: The shape of waves of input sound](image2)

![Figure 8: The shape of waves of sound by ”EXPON”](image3)

![Figure 9: The shape of waves of sound by ”EXPON” and ”Lorentz Attractor”](image4)

![Figure 10: The high and low frequency of a piano sound](image5)

![Figure 11: The high and low frequency of attenuated sound averagely](image6)
4. Conclusion

The proposed method is to bring the attenuation change in the frequency of a digital sound using several oscillators close to the attenuation change in the frequency of a piano sound that is the sample sound. In the ratio of high frequency (540 hertz) and low frequency (205 hertz), the method of moving the filter in chaos has approached the sample sound which is a piano sound more than the method of the average movement which do not use chaos. However, the change in the volume was different from the sample sound in the proposed system.

As future works, the system that executes the proposed method and the method of changing the envelope of the volume at the same time will be researched.

References


Figure 12: The high and low frequency of sound by "EX-PON"

Figure 13: The high and low frequency of sound by "EX-PON" and "Lorentz Attractor"

Figure 14: the ratio corresponding by Fig.10, Fig. 12, and Fig. 13
Numerical Calculation Method of Characteristic Multiplier for the Fixed Point in a Rigid Overhead Wire-Pantograph System

Shota Hirashima†, Shu Karube‡ and Takuji Kousaka†

†Department of Mechanical and Energy Systems Engineering, Oita University, 700 Dannoharu, Oita, Oita, 870–1192 Japan
‡Oita National College of Technology, 1666 Maki, Oita, 870–0152 Japan
Email: shota@bifurcation.jp, karube@oita-ct.ac.jp, takuji@oita-u.ac.jp

Abstract—We propose a numerical calculation method of characteristic multiplier for the fixed point in a rigid overhead wire-pantograph system. First, we show a rigid overhead wire-pantograph system and explain its dynamics. Then, the Poincaré map is defined by using the local section and some objects. In addition, the fixed point is completely calculated based on the derivative of Poincaré map. Finally, above method is applied for a rigid overhead wire-pantograph system.

1. Introduction

The class of system that switched depending on its own state, including jump phenomena when the orbit hits the border is generally called as the impact oscillator. It is well known that there are variety of nonlinear phenomena in the impact oscillator because of its switching complicity. Also, impact oscillator is classified into two types based on the property of borders. The first type of impact oscillator has the fixed border. Many researchers have investigated the nonlinear dynamics in such class of impact oscillator; mechanical system [1, 2], spiking neuron model in biological system [3, 4], impact model of forest fire in ecosystem [5].

The rigid overhead wire-pantograph system with moving border is classified into the other type of impact oscillator. One of a big problem in the rigid overhead wire-pantograph system is the rail corrugation because it generate a noise and contact loss in the running railway. Thus the qualitative analysis of the rigid overhead wire-pantograph system is an important topic in the engineering field in terms of the practical application. In a previous work, the simplified model of the rigid overhead wire-pantograph system has proposed in Ref. [6]. We also have studied above simplified model, and clarified some basic characteristics in terms of the bifurcation theory [7]. On the other hand, the detailed bifurcation analysis is very difficult because the model is divided in high-dimensional system with nonlinear property. For this reason, there is no result of the detailed bifurcation analysis in the rigid overhead wire-pantograph system.

This paper addresses the first step to completely analyze the bifurcation phenomena in a two-dimensional model [7] from the mathematical point of view. More precisely, the numerical calculation method is constructed based on Ref. [8]. Note that Ref. [8] clarified the analyzing method of bifurcation phenomena for the fixed point in two-dimensional impact oscillator with the fixed border. First, we show the model and explain its dynamics. Then, the Poincaré map is defined by using the local section and some objects. In addition, the fixed point is completely calculated based on the derivative of Poincaré map. Finally, above method is applied for a rigid overhead wire-pantograph system.

2. A Rigid Overhead Wire-Pantograph System

The overhead wire model includes the rail corrugation, and the pantograph model is composed of a spring, damper and mass, respectively. Note that the mass in the pantograph model impacts to the stopper that vibrates periodically. Now, we call stopper as the border in the following analysis. In addition, we assume the initial displacement $d$ in the pantograph model because actual pantograph has the upward force. Let us suppose that the initial displacement $d$ is defined by the equation of static equilibrium in our model. Consequently, the model’s equation of motion is shown as follows [6]:

$$\begin{align*}
\frac{dx}{dt} &= v, \\
\frac{dv}{dt} &= -x - 2\zeta v + P_s.
\end{align*}$$

Figure 1: Nonlinear vibration system.
where $\zeta$, $x$ and $v$ denote the damping ratio, the displacement and the velocity in the pantograph model, respectively. On the other hand, the normalized equation of the overhead wire model is given by

$$S(t) = e^\sin\Omega t + 1,$$

where the displacement $S(t)$ depends on the amplitude $e$ and the frequency ratio $\Omega$ in the overhead wire model.

Figure 2 shows an example of the orbit. In the upper section of Fig. 2 denotes the displacements of the mass $(x)$ and border $(S(t))$, respectively. Likewise, the lower section of Fig. 2 corresponds to the velocities of the mass $(v)$ and the border $(dS(t)/dt)$. The orbit is described by Eq. (1) until the parameter $x$ reaches to $S(t)$. After that the tangential force $P_n$ appears between the pantograph and the overhead wire. Now, $P_n$ is expressed as follows:

$$P_n = \begin{cases} 
  e^\{(1 - \Omega^2) \sin \Omega t + 2\zeta \Omega \cos \Omega t\} + 1, & x \leq S(t) \\
  0, & x > S(t).
\end{cases}$$

(3)

The orbit during the periodic interval is classified into two types by the state of $P_n$ at time $x = S(t)$. If $P_n > 0$ is satisfied, the pantograph model keeps the state of contact with the overhead wire model. On the other hand, the impact between the pantograph model and overhead wire model can be observed under the condition of $P_n = 0$; and the velocity immediately switches to the positive vector field. Note that $v_+$ in Eq. 4 denotes the velocity of after the impact.

$$v_+ = -\alpha v_- + (1 + \alpha) \frac{dS(t)}{dt}$$

(4)

Now, $v_-$ and $dS(t)/dt$ are velocities of $x$ and $S(t)$. Also, $\alpha$ is a coefficient of restitution between the pantograph model and the overhead wire model.

3. Numerical Calculation Method of Characteristic Multiplier

3.1. Poincaré map

We calculate the characteristic multiplier in the rigid overhead wire-pantograph system. First of all, we let the behavior of orbit in the system as follows:

$$\begin{cases} 
  \frac{dx}{dt} = f(x, v, \lambda) \\
  \frac{dv}{dt} = g(x, v, \lambda)
\end{cases},$$

where the parameters $t$, $x$, $v$, and $\lambda$ satisfies $t \in \mathbb{R}$, $x, v \in \mathbb{R}^2$, $f, g : \mathbb{R}^2$ and $\lambda$. Now, Eq. (5) is rewritten as follows:

$$\begin{cases} 
  x(t) = \varphi(t; x, v, \lambda), x(0) = x_0 \\
  v(t) = \phi(t; x, v, \lambda), v(0) = v_0
\end{cases},$$

where $x_0$ and $v_0$ means the initial value at time $t = 0$. Next, we define the following local section $\Pi \in \mathbb{R}^2$ by using scalar functions $q : \mathbb{R}^2 \to \mathbb{R}^2$.

$$\Pi = \{x, v \in \mathbb{R}^2 : q(t, x, v) = 0, q : \mathbb{R}^2 \to \mathbb{R}^2\},$$

where $q(t + T, x, v) = q(t, x, v)$.

(7)

Also, we assume that the map $P$ gives the following jumping phenomena in the orbit if $x$ reaches to $\Pi$.

$$P : \mathbb{R}^2 \to \mathbb{R}^2,$$

$$\begin{bmatrix} 
  x_{1a} \\
  v_{1a}
\end{bmatrix} = \begin{bmatrix} 
  \varphi(\tau_0; x_0, v_0) \\
  \phi(\tau_0; x_0, v_0)
\end{bmatrix}$$

(8)

$$\begin{bmatrix} 
  x_{1a+} \\
  v_{1a+}
\end{bmatrix} = \begin{bmatrix} 
  r(\tau_0; x_0, v_0) \\
  s(\tau_0; x_0, v_0)
\end{bmatrix}$$

$\tau_0$ is the time when the orbit reaches to $\Pi$. The orbit $x_1$ at the time $T$ is expressed as follow:

$$\begin{bmatrix} 
  x_1 \\
  v_1
\end{bmatrix} = \begin{bmatrix} 
  \varphi(T - \tau_0; x_{1a+}, v_{1a+}, \lambda) \\
  \phi(T - \tau_0; x_{1a+}, v_{1a+}, \lambda)
\end{bmatrix}.$$

(9)

Next, we define the map.

$$M_0 : \mathbb{R}^2 \to \Pi, (x_0, v_0) \mapsto (x_{1a}, v_{1a})$$

$$M_1 : \Pi \to \mathbb{R}^2, (x_{1a+}, v_{1a+}) \mapsto (x_1, v_1).$$

Consequently, the Poincaré map is given by

$$M : \mathbb{R}^2 \to \mathbb{R}^2$$

$$(x_0, v_0) \mapsto (x_1, v_1) = M_1 \circ P \circ M_0.$$
In the following analysis, we discuss the derivative of the Poincaré map. Eq. (11), is rewritten as follows:

\[
DM(x_0, v_0) = \begin{bmatrix}
\frac{\partial M(x_0)}{\partial x_0} & \frac{\partial M(x_0)}{\partial v_0} \\
\frac{\partial M(v_0)}{\partial x_0} & \frac{\partial M(v_0)}{\partial v_0} 
\end{bmatrix}
\]

\[
= \begin{bmatrix}
\frac{\partial \theta}{\partial t}_{b-r_0} & \frac{\partial \tau_0}{\partial t}_{b-r_0} \\
\frac{\partial \phi}{\partial t}_{b-r_0} & \frac{\partial \tau_0}{\partial t}_{b-r_0}
\end{bmatrix}
\]

Moreover, we should remark that the function \( q = \frac{\partial q}{\partial x_0} \) can be obtained as:

\[
\frac{\partial q}{\partial x_0} = \begin{bmatrix}
\frac{\partial q}{\partial x_0} \\
\frac{\partial q}{\partial v_0}
\end{bmatrix}
\]

\[
= \begin{bmatrix}
\frac{\partial \theta}{\partial t}_{b-r_0} & \frac{\partial \tau_0}{\partial t}_{b-r_0} \\
\frac{\partial \phi}{\partial t}_{b-r_0} & \frac{\partial \tau_0}{\partial t}_{b-r_0}
\end{bmatrix}
\]

A fixed point of the Poincaré map is given by

\[
x_0 - M(x_0) = 0. \tag{18}
\]

The characteristic equation for the fixed point is expressed as

\[
\chi(\mu) = |\mu I_n - DM(x_0)| = 0. \tag{19}
\]

We can obtain the location and stability of fixed point when Eqs. (18) and (19) are solved by arbitrary numerical calculation method.

### 3.2. Application result

Figure 3 shows the example of the orbit and phase plane with various \( \Omega \). In this figure, we can observe that the fixed point bifurcate to the period-2 orbit via the period doubling bifurcation. Now, we pay attention to above period doubling bifurcation, and apply our method for the fixed point. More precisely, we calculate Eqs. (18) and (19). Table 1 shows the analytical result; the fixed point and characteristic multiplier \( \mu_1 \) and \( \mu_2 \). Consequently, we conclude that the period doubling bifurcation is occur at \( \Omega = 4.57641 \). After that the fixed point becomes unstable, and the period-2 orbit is generated. Here, Fig. 4 shows the one-dimensional bifurcation diagram at \( \zeta = 0.049 \). Likewise, the period doubling bifurcation is observed at \( \Omega = 4.57641 \) in this figure. To that end, we can realize that our method proposed in the previous section may applicable to the analysis of stability for the period-n \((n \geq 2)\) orbit in the rigid overhead wire-pantograph system.

### 4. Conclusion

In this paper, we have proposed the numerical calculation method of characteristic multiplier for the fixed point in the rigid overhead wire-pantograph system. First, we explained the model and derived the the Poincaré map. Then, the fixed point was completely calculated based on the derivative of Poincaré map. Finally, we applied our method to the rigid overhead wire-pantograph system. As a result, our method’s validity was confirmed. Also, we consider that our method proposed in this paper may applicable to the analysis of stability for the period-n \((n \geq 2)\) orbit in the rigid overhead wire-pantograph system. Our future work to be studied is to calculate the stability of the period-n \((n \geq 2)\) orbit.
Table 1: Calculation of the fixed point and characteristic multiplier with $\Omega (\zeta = 0.049)$.

<table>
<thead>
<tr>
<th>$\Omega$</th>
<th>$\mu_1$</th>
<th>$\mu_2$</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.45798</td>
<td>-0.02840</td>
<td>-0.73682</td>
<td>Stable</td>
</tr>
<tr>
<td>4.48592</td>
<td>-0.02619</td>
<td>-0.79956</td>
<td>Stable</td>
</tr>
<tr>
<td>4.51550</td>
<td>-0.02422</td>
<td>-0.86545</td>
<td>Stable</td>
</tr>
<tr>
<td>4.56359</td>
<td>-0.02160</td>
<td>-0.98157</td>
<td>Stable</td>
</tr>
<tr>
<td>4.56804</td>
<td>-0.02139</td>
<td>-0.99031</td>
<td>Stable</td>
</tr>
<tr>
<td>4.57201</td>
<td>-0.02120</td>
<td>-1.00000</td>
<td>Period-doubling</td>
</tr>
<tr>
<td>4.57641</td>
<td>-0.02100</td>
<td>-1.00861</td>
<td>Unstable</td>
</tr>
<tr>
<td>4.58033</td>
<td>-0.02082</td>
<td>-1.00861</td>
<td>Unstable</td>
</tr>
</tbody>
</table>

Acknowledgment

The authors would like to thank Prof. Y. Yoshitake of Nagasaki University, Prof. T. Ueta of The University of Tokushima and Assist. Prof. K. Tsumoto of Osaka University for their valuable discussion. This work is partially supported by the Grant-in-Aid for Young Scientists B (KAKENHI) and Research Fund at the Discretion of the President, Oita University.

References


Nonlinear Dynamics in Buck-Boost Converter with Spike Noise

Hiroyuki Asahara and Takuji Kousaka

Department of Mechanical and Energy Systems Engineering,
Oita University, 700 Dannoharu, Oita, Japan 870-1192
Emails: asahara@bifurcation.jp, takuji@oita-u.ac.jp

Abstract—This paper approaches to the nonlinear dynamics in buck-boost converter with spike noise from the mathematical point of view. First, we explain the circuit dynamics. Then, the discrete map is expressed in detail for the rigorous analysis. Finally, we derive some bifurcation diagrams and discuss the dynamic behavior of the waveform in wide parameter space. In particular, we focus on the dynamical effect of spike noise when the circuit dynamics is in discontinuous conduction mode (DCM).

1. Introduction

Analysis of nonlinear dynamics in the piecewise smooth dynamical system is one of a fast developing area of engineering. In the field of electrical engineering as an example, discrete map of the DC/DC power converters is classified into the piecewise smooth map, and have actively studied in the past decade [1, 2, 3, 4].

Most of the previous studies have been conducted under the assumption of the theoretical switching action. Although, unavoidable missed switching actions such as the time delay (switching delay) and high frequency ripple (spike noise) can occur via the switching action in the practical systems; and effect in the circuit’s qualitative property [5]. Therefore we have completely clarified the dynamical effect of missed switching action from both of the mathematical and experimental viewpoints in a simplified switched dynamical system with one dimensional discrete map [6]. Also, we have studied the switched dynamical system containing missed switching with attention to the circuit’s discontinuous conduction mode (DCM) as the next step [7], e.g., clarified map’s property and appearance of some bifurcation phenomena. On the other hand, the dynamical effect of missed switching in a wide parameter space is unclear when the circuit dynamics is in DCM.

In this paper, we focus on the circuit’s DCM dynamics in current mode controlled buck-boost converter as an example, and clarify the dynamical effect of spike noise in a wide parameter space; to developing the circuit theory of such kinds of power converter. First, we explain the circuit dynamics. Then, the consecutive waveform is discretized by the clock interval, and discrete map (return map) is mathematically defined for the rigorous analysis. Finally, we derive some bifurcation diagrams and discuss the dynamical effect of spike noise in a wide parameter space.

2. Circuit Dynamics

Figure 1 shows the circuit model. Now, we let the capacitance voltage be a constant value $E_d$ under the assumption of the capacitance $C$ and resistance $R$ are large enough against the clock interval $T$. Thus, the circuit equation is given by

$$L \frac{di}{dt} = \begin{cases} E_d, & \text{state-1} \\ -E_d, & \text{state-2} \\ 0, & \text{state-3} \end{cases}$$

where state-1, state-2 and state-3 are expressed as follows:

- **state-1**: the switch is conducting, diode is blocking and current satisfies $0 < i < i_{\text{ref}}$,
- **state-2**: the switch is blocking, diode is conducting and current satisfies $0 < i < i_{\text{ref}}$,
- **state-3**: the switch and diode are blocking and the current satisfies $i = 0$.

In the following analysis, the dimensionless variables $\tau = E_d T / (L I_d)$ and $y = i / I_d$ are used. In addition, we let a variable $M$ that satisfies $M = E_d / E_d$.

Figure 2 shows the circuit behavior with ideal switching and spike noise, respectively. The detailed analysis in the circuit with ideal switching has been already studied in Ref. [4]. However, we show the behavior of waveform in the circuit with ideal switching again for the comparison between the circuit with spike noise. Also, the operation time of spike noise does not influence into the circuit dynamics so much, because it is short enough for the clock interval (see Ref. [5]). Thus, we ignore the operation time of spike noise in the following analysis to study the circuit’s fundamental property. The basic switching rule in the circuit falls into three types as follows.

- **Rule-1**: The switch changes from state ON to OFF if the waveform hits the reference value ($y = 1$).
- **Rule-2**: The switch changes from state OFF to ON if the clock pulse is impressed when the switch is set at state OFF.
- **Rule-3**: The switch keeps state OFF until the next clock pulse is impressed if the waveform reaches to zero.
Note that Rule-2 can not be observed if the maximum size of spike noise \( h \) reaches to the reference value, i.e., the switch immediately changes from state OFF to ON and then it keeps state OFF during the clock interval. It also should be mentioned that the circuit dynamics is in DCM if the waveform reaches to zero.

### 3. Return Map

The waveform during the clock interval \( T \) is discretized into ten types by using the critical values \( D_1, D_2, D_3 \) and \( D_4 \) in the circuit with spike noise.

\[
D_1 = 1 - T, \quad D_2 = 1 - T + \frac{1}{M}, \quad D_3 = 1 - h, \quad D_4 = MT \tag{2}
\]

Consequently, we can define the return map in the circuit with spike noise as follows:

\[
y_{k+1} = F(y_k) = \begin{cases} 
  y_k + 1, & (x_k, y_k) \in I_{s1} \\
- M(y_k + T - 1) + 1, & (x_k, y_k) \in I_{s2}, I_{s6}, I_{s7} \\
  0, & (x_k, y_k) \in I_{s3}, I_{s8} \\
  y_k - MT, & (x_k, y_k) \in I_{s4}, I_{s9} \\
  0, & (x_k, y_k) \in I_{s5}, I_{s10}
\end{cases}
\] (3)

where \( I_{s_j} \) \( (j = 1 \sim 10) \) denotes the parameter space of the discretized waveform during the clock interval.

\[
I_{s1} = \{ y_{k-1}, y_k | y_k < D_1 \}, \\
I_{s2} = \{ y_{k-1}, y_k | D_1 \leq y_k < D_3 \text{ or } (D_3 \leq y_k, y_{k-1} \leq D_1) \}, \\
I_{s3} = \{ y_{k-1}, y_k | D_2 \leq y_k, y_{k-1} \leq D_1 \}, \\
I_{s4} = \{ y_{k-1}, y_k | D_3 \leq y_k, D_1 < y_{k-1} \}, \\
I_{s5} = \{ y_{k-1}, y_k | D_1 \leq y_k < D_4, D_1 < y_{k-1} \}, \\
I_{s6} = \{ y_{k-1}, y_k | D_1 \leq y_k < D_3 \text{ or } (D_3 \leq y_k < D_2, y_{k-1} \leq D_1) \}, \\
I_{s7} = \{ y_{k-1}, y_k | D_1 \leq y_k < D_2 \}, \\
I_{s8} = \{ y_{k-1}, y_k | D_2 \leq y_k < D_3 \text{ or } (D_3 \leq y_k, y_{k-1} \leq D_1) \}, \\
I_{s9} = \{ y_{k-1}, y_k | D_1 \leq y_k, D_1 \leq y_k, D_1 < y_{k-1} \}, \\
I_{s10} = \{ y_{k-1}, y_k | D_1 \leq y_k, D_1 < y_{k-1} \}
\] (4)

Let us suppose that we need the information of \( y_k \) and \( y_{k-1} \) to classify the waveform during the clock interval; that are the waveform at \( \tau = kT \) and \( \tau = (k - 1)T \), respectively. In other words, the circuit with spike noise is divided into the two dimensional system, because of a solution \( y_{k+1} \) depends on \( y_k \) and \( y_{k-1} \). However the parameter \( y_{k-1} \) is only used in the classification of the map in the circuit with spike noise. Note that the detailed process to the derivation of a discrete map is shown in Ref. [7]. On the other hand, the consecutive waveform during the clock interval is classified into three types in the circuit with ideal switching by using the border \( D_1 \) and \( D_2 \). Also, we can easily derive the return map in the circuit with ideal switching (see Ref. [4]).

Figure 3 shows the examples of waveform and its corresponding return map on \( y_{k-1} - y_k - y_{k+1} \) plane in the circuit with spike noise. In the figure, (a), (b) and (c) denote the circuit’s possible three dynamics; that the circuit dynamics is in CCM (a), is in CCM and DCM (b) and is in DCM (c), respectively.

### 4. Analytical Results

Figure 4 shows the two dimensional bifurcation diagrams in the circuit with ideal switching and spike noise, respectively. The blue line denotes the bifurcation set of the period doubling bifurcation described as follows:

\[
\int_{k=1}^{m} \frac{dF(y_k)}{dT} + 1 = 0, \tag{5}
\]

where \( m \) represents the period-\( m \) waveform. Likewise, green, red and yellow lines are the border for classification of the circuit dynamics. Note that the dashed or continuous lines are used for the circuit with ideal switching and the circuit with spike noise, respectively.

---

**Figure 1:** Buck-Boost converter.

**Figure 2:** Examples of the waveform.

- 254 -
The parameter space in the circuit with spike noise is classified into five regions based on the property of waveform as follows (see Fig. 4 (b)):

Region-s0 = \{T, M : T < h, MT > h\}

Region-s1 = \{T, M : M < 1, 1 - T + \frac{1}{M} > 0\}

Region-s2 = \{T, M : 1 - T + \frac{1}{M} \leq 0\}

Region-s3 = \{T, M : MT > 1 - h, M > 1, 1 - T + \frac{1}{M} > 0\}

Region-s4 = \{T, M : T \geq h, M > 1, MT \leq 1 - h\} \tag{6}

More precisely, we let Region-s0 be the not practical region, because occurrence of spike noise too much influences into the circuit dynamics in Region-s0. Region-s1 and Region-s2 are the existence region of the fixed point. Note that the circuit dynamics certainly be in CCM in Region-s1; however it is in DCM in Region-s2. Also, if the waveform at \(\tau = kT\) is classified into the parameter space \(I_{s3}, I_{s8}, I_{s5}\) or \(I_{s6}\) (see Eq. (4)), the circuit’s characteristic multiplier equal to \(-1\) and the super stable periodic waveform is generated. Now, Region-s3 denotes the existence region of the super stable periodic waveform. Moreover, circuit dynamics surely be in CCM, and all of the waveform is the chaotic attractor in Region-s4. Let us suppose that Region-i1, Region-i2, Region-i3 and Region-i4 in the circuit with ideal switching have same properties explained in the above sentence (see Fig. 4 (a)). In the following analysis, we focus on the circuit’s DCM dynamics and clarified the dynamical effect of spike noise in a wide parameter space (see Region-s3 and Region-i3 in Fig. 4).

Figure 5 shows the enlarged two parameter bifurcation diagram in the circuit with ideal switching and spike noise (it correspond to the pink area in Fig. 4). We notice that the occurrence of spike noise effects in the blue-colored region in Fig. 5. More precisely, it is clear that the circuit dynamics is in DCM at the earlier timing in the circuit with spike noise compared with the circuit with ideal switching (see the dashed or continuous lines in Fig. 5). In other words, Region-i3 is expressed by the following equation.

Region-i3 = \{T, M : MT > 1, M > 1, 1 - T + \frac{1}{M} > 0\} \tag{7}

Thus we can theoretically understand that the occurrence of spike noise makes the existence region of the super stable periodic waveform large, because the parameter \(h\) is a positive value in Region-s3 (see Eq. (6)). Now, Fig. 3 (b) corresponds to the above case, i.e., a part of the return map containing the new switching rule; that keeps state off during the clock internal. Consequently, occurrence of spike noise makes the new switching rule, and effect in the circuit’s DCM dynamics.

On the other hand, appearance of spike noise does not effect in the yellow-colored region in Fig. 5. Here, the gray line that is the border between the blue and yellow region is expressed as follows:

\[-T + \frac{1}{M} + h = 0,\] \tag{8}

where this is the existence condition of the critical value \(D_3\) (see Eq. (2)). In other words, yellow-colored region satisfies the condition of \(D_3 \geq D_2\), i.e., the critical value \(D_3\) is never been observed in the yellow-colored region. Figure 3 (c) corresponds to this case; the new type of switching
rule is never been observed in the circuit with spike noise. Therefore the occurrence of spike noise has no influence into the circuit’s DCM dynamics in the yellow-colored region.

Consequently, we conclude that the appearance of spike noise makes the new switching rule, i.e., the switch keeps state OFF during the clock interval, and can effect into the circuit’s DCM dynamics. Although, occurrence of spike noise has no influence into the circuit dynamics; if both of the circuit with ideal switching and spike noise is in DCM.

5. Conclusion

In this paper, we focus on the circuit’s DCM dynamics in current mode controlled buck-boost converter, and have clarified the dynamical effect of spike noise in a wide parameter space. First, we explained the circuit dynamics. Then, the return map was mathematically defined for the rigorous analysis. Finally, we derived some bifurcation diagrams and discussed the dynamical effect of spike noise in a wide parameter space. As the results, we concluded that the appearance of spike noise makes the new switching rule; and can effect into the circuit’s DCM dynamics. Although, occurrence of spike noise had no influence into the circuit dynamics; if both of the circuit with ideal switching and spike noise was in DCM. Our future work to be studied is the rigorous analysis of the two or more dimensional systems with missed switching.

References

Decomposition of Symmetric Almost Periodic Oscillation in Three-Phase Circuit

Takashi Hisakado† and Shota Ukai‡

†School of Electrical Engineering, Kyoto University
Kyoto daigakukatsura, Nishikyo, 615-8510, Kyoto, Japan

Abstract—This paper describes a method to decompose symmetric almost periodic oscillations in a three-phase circuit based on symmetries. The decomposition classifies frequency components exclusively and we calculate the bifurcation diagram of almost periodic oscillations using 2-dimensional harmonic balance.

1. Introduction

Symmetries provides crucial clue to understand general mechanism for nonlinear systems. Specifically, the symmetries of a system can be used for finding typical modes. That is, if a system has symmetries, we can define invariant subspaces and find out typical periodic modes [1, 2]. This paper extends the concept to almost periodic oscillations and proposes a method to decompose symmetric almost periodic oscillations in a symmetric three-phase circuit.

The symmetric three-phase circuit in Fig.1 is a fundamental model of power systems. The nonlinearity of the inductors generates many kinds of nonlinear oscillations, e.g., subharmonic oscillations[3], asymmetric oscillations[4], cnoidal waves[5] and ILMs[6]. The generation of those periodic oscillations is clarified by symmetries in [2]. We consider almost periodic oscillations and decompose the oscillations by the symmetries.

First, we show the symmetry of the three-phase circuit and define the symmetric almost periodic oscillations. Next, we propose a method to decompose the oscillations. Further, we calculate bifurcation diagram of the almost periodic oscillations using 2-dimensional harmonic balance.

2. Equation of Three-Phase Circuit

The normalized equation of the three-phase circuit in Fig.1 is

\[
\frac{d}{dr} \begin{pmatrix} \psi_{abc} \\ u_{abc} \end{pmatrix} = \begin{pmatrix} -A_{abc} \psi_{abc} + e_{abc} - R_{abc} i_{abc} \\ A_{abc}^T \end{pmatrix},
\]

(1)

where \( \psi_{abc} \) and \( u_{abc} \) are the flux linkages of the inductors and voltages of the capacitors, respectively. The \( I \) denotes unit matrix, and \((*)^T\) and \( \mathbb{R} \) denote transpose and the set of real numbers, respectively. \( R, r, \omega_\phi \) represent normalized circuit parameters which correspond to Y-connected resistors, \( \Lambda \)-connected resistors, and angular frequency of the voltage sources, respectively. We assume that the characteristics of the flux linkages \( i(\psi) \) are represented by monotone increasing odd function.

In order to describe the symmetry of the circuit easily, we introduce phasor vector \( E_{abc} \) and rewrite (1) by the following autonomous equation:

\[
\frac{d}{dr} \begin{pmatrix} \psi_{abc} \\ u_{abc} \\ E_{abc} \end{pmatrix} = \begin{pmatrix} -A_{abc} u_{abc} + \text{Re}[E_{abc}] - R_{abc} i_{abc} \\ A_{abc}^T \end{pmatrix},
\]

(2)

where \( E_{abc} = (E_a, E_b, E_c)^T \in \mathbb{C}^3 \), the \( \mathbb{C} \) denotes the set of complex numbers, \( j \) is imaginary unit and \( \text{Re}[] \) denotes the real part of \( \cdot \). The argument of the initial value \( E_{abc}(0) \) satisfies

\[
\begin{align*}
\text{arg}[E_b(0)] - \text{arg}[E_a(0)] & = -k_3 I, \quad k_3 \equiv \frac{2\pi}{3} \\
\text{arg}[E_c(0)] - \text{arg}[E_a(0)] & = \text{arg}[E_b(0)]
\end{align*}
\]

(3)

where \( \text{arg}[] \in \mathbb{T} \) denotes the argument, the \( \mathbb{T} \) denotes torus and \( I = (1, 1, 1)^T \).

Further, we rewrite (2) as

\[
\frac{dx_{abc}}{dr} = f_{abc}(x_{abc}),
\]

(4)

where \( x_{abc} = (\phi_{abc}^T, u_{abc}^T, E_{abc}^T)^T \).

![Figure 1: Three-phase circuit and nonlinear characteristics of flux interlinkages i(\psi).](image)
3. Symmetries of Three-Phase Circuit

In order to describe the symmetries of the three-phase circuit, we introduce the following permutations \( \tilde{\gamma} \):

\[
\tilde{\gamma} = \begin{pmatrix} a & b & c \\ a & b & c \\ a & b & c \end{pmatrix}, \quad \gamma_n \in \{a, b, c\} \quad (n \in \{a, b, c\}),
\]

\[\chi_n \neq \chi_m \quad (n \neq m \in \{a, b, c\}).\]

For example, a cyclic permutation is represented by

\[
\tilde{\varepsilon}_3 \equiv \begin{pmatrix} a & b & c \\ b & c & a \end{pmatrix}. \tag{5}
\]

The action \( \tilde{\varepsilon}_3 \) satisfies the following commutativity:

\[
\tilde{\varepsilon}_3 f_{abc}(x_{abc}) = f_{abc}(\tilde{\varepsilon}_3 x_{abc}). \tag{6}
\]

This relation shows that the three-phase circuit has the cyclic symmetry. The action of \( \tilde{\varepsilon}_3 \) can be represented also by the following matrix \( c_3 \in \mathbb{R}^3 \times \mathbb{R}^3 \):

\[
c_3 \equiv \begin{pmatrix} c_3' & 0 & 0 \\ 0 & c_3' & 0 \\ 0 & 0 & c_3' \end{pmatrix}, \quad c_3' \equiv \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}. \tag{7}
\]

The matrix \( c_3' \) has three eigenvalues \( 1, a, a^2 \) where \( a = e^{ik} \) and the eigenvectors are

\[
w_0' \equiv \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, \quad w_+ \equiv \begin{pmatrix} 1 \\ a \\ a^2 \end{pmatrix}, \quad w_- \equiv \begin{pmatrix} 1 \\ a^2 \\ a \end{pmatrix}. \tag{8}
\]

respectively. In the same way, the eigenvectors of \( c_3 \) are

\[
w_0 \equiv (w_0'^T, w_0'^T, w_0'^T)^T, \quad w_+ \equiv (w_+^T, w_+^T, w_+^T)^T \quad \text{and} \quad w_- \equiv (w_-^T, w_-^T, w_-^T)^T.
\]

Next, we consider inversion symmetry based on the odd symmetry of the function \( i(\theta) \):

\[
\tilde{x}_{abc} = i x_{abc} = -x_{abc}. \tag{9}
\]

The action \( i \) satisfies

\[
i f_{abc}(x_{abc}) = f_{abc}(i x_{abc}). \tag{10}
\]

This relation shows that the three-phase circuit has the inversion symmetry. The action \( i \) can be represented also by the following matrix \( i \in \mathbb{R}^9 \times \mathbb{R}^9 \)

\[
i \equiv -I \tag{11}
\]

where \( I \) denotes \( 9 \times 9 \) unit matrix and the eigenvalue of \( i \) equals \(-1\).

From the 2 symmetries, the three-phase circuit has the symmetry with respect to the commutative group \( \tilde{\Gamma} \)

\[
\tilde{\Gamma} \equiv \{ \tilde{\varepsilon}, \tilde{\varepsilon}_3, \tilde{\varepsilon}_3, \tilde{i}, \tilde{i} \tilde{\varepsilon}_3, \tilde{i} \tilde{\varepsilon}_3^2 \}. \tag{12}
\]

Subgroups of the group \( \tilde{\Gamma} \) are listed in Tab.1 and the lattice of the subgroups is shown in Fig.2.

![Figure 2: Lattice of symmetries in three-phase circuit](image)

4. Decomposition of Almost Periodic Oscillation

4.1. Definition of symmetric almost periodic oscillation

Almost periodic oscillation with normalized phase \( \theta \in \mathbb{T}^2 \) on a 2 dimensional torus is defined by \( x(\theta) \) and \( x_{abc}(\theta) \) is represented by

\[
x_{abc}(t) = x(\theta), \quad \theta = \omega, \quad \omega \in \mathbb{R}^2, \tag{13}
\]

where \( \omega \) is angular frequency of torus.

Let us consider a subgroup \( H \subset \tilde{\Gamma} \). If an almost periodic oscillation \( \hat{x}(\theta) \) satisfies

\[
\hat{H} = \{ \hat{\gamma} \in \tilde{\Gamma} | \hat{\gamma}(\hat{x}(\theta)) = \hat{x}(\theta) \} \tag{14}
\]

for all the actions \( \hat{\gamma} \in \hat{H} \), the almost periodic oscillation has spatio-temporal symmetry[1]. This relation shows that the \( \hat{H} \)-action preserves the trajectory of \( \hat{x}(\theta) \) and an action \( \hat{\gamma} \) causes only a shift \( k \in \mathbb{T}^2 \):

\[
\hat{\gamma} \theta + k = \hat{\gamma} \theta - k. \tag{15}
\]

We denote the correspondence between \( \hat{\gamma} \) and \( k \) by a map \( k = \Theta(\hat{\gamma}) \). An example of \( k = \Theta(\hat{\gamma}) \) for the group \( \tilde{\Gamma} \) is shown in Tab.2.

![Figure 2: Lattice of symmetries in three-phase circuit](image)

### Table 1: Subgroup of \( \tilde{\Gamma} \)

<table>
<thead>
<tr>
<th>Order</th>
<th>Subgroup of ( \tilde{\Gamma} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( \tilde{E} \equiv { \tilde{\varepsilon} } )</td>
</tr>
<tr>
<td>2</td>
<td>( \tilde{I} \equiv { \tilde{i} } )</td>
</tr>
<tr>
<td>3</td>
<td>( \tilde{C}_3 \equiv { \tilde{\varepsilon}, \tilde{\varepsilon}_3, \tilde{i} \tilde{\varepsilon}_3 } )</td>
</tr>
<tr>
<td>4</td>
<td>( \tilde{C}_3 \equiv { \tilde{\varepsilon}, \tilde{\varepsilon}_3, \tilde{i} \tilde{\varepsilon}_3, \tilde{i} \tilde{\varepsilon}_3^2 } )</td>
</tr>
</tbody>
</table>

### Table 2: Example of the map \( k = \Theta(\hat{\gamma}) \)

<table>
<thead>
<tr>
<th>( \hat{\gamma} )</th>
<th>( \hat{\varepsilon} )</th>
<th>( \hat{\varepsilon}_3 )</th>
<th>( \hat{\varepsilon}_3^2 )</th>
<th>( \hat{i} )</th>
<th>( \hat{i} \tilde{\varepsilon}_3 )</th>
<th>( \hat{i} \tilde{\varepsilon}_3^2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( k )</td>
<td>( \begin{pmatrix} 0 \ 0 \end{pmatrix} )</td>
<td>( \begin{pmatrix} \frac{2\pi}{3} \ 0 \end{pmatrix} )</td>
<td>( \begin{pmatrix} \frac{2\pi}{3} \ 0 \end{pmatrix} )</td>
<td>( \begin{pmatrix} \pi \ \pi \end{pmatrix} )</td>
<td>( \begin{pmatrix} \frac{2\pi}{3} \ \pi \end{pmatrix} )</td>
<td>( \begin{pmatrix} \pi \ \pi \end{pmatrix} )</td>
</tr>
</tbody>
</table>
4.2. Decomposition of oscillation

We represent \( \hat{x}(\theta) \) by 2-dimensional Fourier series expansion

\[
\hat{x}(\theta) = \sum_{j \in \mathbb{Z}^2} X_j \exp(j^T \theta) + \text{c.c., } X_j \in \mathbb{C}^9, \tag{16}
\]

where c.c. represents complex conjugate and \( \mathbb{Z} \) denotes the set of integers. Then, the 2-dimensional Fourier series expansion of Eq.(15) derives

\[
\hat{y} \in \mathbb{H}, \quad \hat{y} X_j = \exp(j^T k) X_j. \tag{17}
\]

This equation shows that the vector \( X_j \) is in the eigenspace of the action \( \hat{y} \) with respect to the eigenvalue \( \lambda \equiv \exp(j^T k) \).

For example, the cyclic symmetry \( \hat{y} = \hat{c}_3 \) has three eigenvalues \( 1, a, a^2 \) and the \( X_j \) can be classified into three eigenspaces \( w_0, w_+, \text{ and } w_- \). That is, from Tab.2 we can classify the vector \( j \) into three set

\[
k_0 \equiv \{ j \in \mathbb{Z}^2 \mid j_1 \mod 3 = 0 \},
\]

\[
k_+ \equiv \{ j \in \mathbb{Z}^2 \mid j_1 \mod 3 = 1 \}, \tag{18}
\]

\[
k_- \equiv \{ j \in \mathbb{Z}^2 \mid j_1 \mod 3 = -1 \}.
\]

In the case of the inversion symmetry \( \hat{y} = \hat{l} \), the eigenvalues are \(-1\) and the \( j \) belongs to the set \( k_{\text{odd}} \):

\[
k_{\text{odd}} \equiv \{ j \in \mathbb{Z}^2 \mid j_1 + j_2 \mod 2 = 1 \}. \tag{19}
\]

As a result, the flux \( \hat{\psi}_{abc} : \mathbb{T}^2 \mapsto \mathbb{R}^3 \) can be decomposed as

\[
\hat{\psi}_{abc}(\theta) = \sum_{j \in k_0 \cap \mathbb{K}_{\text{odd}}} \Psi_{0,j_1,j_2} w'_0 \exp(\ii(j_1 \theta_1 + j_2 \theta_2))
\]

\[
+ \sum_{j \in k_+ \cap \mathbb{K}_{\text{odd}}} \Psi_{+,j_1,j_2} w'_+ \exp(\ii(j_1 \theta_1 + j_2 \theta_2))
\]

\[
+ \sum_{j \in k_- \cap \mathbb{K}_{\text{odd}}} \Psi_{-,j_1,j_2} w'_- \exp(\ii(j_1 \theta_1 + j_2 \theta_2)) + \text{c.c.} \tag{20}
\]

We call the components in \( w'_0, w'_+, w'_- \) common mode, forward mode and backward mode, respectively. This decomposition (20) indicates that the spectra are also decomposed exclusively in the modes.

4.3. Example of decomposition

In order to confirm the decomposition, we apply the method to an almost periodic oscillation with \( \hat{\Gamma} \) symmetry shown in Fig.3. The figure shows the waveform of fluxes \( \psi_a(t), \psi_b(t), \psi_c(t) \). The frequency components of the fluxes and the decomposed frequency components are shown in Fig.4 and Fig.5. The pattern diagram of Fig.5 is shown in Fig.6, where \( \omega_0 \) and \( \omega_+ \) correspond to \( j = (0, 1)^T \) and \( j = (1, 0)^T \), respectively. The common, forward and backward modes enables the exclusive decomposition of the frequency components.

5. Bifurcation Analysis by 2-Dimensional Harmonic Balance

Almost periodic oscillations can be analyzed by 2-dimensional harmonic balance method [7]. Although the method requires large number of frequency components, the proposed method decomposes the spectra exclusively and reduces the number of unknowns.

Table.3 and Fig.7 show the considered frequency components for the harmonic balance. The bifurcation diagram obtained by the method is shown in Fig.9. The lines \( A_1 \) and \( A_2 \) represent the bifurcation diagram of periodic oscil-
6. Conclusion

We showed that symmetric almost periodic oscillations can be decomposed into the subspaces which correspond to the eigenspaces. Using the decomposition, we classified also the frequency components of the almost periodic oscillation exclusively. Further, applying the decomposition to 2-dimensional harmonic balance method, we obtained the bifurcation diagram of the almost periodic oscillation.

Although the decomposition is shown in the three-phase circuit, the method can be applied to symmetric almost periodic oscillations in general systems.

References

A mathematical-structure-based aVLSI silicon neuron model

Takashi Kohno†‡ and Kazuyuki Aihara†

†Institute of Industrial Science, University of Tokyo
4-7-1 Komaba, Meguro-ku, Tokyo, 153-8505 Japan
‡PRESTO, Japan Science and Technology Agency
4-1-8 Honmachi, Kawaguchi-shi, Saitama, 332-0012 Japan
Email: kohno@sat.t.u-tokyo.ac.jp, aihara@sat.t.u-tokyo.ac.jp

Abstract—The mathematical structures under the conductance-based neuron models have been studied extensively from the perspective of the nonlinear dynamics and bifurcation theory. We proposed to design silicon neuron models by re-constructing the topological structures in the phase portraits and the bifurcation diagrams of the conductance-based neuron models utilizing device-native curves. It not only allows us to design simple circuitry retaining the neuronal dynamics but also provides effective procedures to determine the parameter voltages applied to operate the circuits. An analog Very-Large-Scale Integration (aVLSI) silicon neuron model that mimics the mathematical structures in two groups of bursting neurons was designed based on this idea. The results of the theoretical model and HSpice simulations are reported.

1. Introduction

The silicon neuron is an artificial copy of the neuronal cells made of electronic circuit, which is designed to simulate electrophysiological behavior in real time. Most of the silicon neurons are comprised of low-power consuming analog circuit that solves the models of neuronal cells. Implementing a detailed model of ionic conductances such as Leech heart interneuron model yields a silicon neuron circuit very similar to its biological counterpart [1]. However, such circuits, the conductance-based silicon neurons, reflect the complexity of the neuron models into their circuitry, which affects their stability, power consumption, and tunability of parameters. Particularly, the last point is a critical problem because these neuron models are dependent on large number of parameters and the characteristics of the circuit cannot be free from deviation. Some researchers adopt ultimately simple neuron models, the leaky integrate-and-fire model and its expansion, to resolve these problems [2], though it sacrifices the dynamical behavior of the silicon neurons, because such models describe limited aspects of the neuronal behavior.

In our previous works [3][4][5], we applied the techniques of qualitative modeling to design silicon neuron models that can be implemented by a simple and low-power consuming Metal-Oxide-Semiconductor Field-Effect Transistors (MOSFETs) circuit. These mathematical techniques have been studied over 50 years to elucidate the mechanisms of various neuronal behaviors utilizing the phase plane and the bifurcation analysis [6]. By reproducing such mechanisms utilizing the characteristics curves of MOSFET circuits, we can design device-native silicon neuron models that have the intrinsically same dynamics as the original neuron model. A silicon neuron model designed by this approach is presented in the following sections. It is designed to be implemented by differential pair and current-mode integrator circuitries operated in the subthreshold region of the MOSFET. These ultimately low-power-consuming circuitries are established and commonly utilized in conductance-based silicon neurons [1][7]. Another advantage to the simple circuitry and the preservation of dynamical behavior is that we can determine the parameter voltages adapting the variation of the circuit at fabrication by the mathematical techniques.

2. Design of system equations

The model of our silicon neuron is designed so that it copies the mathematical structures in two groups of bursting neurons, the square-wave and the elliptic bursters [8]. It had been elucidated that both of the bursters comprise a fast subsystem and a slow negative feedback current. The former is a basic excitable system that produces action potentials and the latter operates as an intrinsic stimulus to the former. The key property in these types of bursting is the bistability in the fast subsystem, which is produced by a saddle-loop homoclinic orbit bifurcation in the square-wave bursters and a subcritical Hopf bifurcation in the elliptic bursters. Our model re-constructs the topological structures in their phase portraits and bifurcation diagram.

The system equations of our silicon neuron model are as follows:

\[ \frac{dv}{dt} = -g(v) + f_m(v) - n - q + I_a + I_{stim}, \]  
\[ \frac{dn}{dt} = \frac{f_p(v) - n}{T_n}, \]  
\[ \frac{dq}{dt} = \frac{f_p(v) - q}{T_q}, \]  

where \( n \) is a recovery variable and \( v \) and \( q \) respectively represent the membrane potential and the slow negative feed-
back that operates as an intrinsic stimulus current in parallel with an external one \( I_{\text{stim}} \) and a constant bias current \( I_b \). The first two variables are the component of the fast subsystem. The time constants of the variables \( v, n, \) and \( q \) are represented by \( C_v, T_n, \) and \( T_q \), respectively. The functions \( g(v) \) and \( f_x(v) \) \((x = m, n, p)\) are the characteristic curves of differential-pair-based circuits. They are functions similar to the hyperbolic tangent as follows:

\[
g(v) = S \frac{1 - \exp\left(-\frac{\kappa}{U_T} (v - \theta)/2\right)}{1 + \exp\left(-\frac{\kappa}{U_T} (v - \theta)/2\right)},
\]

\[
f_x(v) = M_x \frac{1}{1 + \exp\left(-\frac{\kappa}{U_T} (v - \delta_x)/2\right)}.
\]

where \( x = m, n, \) and \( p \). In these equations, \( \kappa \) and \( U_T \) are respectively the capacitive coupling ratio and the thermal voltage, \( S \) and \( M_x \) are non-negative constants that control the amplitude of these functions, and \( \theta \) and \( \delta_x \) are constants that displace these functions in the direction of the \( v \)-axis.

2.1. Square-wave burster mode

When the parameters are selected appropriately, a saddle-loop homoclinic orbit bifurcation emerges in the fast subsystem when \( q \) is varied. The \( n-v \) phase plane near this bifurcation is drawn in Fig. 1. There exists a bistability between a stable limit cycle that represents a tonic firing state and a stable equilibrium ((S) in the figure) that represents a silent state. Figure 2 illustrates the bifurcation diagram of the fast subsystem where the bifurcation parameter is \( q \), which draws the \( v-q \) plane of the whole system. When the system state is at the left (right) side of the \( q \)-nullclines, \( dq/dt \) is positive (negative). This negative-feedback nature of \( q \) produces alternation between the tonic firing and the silent states where \( v \) is high and low, respectively. It is a mechanism of the burst firing. A trajectory of the state point is drawn in the figure (the closed curve labeled “orbit”).

The waveforms of the membrane potential \( v \) are shown in Fig. 3, where (b) corresponds to the trajectory drawn in Fig. 2 and (a) shows a chaotic firing patterns observed when \( M_p \) is varied. In the previous work[5] we showed that our model produces chaotic firing patterns very similar to those reported both in a qualitative[9] and a biological models[10] of the square-wave burster. The fact that our model retains appropriately the ability to produce very complex firing patterns in the square-wave bursters supports that our model is their successful silicon-optimized model.

2.2. Elliptic burster mode

In this configuration, the parameters are selected so that the stable state loses stability via a subcritical Hopf bifurcation when \( q \) is decreased. The \( v-q \) plane of our silicon neuron model is drawn in Fig. 4. There is a unique stable equilibrium when \( q \) is sufficiently large and more than half part of the limit cycle is at the left side of the \( q \)-nullcline. Accordingly, \( q \) operates as a slow negative feedback current, which produces burst firing whose trajectory is drawn in the figure (the closed curve labeled “orbit”). The waveform of the membrane potential \( v \) that corresponds to this trajectory is shown in Fig. 5.
3. Circuit Design and Parameter Tuning

The silicon neuron model designed in the previous section is implemented by a circuit whose block diagram is shown in Fig. 6. The circuitries of the blocks \( f_x(v) \) and \( g(v) \) are composed of differential pairs and current mirrors (Fig. 7). Integration of the variables \( n \) and \( q \) are realized by current-mode integrator circuits, whose circuitry is shown in Fig. 8. Their output currents \( I_n \) and \( I_q \) and the voltage of the capacitor \( C_v \) correspond to \( n \), \( q \), and \( v \) in the system equations, respectively.

The three switches and the feedback amplifier (SW1–3 and Voltage clamp amp. in Fig. 6) allow us to make voltage clamp measurement. For the normal operation as a silicon neuron, SW1 and SW2 are turned on and SW3 is off. If SW1 and SW2 are turned off and SW3 is on, the membrane potential \( v \) is fixed to \( V_c \) by injection of \( I_v \). The currents \( I_n \), \( I_m \), and \( I_q \) at the stationary state are measured for each \( V_c \) value, which draw the \( v \)-, \( n \)-, and \( q \)-nullclines, respectively. The \( n \)-\( v \) phase plane of the fast subsystem is drawn by combination of the first two nullclines. Because their topological structures, not exact shape, are responsible to the system’s dynamics, we can determine the externally applied parameter voltages viewing the \( n \)-\( v \) phase plane structures dependent on the deviation of the circuits. Once the fast subsystem has set up, we can determine the parameter voltages for the slow feedback by drawing a part of the \( v \)-\( q \) plane. When SW1 is turned on and SW2 and SW3 are off, the fast subsystem operates independent of \( I_q \). Because \( -I_{stim} \) is equivalent to \( I_q \), the stable structures of the \( v \)-\( q \) plane are drawn by plotting \( v \) for each \( I_{stim} \) value while decreasing and increasing it. We performed HSpice simulation utilizing TSMC CMOS .35\( \mu m \) mixed signal process PDK. Figure 9 shows an example \( v \)-\( q \) plane drawn in...
the simulation, where the two filled regions represent stable limit cycle. We can presume that there exists a stable limit cycle between them, and the Hopf bifurcation point is on the left edge of the right one. According to the procedures described above, we succeeded to find appropriate parameter sets for both of the square-wave and the elliptic burster modes. Total power consumption was estimated to be lower than 20 nW. In Fig. 10(a) and (b), the transient simulation results are shown that closely resemble the theoretical simulation results of the model shown in Fig. 3. We also succeeded to find a parameter set for the elliptic burster mode (Fig. 10(c)) in HSpice simulation.

4. Conclusion

We proposed a mathematical-structure-based approach, a new strategy for designing silicon neuron circuits based on the mathematical techniques. It not only allows us to design simple circuitry retaining the neuronal dynamics but also provides powerful theoretical procedures to determine the parameter voltages to realize an intended dynamics. This is demonstrated by a silicon neuron circuit designed according to this strategy that can be configured into two bursting modes, the square-wave and the elliptic burster modes. Though we could not describe because of space limitations, our circuit can be configured to copy the dynamics in the non-bursting neuron models, either the Class I or II in the Hodgkin’s classification by turning SW2 off. If a slow variable is added that can be implemented by the same circuitry as \( q \), our silicon neuron can produce dynamics of the parabolic burster [8] (Fig. 10(d)), another group of bursting neurons including Aplysia R15 cell.

Acknowledgments

The study was supported by JST PRESTO program and was partially supported by a Grant-in-Aid for Young Scientists (A) 19680015 from the Ministry of Education, Culture, Sports, Science, and Technology, the Japanese Government.

References


Circuit Implementation of an A/D Converter Based on the Negative $\beta$-Map with a Discrete-Time Integrator

Yoshihiko Horio†, Kenya Jin’no‡, Tohru Kohda§ and Kazuyuki Aihara*

†Dept. of Electrical & Electronic Eng., Tokyo Denki University, Tokyo, 101-8457, Japan
‡Dept. of Electrical & Electronic Eng., Tokai University, Saitama, 345-8501, Japan
§Dept. of Computer Science & Communication Eng., Kyushu University, Fukuoka, 819-0395, Japan
* Institute of Industrial Science, Tokyo University, Tokyo, 153-8505, Japan
Email: horio@eee.dendai.ac.jp, jinno@ieee.org, kohda@csce.kyushu-u.ac.jp

Abstract—We propose design methods of an A/D converter based on a negative $\beta$-map, which is robust against noise, and mismatches and deviations in circuit devices, using a discrete-time damped-integrator suitable for IC implementation. In addition, we propose two SC circuit examples for the A/D converter based on the negative $\beta$-map. Moreover, we directly apply the A/D converter to a chaos generator with simple modifications. Because the trajectories of the negative $\beta$-map is confined in an invariant subinterval, the resulting chaos generator is robust against circuit non-idealities and noise. Finally, we confirm the proposed methods and circuits through SPICE simulations with ideal circuit elements.

1. Introduction

A data conversion method with a radix of a positive real number ($\beta$-converter) has been proposed [1, 2]. The $\beta$-converters are Nyquist-rate converters with exponential bit-rate accuracy like PCM-type converters, and also robust against circuit deviations and noise like oversampled $\Sigma\Delta$-type converters. Furthermore, another data conversion method with a radix of a negative real number has been proposed based on a negative $\beta$-map [3, 4]. The size of an invariant subinterval of an ordinary $\beta$-map is fixed, but the position of that is translated according to the quantizer threshold [1, 2]. In contrast, an invariant subinterval of the negative $\beta$-map stays around the center of the domain of the map, however, the size of it varies with the quantizer threshold. In addition, the size of the invariant subinterval most expands when the quantizer threshold is set at either edge of the permissible range, so that a possible dynamic range of the converter becomes large [3, 4]. Therefore, the conversion method based on the negative $\beta$-map improves conversion accuracy, in particular, near the upper and lower bounds of the permissible range of the quantizer threshold value.

A circuit block diagram of the data encoder (A/D converter) based on the negative $\beta$-map has been proposed [3, 4]. However, the block diagram is not suitable for practical circuit implementation, especially for IC implementation. Therefore, in this paper, we propose design methods of the A/D converter based on the negative $\beta$-map with a discrete-time integrator suitable for IC implementation. Moreover, two examples of switched-capacitor (SC) circuits for the A/D converter based on the negative $\beta$-map are constructed with the proposed methods.

In addition, we apply the proposed A/D converter circuits to a chaos generator by setting the bit-length of the A/D converters infinity. The trajectories of the negative $\beta$-map are confined in a finite invariant subinterval because the map is eventually locally onto. Therefore, the proposed chaos generators are robust against mismatches of the circuit elements and noise, that is, they are stable without divergence of the trajectories.

Finally, SPICE simulation results with ideal circuit elements are shown in order to demonstrate feasibility of the proposed methods and circuits.

2. Data Converter Based on the Negative $\beta$-Map

The negative $\beta$-map $R(\cdot)$ is given by eq. (1) [3, 4],

$$R(x) = \begin{cases} s - \beta x, & x \in [0, \gamma \nu), \\ \beta s - x, & x \in [\gamma \nu, s), \end{cases}$$

(1)

where $\nu \in [s(\beta - 1), s]$ is a threshold parameter, $-2 < -\beta < -1$ is a radix of the conversion, $s > 0$ is a scaling parameter, and $\gamma = 1/\beta$. Let us define discrete-time $t_n$ ($n$ is an integer). Then, we can rewrite eq. (1) as a one-dimensional discrete-time dynamical system as

$$x(t_{n+1}) = R(x(t_n)) = \begin{cases} s - \beta x(t_n), & x(t_n) \in [0, \gamma \nu), \\ \beta s - x(t_n), & x(t_n) \in [\gamma \nu, s). \end{cases}$$

(2)

Next, we define a binary variable $b(t_n) \in \{0, 1\}$ as

$$b(t_n) = Q_\theta(x(t_n)) = \begin{cases} 0, & x(t_n) \in [0, \theta), \\ 1, & x(t_n) \in [\theta, s). \end{cases}$$

(3)

where $Q_\theta(\cdot)$ is a quantizer with threshold of $\theta = \gamma \nu$. As a result, eq. (2) can be rewritten as

$$x(t_{n+1}) = b(t_n)s + b(t_n)\beta s - \beta x(t_n) = s \left( b(t_n) + b(t_n)\beta \right) - \beta x(t_n).$$

(4)

We can rewrite eq. (2) in a different way as

$$x(t_{n+1}) = s \left( b(t_n) + b(t_n)\beta \right) - \beta x(t_n) = s (b(t_n) + b(t_n) + (\beta - 1)b(t_n)) - \beta x(t_n).$$

(5)

Alternatively,

$$x(t_{n+1}) = s (1 + (\beta - 1)b(t_n)) - \beta x(t_n).$$

(6)
Let us sample an input signal \( x_{\text{input}} \) at \( t = t_1 \) as
\[
x(t_1) = x_{\text{input}}.
\]
Then, we can obtain a binary sequence \( BS_L(x_{\text{input}}) \) given in eq. (8) by repeating eq. (4) or eq. (6) from \( t = t_1 \) to \( t = t_L \) (\( L \) is a bit-length of the A/D conversion).
\[
BS_L(x_{\text{input}}) = (b_1 b_2 \cdots b_L)_{\beta, \sigma_v}
\]
where \( b_n = b(t_n) \) (\( n = 1, 2, \ldots, L \)), \( b_1 = b(t_1) \) is the MSB, and \( b_L = b(t_L) \) is the LSB.

The permissible range (tolerance \( \sigma_v \)) of \( v \) is given by \( \sigma_v = s(2 - \beta) \) [3, 4]. Therefore, \( \theta \) can fluctuate in a range \( \sigma_{\theta} \) given by
\[
\sigma_{\theta} = \gamma \sigma_v = \gamma s(2 - \beta) = s(2\gamma - 1).
\]
That is, even if the value of \( \theta \) is fluctuated because of environmental changes, non-ideal characteristics of the circuit elements, and noise, the A/D converter based on the negative \( \beta \)-map works correctly if the value of \( \theta \) remains in the range of \( \sigma_{\theta} \).

Moreover, the value of \( \beta \) in the real circuit can be well estimated from the observed bit-sequences using the characteristic equation [1, 2, 3, 4].

3. Implementation of the A/D Converter Based on the Negative \( \beta \)-Map with a Discrete-Time Integrator

A discrete-time integrator is favorable in analog sampled-data ICs. Many switched-capacitor (SC) and switched-current (SI) circuits, for example, which are effective for IC implementation of the discrete-time integrators, have been proposed and matured. Some of those discrete-time integrator circuits are robust against the non-ideal characteristics and mismatches of circuit devices, parasitic devices, and noise. Therefore, we propose two design methods to construct the A/D converter based on the negative \( \beta \)-map through the discrete-time integrator in this section.

3.1. Design Method Based on eq. (4)

The \( Z \)-transformation of eq. (4) results in
\[
X(z) = s \left( \frac{B(z)}{1 + (2\gamma - 1)z^{-1}} - \beta X(z) z^{-1} \right),
\]
where \( X(z) \), \( B(z) \), and \( \overline{B(z)} \) are \( Z \)-domain variables for \( x(t_n) \), \( b(t_n) \), and \( \overline{b(t_n)} \), respectively. Moreover, from eq. (3),
\[
B(z) = Q_{\bar{d}}(X(z)), \quad \text{and} \quad \overline{B(z)} = Q_{\bar{d}}(X(z)).
\]
Therefore,
\[
X(z) = s \left( Q_{\bar{d}}(X(z)) + \beta Q_{\overline{d}}(X(z)) \right) z^{-1} - \beta X(z) z^{-1}.
\]
As a consequence,
\[
X(z) = s \cdot \frac{z^{-1}}{1 + \beta z^{-1}} \cdot \left( \beta Q_{\bar{d}}(X(z)) + Q_{\overline{d}}(X(z)) \right).
\]
This shows that a discrete-time integrator with a damping-factor of \( \beta \) can construct the A/D converter circuit based on the negative \( \beta \)-map.

3.2. Design Method Based on eq. (5)

The \( Z \)-transformation of eq. (5) gives
\[
X(z) = s \left( \frac{B(z)}{1 + (2\gamma - 1)z^{-1}} - \beta X(z) z^{-1} \right)\beta X(z) z^{-1}
\]
\[
= s \left( 1 + \beta z^{-1} - \beta X(z) z^{-1} \right)
\]
\[
= s \left( 1 + \beta Q(X(z)) \right) z^{-1} - \beta X(z) z^{-1}.
\]
As a result,
\[
X(z) = s \cdot \frac{z^{-1}}{1 + \beta z^{-1}} \cdot \left( (\beta - 1)Q(X(z)) + 1 \right).
\]
This suggests another method to construct the A/D converter circuit based on the negative \( \beta \)-map utilizing a discrete-time integrator with a damping-factor of \( \beta \).

4. SC Circuit Implementation of the A/D Converter Based on the Negative \( \beta \)-Map

We use a SC damped-integrator circuit shown in Fig. 1, as an illustration, to implement the A/D converter based on the negative \( \beta \)-map given by eqs. (13) and (15). The \( Z \)-domain transfer function of the circuit in Fig. 1 is
\[
V_o(z) = \frac{z^{-1}}{1 + (2\gamma - 1)z^{-1}} \left( \frac{C_{\bar{f}}/C_i}{C_{\bar{f}}/C_i} \right) V_{\overline{d}}(z) + \left( \frac{C_{\bar{f}}/C_i}{C_{\bar{f}}/C_i} \right) V_{\bar{d}}(z),
\]
where \( V_o(z) \), \( V_{\overline{d}}(z) \), and \( V_{\bar{d}}(z) \) are \( Z \)-domain variables for \( v_i(t_n) \), \( v_{\overline{d}}(t_n) \), and \( v_{\bar{d}}(t_n) \), respectively.

First, by comparing eq. (13) and eq. (16), we find that we can implement eq. (13) with the circuit in Fig. 1 by setting \( V_o(z) = X(z), \quad V_{\overline{d}}(z) = Q_{\overline{d}}(X(z)), \quad V_{\bar{d}}(z) = Q_{\bar{d}}(X(z)), \quad C_{\overline{d}}/C_i = \delta \beta, \quad C_{\bar{d}}/C_i = \delta, \quad C_{\bar{f}}/C_i = s, \) and \( C_{\overline{f}}/C_i = \beta + 1 \).

This leads a SC A/D converter circuit based on the negative \( \beta \)-map shown in Fig. 2 which includes an additional SC circuit to sample the input signal \( x_{\text{input}} \), and a quantizer circuit \( Q_{\bar{d}}(\cdot) \) in a feedback loop. The clock waveforms which drive the circuit in Fig. 2 are shown in Fig. 3. In the figure, \( L \) is a bit-length for conversion. Moreover, the \( \Phi \) samples \( x_{\text{input}} \); at the same time, it resets the charge on \( C_i \). Furthermore, \( \Phi P = \Phi A \cdot \Phi C \). In order to sample \( x_{\text{input}} \) according to eq. (7), we set the value of \( C_s \) in Fig. 2 as \( C_s/C_i = 1 \).

Figure 1: A switched-capacitor damped-integrator.
Figure 2: A SC A/D converter circuit based on the negative β-map derived from eq. (13). \( \phi P = \phi A \cdot \phi C \). The circuit inside the chain-lines is unnecessary for the chaos generator without setting the initial condition. In this case, \( \phi P = \phi A \).

Figure 3: Clock waveforms for the circuit in Fig. 2 and Fig. 4. The output bit-sequence \( b_n \) is sampled at the falling edge of \( \phi B \), i.e., at \( n+1/2 \).

Second, we propose another SC circuit implementation of the A/D converter based on the negative β-map according to eq. (15). Comparison between eq. (15) and eq. (16) shows that we can implement eq. (15) with the SC integrator shown in Fig. 1 by setting \( V_0(z) = X(z) \), \( V_0(z) = Q_0(X(z)) \), \( V_1(z) = 1 \), \( C_0/C_1 = s(\beta - 1) \), \( C_1/C_1 = s \), and \( C_1/C_1 = \beta + 1 \).

The final circuit including a SC circuitry \( C_i \) to sample \( x_{input} \), and a quantizer \( Q_0() \) is shown in Fig. 4. In the figure, \( C_i/C_i = 1 \), and \( V_U = 1 \) V. The clock waveforms for Fig. 4 are the same as those in Fig. 3. Moreover, \( \phi P = \phi A \cdot \phi C \).

4.1. Application to a Chaos Generator

Chaos generator circuits with a uniform invariant measure distribution have been proposed based on, e.g., the Bernoulli-map and the tent-map. However, most of those circuits are unstable when their trajectories approach to both ends of the domain of the maps. In contrast, the chaos generator based on the negative β-map proposed in this subsection is stable because the negative β-map is eventually locally onto, so that the trajectories of the map remains in a finite invariant subinterval smaller than the domain of the map.

The A/D converter circuits in Figs. 2 and 4 can be directly used as chaos generators by setting their bit-length infinity \( (L = \infty) \) by making \( \phi C \) a mono-pulse. Moreover, if no setting for the initial condition is needed, the SC circuits for input sampling enclosed by the chain-lines in Figs. 2 and 4, and the clock \( \phi C \) can be omitted to make the circuits compact.

5. SPICE Simulations

In order to confirm the feasibility of the proposed circuits, we simulate the circuits with ideal circuit elements. Although both circuits in Figs. 2 and 4 were simulated and confirmed, the results from the circuit in Fig. 4 are shown in this paper for want of space. Moreover, we assume that the value of \( \beta \) is known, because the object of simulations is to confirm the circuit operations under ideal conditions. In the following simulations, we use \( \beta = 5/3 \), and \( s = 3 \).

We simply compare the bit-sequences \( BS_{L}(x_{input}) \) from the circuit simulations and those obtained from theoretical calculations in the first simulation. Table 1 shows the comparison for different values of \( x_{input} \) where \( L = 8 \). From the table, we can confirm that the proposed A/D converter circuit properly converts the input signals to corresponding bit-sequences.

Next, we evaluate the conversion errors obtained from the SPICE simulations. With the bit-length of \( L \), the bound for the conversion errors is given by \( \epsilon_L(x) = |x_{input} - \hat{x}_L| \leq s^{L}/2 \), where \( \hat{x}_L \) is a decoded value from \( BS_{L}(x_{input}) \) [3, 4]. From this equation, we select \( L = 13 \) for \( \epsilon_L(x) \leq 2^{-9} \).

The conversion errors for different input values \( x_{input} \) are shown in Fig. 5. In the figure, the quantizer threshold \( \theta \) is a parameter. On the other hand, the conversion errors when we swept \( \theta \) are shown in Fig. 6 for different values of \( x_{input} \). From these figures, we can conclude that the proposed circuit properly operates with the specified conversion accuracy even if the quantizer threshold changes. That is, we
Table 1: Comparison between the output sequences obtained from the SPICE simulations of the circuit in Fig. 4 and theoretical calculations for \( L = 8 \).

<table>
<thead>
<tr>
<th>( x_{\text{input}} )</th>
<th>( \theta )</th>
<th>Output Bit-Sequences ( BS(L(x_{\text{input}})) )</th>
<th>SPICE</th>
<th>Theory</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2</td>
<td>1.21</td>
<td>01011101 01011101</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1.5</td>
<td>01011100 01011100</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1.79</td>
<td>01010010 01010010</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0.5</td>
<td>1.21</td>
<td>01101111 01101111</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1.5</td>
<td>01001111 01001111</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1.79</td>
<td>01001111 01001111</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0.8</td>
<td>1.21</td>
<td>01110110 01110110</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1.5</td>
<td>01010110 01010110</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1.79</td>
<td>00010100 00010100</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

have confirmed robustness of the proposed circuit against the fluctuation of \( \theta \), which is one of the advantages of the A/D converter based on the negative \( \beta \)-map.

Finally, we set \( L = \infty \) in the A/D converter circuit of Fig. 4 to realize a chaos generator. Figure 7 shows an example of the chaos attractor from the circuit with \( \theta = 1.5 \). The invariant subinterval of the map in this case is \( [LB = x - \nu = 0.5, UB = \beta s - \nu = 2.5] \) [3, 4] while the domain of the map is \([0, 3]\). As shown in the figure, the chaotic trajectory is confined in the invariant subinterval, which guarantees stable behavior of the circuit.

6. Conclusions

We have proposed the design methods for constructing the A/D converter based on the negative \( \beta \)-map using the discrete-time damped-integrator. The proposed methods are suitable to implement the A/D converter in ICs because the discrete-time integrator is a preferred circuit element in ICs. In addition, we have proposed, as an illustration, two SC circuit examples for the A/D converter based on the negative \( \beta \)-map. Moreover, we have applied the proposed A/D converter to the chaos generator by simply making the conversion bit-length \( L = \infty \). Because the trajectories of the negative \( \beta \)-map is confined in the invariant subinterval, the resulting chaos generator is robust against circuit non-idealities and noise. Finally, we have confirmed the proposed methods and circuits through SPICE simulations with ideal circuit elements.

This research was supported in part by JSPS through its Funding Program for World-Leading Innovative R&D on Science and Technology (FIRST Program). This work was also supported in part by Kakenhi (20300085).

References


Abstract—Digital-signal waveforms propagating in the PCB traces are being distorted seriously causing system failures as the clock-frequency increases. And it is getting difficult to improve the distortions with conventional impedance-matching technique as the clock frequency increases to GHz, which is called “Signal-Integrity (SI) problem in the GHz era”. In order to overcome this problem, we have already proposed a novel PCB-trace structure called “Segmental Transmission Line (STL)”, which is designed using the genetic algorithms. In this paper, we develop a 250MHz-scale-up prototype of the STL and show effectiveness of the STL by measuring the signals propagating in the prototype.

1. Introduction

Digital-signal waveforms in PCBs (Printed Circuit Boards) have been distorted gradually since the clock-frequency got into the 100MHz-era. And in the GHz-era, the waveform distortion is one of the most serious problems in the PCB design for the VLSI packaging.

The waveform-distortion problem, or the so-called “the Signal-Integrity problem,” comes from the fact that the PCB traces (wires) behave as transmission lines as their lengths are approaching the wavelengths of the digital signals propagating on them (e.g., the base-wavelength of 1GHz digital-signal in the PCB is about 15cm). The transmission line generates reflection waves, or noises at characteristic-impedance \( Z \) mismatched points, and the noises degrade the waveform terribly.

In the STL, a transmission line is divided into multiple (\( N \)) segments of individual characteristic impedance \( Z_i \) as shown in Fig. 2. And each \( Z_i \) is adjusted to achieve an ideal digital waveform at target points such as input-points to the LSIs on the line by superposing reflection waves generated at the interfaces between adjacent segments \( Z_i \) and \( Z_{i+1} \). The adjustment of all \( Z_i \)s however, results in a combinatorial-explosion-problem because the search-space expands to \( n^m \), where \( n \) is the impedance range and \( m \) is the number of segments and it comes to 10010-order usually. We have proposed to apply the genetic algorithms (GAs) to solve this problem and have already shown its effectiveness [1][2][3].

Figure 3 shows a bird-eye view and a cross-sectional view of the STL in the PCB. Characteristic impedance \( Z \) is a function of trace width \( W \), trace thickness \( T \), and insulator thickness \( D \). In the STL, \( Z \) is thus controlled easily by adjusting \( W \).

The STL structure can be easily and well mapped onto the chromosome of one-dimensional array of genes as shown in the upper figure in Fig. 4. This chromosome is called “Simple Chromosome”. In the STL, each...
segment-length $L_i$ can be also used as a parameter, which expands the search-space more widely to include better solutions as shown in the lower figure in Fig. 3. The chromosome composed of both characteristic impedance $Z_i$ and segment-length $L_i$ is called “Hybrid Chromosome”. In this variable-length-segment approach, however, we have to be careful not to produce fatal genes under the constraint that the total length is fixed.

**3. Genetic Algorithm for STL Design**

We use the Minimal Generation Gap (MGG) algorithm, which is one of well-known genetic algorithms and was proposed to prevent the population from being occupied with the same kind of chromosomes and losing its diversity. In the MGG, a family is made from randomly selected parents in the $N$-th generation and their children generated by crossover operation as shown in Fig. 5. And one individual is selected from the family based on the elite or roulette selection and it is exchanged with one individual in the $N$-th generation to produce the $N+1$ generation. Since the waveform is very sensitive to the characteristic-impedance and segment-length in the STL, diversity of the population in the STL decreases frequently in its evolution. We thus use the MGG to avoid diversity-degradation in the evolutionary STL design.

We use a fitness shown in Fig. 6. The difference area $\text{Diff}$ between the ideal waveform $I(t)$ and the current waveform $R(t)$ is used for the fitness. The fitness is defined as the reciprocal of this difference area $\text{Diff}$. 

![Fig. 2 Segmental Transmission Line (STL).](image)

![Fig. 3 Structure of STL in PCB.](image)

![Fig. 4 Mapping of STL onto Chromosome](image)

![Fig. 5 MGG for STL Design.](image)

![Fig. 6 Fitness function in STL Design.](image)
3. 250MHz STL Scale-up Prototype

3.1. Model and Design

We have developed a 250MHz scale-up prototype targeting a 1GHz memory-bus clock-distribution system shown in Fig. 6. One memory module (DIMM: Dual in-line Memory Module) is connected to a PCB trace, or transmission line. A clock-driver-LSI connected to the one end of the transmission line outputs a clock signal of 1GHz. The clock-signal propagates in the line of 15cm long and is distributed into the DIMM at two inputs connected to the line. Each input is equivalent to a load capacitance of 4pF, which causes the characteristic-impedance mismatch resulting the waveform distortion.

The 250MHz STL scale-up prototype is designed based on the 1GH DIMM: The transmission line length $L$ and the load capacitance $C_L$ are designed to 60cm and 15pF, respectively from the scale-up ratio of 4, i.e., 250MHz to 1GHz. And the transmission line of 60cm is divided into 15 segments (T1 to T15), totally. $P_O$ in the figure is the first input from the clock-driver-LSI to the DIMM and it is the waveform-observation-point in this paper.

Table 1 shows the design result of characteristic impedance $Z$ and length $L$ of each segment (T1 to T15). We used a set of characteristic impedances from 30Ω to 120Ω at 5Ω intervals in the MGG programs. As shown in the table, low-characteristic-impedance segments of 30 to 65Ω and high-characteristic-impedance segments of 85 to 115 are assigned alternately. This means many reflection waves are generated in each boundary between adjacent segments, and they are used to improve the distorted wave.

<table>
<thead>
<tr>
<th>Seg.</th>
<th>T1</th>
<th>T2</th>
<th>T3</th>
<th>T4</th>
<th>T5</th>
<th>T6</th>
<th>T7</th>
<th>T8</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Z$(Ω)</td>
<td>50</td>
<td>100</td>
<td>65</td>
<td>35</td>
<td>85</td>
<td>30</td>
<td>115</td>
<td>35</td>
</tr>
<tr>
<td>$L$(mm)</td>
<td>36.5</td>
<td>38.5</td>
<td>36.5</td>
<td>38.5</td>
<td>43.5</td>
<td>49.5</td>
<td>43</td>
<td>41.5</td>
</tr>
</tbody>
</table>

3.2. Prototyping and Its Evaluation

We have fabricated a 250MHz STL scale-up prototype based on the design result shown in Tab. 1. The lower photograph in Fig. 7 is the STL prototype while the upper one is the conventional-transmission-line of $Z = 50$Ω uniformly. In the figure of STL prototype, a part of the STL is magnified and shown also with values of the characteristic-impedance and the segment-length.

The STL and the conventional-transmission-line both run in each board of 31cm long in U-turn way, where the line-space is designed widely enough to avoid cross-talk noises. Two chip-capacitors ($C_L$) of 15pF each are connected to the line, which represent the input-capacitances of the memory module (DIMM). In the waveform observation, we used a digital-storage oscilloscope of 2GHz bandwidth and 10GS/s sampling-rate (WR204Xi, LeCroy Ltd.).

The upper waveform in Fig. 8 was observed at $P_O$ in the conventional-transmission-line. The waveform is terribly and alarmingly distorted in its shape and amplitude: each pulse loses its sharpness in its rise and fall curves and have small jagged reflections on it. In addition, its amplitude is reduced to 1.06V from 1.8V of original input clock signal. As a result, the clock-signal in the conventional transmission line shown in Fig. 8 cannot be used in the DIMM clock-signals.

The lower waveform in Fig. 8 was observed at $P_O$ in the STL. The distorted waveform in the conventional transmission line is well improved in the STL successfully: the reduced amplitude is restored to 1.6V, and jagged reflection noises disappear. The waveform is almost like a trapezoidal wave in its shape, which is close to the original clock-signal. The signal in the STL is thus enough for the clock-signal in its shape.
3.3. Comparison with Simulations (Waveforms Designed Using GA)

Figures 9 and 10 show simulation results (upper) and observed waveforms (lower) in the conventional-transmission-line and STL, respectively. In the simulation results, the input-signal, which is sampled in the digital-sampling-oscilloscope and is used as the signal-source in the circuit simulator NGSPICE, is also shown synchronized with the distorted and improved waves, respectively.

In Fig. 9, the amplitude of the distorted signal in simulation is 1.23V, which is larger than the observed one in the prototype by 0.17V, but it is still smaller than the amplitude of the signal-source (1.8V). Even though the small jagged noises do not appear in the simulation, the waveforms in simulation and in the prototype almost resemble each other.

In Fig. 10, the improved signal in simulation matches the signal-source almost completely except the slight difference in its rising points. And the improved waveforms in the simulation and from the prototype observation well resemble each other. This means that the waveform well evolved taking after the source-signal.

4. Conclusions

A 250MHz scale-up STL (Segmental Transmission Line) to 1GHz DIMM (Dual In-line Memory Module) was designed using MGG (Minimum Generation Gap), which was one of well known genetic algorithms, and its prototype was fabricated. The observation result showed that the distorted waveform in the conventional transmission line was restored almost completely in the STL prototype. The observed waveform in the prototype was also well matched the design result, or the simulation result. Effectiveness of the STL to restore the distorted waveform and to ensure the SI (signal integrity) was thus well demonstrated experimentally using the prototype.

Acknowledgments

This work is partly supported by the 2010 cooperative research with The Semiconductor Technology Academic Research Center (STARC) in Japan.

References


Realization of Three-dimensional DT-CNN on FPGA

Nguyen Tien Dat, Nguyen Tien Dzung, Thang Manh Hoang

Faculty of Electronics and Telecommunications
Hanoi University of Science and Technology
1 Dai Co Viet, Hanoi, Vietnam
Email: thang@ieee.org

Abstract—In this paper, realization for the architecture of three-dimensional cellular neural network is introduced for the first time. The complete parallelism is to maximize the performance of CNN. Implementation for 3D-CNN with the size of 4x4x3 on FPGA will be described. Simulation result shows the behavior of the network.

1. Introduction

Cellular neural network (CNN) was introduced by Chua and Yang in 1988 [1]. It is considered as a special case of artificial neural networks. Due to their parallel processing mechanism, the CNN has been successfully used for application requiring high speed processing such as image processing [2], pattern recognition [3], solving PDEs [4], etc. The structure of CNN is formed by local connection, and this feature allows ones to implement on VLSI [5, 6].

There are two approaches to realize CNN on digital platforms. The first is to develop a complete single layer network [7]. In order to emulate 3D-CNN, this layer must be shifted along the input array of its neighbor layers, so it requires architecture that allows to multiplexes inputs and stores current and next output values in memory. This architecture obtains the efficient resource usage and ability in emulating large size of 3D-CNN, but not gets an optimum computing performance. The other way is to direct implementation of entire 3D-CNN on chip. Due to massively parallelism of CNN, it can result the best performance of CNN. The main disadvantage of this approach is the usage of large resources. In this paper, the last approach is chosen for implementation.

In recent years, implementation of CNN has been carried out in various platforms i.e. computer simulation program, ASIC, emulated – digital in FPGA and DSP [8]. The analog ASIC implementation of CNN is the most powerful processor, but its disadvantages are long development time and high cost. Computer program and emulated CNN on reconfigurable chip, FPGA are suitable for verifying a model. In this paper, we present the design of complete 3D-Cellular Neural Network of the size of 4x4x3 with the template of 3x3x3 on FPGA.

2. Cellular Neural Networks

In theoretical, the structure of CNN can be of n-dimension (n=1, 2, 3…). Following, fundamental of 2D-CNN and 3D-CNN is described.

2.1. 2D-CNN

2D-CNN is the dynamical system of identical cells, a cell connects locally to neighbor cells to form a two dimensional array as illustrated in Figure 1.

![Figure 1. A two dimensional 4x4 CNN with neighboring distance r= 1.](image)

Cell C(i,j) contains linear and nonlinear circuit elements. Specifically, it may include independent sources, a linear capacitor, linear resistors, linear and nonlinear controlled sources. A cell C(i,j) couples with neighbor cells via the controlling input voltage and it receives feedbacks from the output voltage of neighbor cells C(k,l). The templates A(i,j;k,l) and B(i,j;k,l) are the weight parameters of links between cells C(i,j) and neighbor ones, C(k,l). The behavior of CNN depends on these template values. The state of cell C(i,j) is given by the following equation:

\[
\frac{dx_{ij}}{dt} = -\frac{1}{R}x_{ij} + \sum_{C(k,l) \in N_r(i,j)} A(i,j;k,l)y_{kl}
\]

\[
+ \sum_{C(k,l) \in N_r(i,j)} B(i,j;k,l)u_{ij} + z_{ij}
\]

\[1 \leq i \leq M, 1 \leq j \leq N\] (1)

The output of cell C(i,j) is as

\[
y_{ij} = \frac{1}{2} (|x_{ij} + 1| - |x_{ij} - 1|)
\]

\[1 \leq i \leq M, 1 \leq j \leq N\] (2)

2.2. 3D-CNN

In the 3D-CNN, the cells are similar to those in the 2D-CNN. The 3D-CNN is formed as collection multiple 2D-CNN layers, and connections between cells is not only in the existing layer of cell, but also to cells in the two neighbor layers (under and upper ones) as depicted in Figure 2. We consider the neighbor...
as a sphere, so if r = 1 the cell C(i,j,k) has 26 neighbor cells. Hence, the templates have three indexes as A(i,j,k); B(i,j,k).

**Figure 2. 3D-CNN with r = 1**

The state and output equations of 3D-CNN is derived as follow:

\[
\begin{align*}
C \frac{dx_{ij,k}}{dt} &= \frac{1}{R} x_{ij,k} + \sum_{C(k,l,m) \in N(i,j)} A(i, j, k; l, m) y_{lmn} \\
&+ \sum_{C(l,m,n) \in N(i,j,k)} B(i, j, k; l, m, n) u_{lmn} + z_{jk} \\
\end{align*}
\]

**Figure 3. Discrete time model of cell**

The discrete time CNN model in Eq. (5) is illustrated in Figure 3. The input data, the initial, output data, the bias and templates are in the format of 16 bit fixed-point signed value, (8.8), and the output state of cell is represented in 18 bit fixed-point signed value, (10.8). The architecture of single cell has two convolution blocks and non-linear function block which realize Eq. (5) and Eq. (6), respectively. These convolution units are composed simply of arithmetic adders and multipliers. The structure of convolution blocks is shown on Figure 4. The data from 27 neighbor cells is multiplied by the corresponding template coefficients, and then is truncated to 16 bit. After then, the result is added together by 5-pipelined-stages tree adders.

**Figure 4. Architecture of convolution block**

The 3D-CNN with the size of 4x4x3 is considered. The discrete time CNN model in Eq. (5) is illustrated in Figure 3. The input data, the initial, output data, the bias and templates are in the format of 16 bit fixed-point signed value, (8.8), and the output state of cell is represented in 18 bit fixed-point signed value, (10.8). The architecture of single cell has two convolution blocks and non-linear function block which realize Eq. (5) and Eq. (6), respectively. These convolution units are composed simply of arithmetic adders and multipliers. The structure of convolution blocks is shown on Figure 4. The data from 27 neighbor cells is multiplied by the corresponding template coefficients, and then is truncated to 16 bit. After then, the result is added together by 5-pipelined-stages tree adders.

**Figure 5. Datapath circuit architecture**

Our proposed architecture has been implemented on Altera Cyclone II EP2C70F896C6 FPGA using VHDL. The synthesis result of place-and-route and speed are shown in Table 1.
4. Results

To verify the behavior of the system, we have simulated our architecture in ModelSim simulator tool.

For example, by setting zero for initial states of Layer 1(L1) and 3(L3), boundary conditions are set to zero, and templates are chosen as noise removal as below:

\[
A = \begin{bmatrix}
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0
\end{bmatrix},
\]

\[
B = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix},
\]

\[
z = 0
\]

Initial state \((L2) = \begin{bmatrix}
-0.8 & 1.0 & -1.0 & -0.6 \\
1.0 & 1.0 & 1.0 & -1.0 \\
-1.0 & 0.9 & -1.0 & -0.8 \\
-0.9 & -1.0 & -0.7 & -0.8
\end{bmatrix}
\]

Simulation result of this example is shown in Figure 6. In this example, the stable state of cell is given by

\[
x_{i,j,k} = 2y_{i,j,k} + y_{i-1,j,k} + y_{i+1,j,k} + y_{i,j-1,k} + y_{i,j+1,k}
\]

where \(y_{i,m,n}\) are output of cell at steady state. Initial state of layer 2 is similar to initial condition of Chua’s 4x4 arrays as given in Figure 9(a) of [1]. Without interaction with neighbor layers, the output result of Layer 2 is same with the result depicted in Figure 9(c) of [1].

By keeping above templates and initial states and changing template \(A\) as

\[
A = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 1 & 2 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0
\end{bmatrix}
\]

the stable state of cell in the inner-layer becomes

\[
x_{i,j,k} = 2y_{i,j,k} + y_{i-1,j,k} + y_{i+1,j,k} + y_{i,j-1,k} + y_{i,j+1,k} + y_{i,j,k-1} + y_{i,j,k+1}
\]

(8)

And, the outputs corresponding to this defined template are shown in Figure 7.

5. Conclusion
In this paper, we have described an approach to realize the 3D-CNN model with full parallel computation on FPGA platform. We also verify the approach by means of simulation for 3D-CNN with the size of 4x4x3. Future works will be focused on implementation of 3D-CNN with larger sizes.

Acknowledgments

This work has been supported by Ministry of Science and Technology of Vietnam with the project number DTDL.2009G/44. The authors would like to thank Vietnam’s National Foundation for Science and Technology Development (NAFOSTED).

References

Implementation of CNN-based FFT/IFFT Algorithms on FPGA

Hiep Hoang Le, Nguyen Tien Dzung, Thang Manh Hoang

Faculty of Electronics and Telecommunications
Hanoi University of Science and Technology
1 Dai Co Viet, Hanoi, VIETNAM
Email: thang@ieee.org

Abstract-- This paper describes realization of FFT/IFFT algorithms on the CNN model. The implementation for the model is carried out using kit DE2 of Altera and the experimental result demonstrates the effectiveness of proposed model.

1. Introduction

Nowadays, Discrete Fourier Transform/Inverse Discrete Fourier Transform (DFT/IDFT) is used in many applications in the information processing area such as electronics and telecommunications, digital signal processing, and spectral analysis, etc. Due to the complexity in computation of practical applications with large input, realization of DFT/IDFT requires a system with highly computational performance. It is well-known that FFT/IFFT [1] algorithms are useful methods which help to significantly reduce computation time of DFT/IDFT. In most cases, FFT/IFFT has been computed in a sequential fashion such as in hardware platform of digital signal processors, in computers. As a result, it is difficult to get a high speed FFT/IFFT since a large number of elements are used. The speed requirement on the FFT/IFFT can be seen in high speed signal processing systems such as radar, missile, optical communications, etc.

Cellular neural network (CNN) [2] consists of multiple cells which plays a role as processors running in parallel fashion. A cell in CNN locally connects with neighbor ones. CNN offers a potential of real parallel computing. So far, investigation on the idea of utilizing CNN to solve the issue of speed in FFT/IFFT has been limited [3]. As shown in [3], FFT combining with DTCNN (discrete-time CNN) has been proposed and proved theoretically. Cell neighborhood is defined in a sense of functionality, rather than topology which is possible to build many applications, thus it provides extending possibilities of CNN[3].

In this paper, implementation of 16-FFT on CNN will be described by modeling and simulation. The structure of a 16-element input array FFT using CNN model is designed on Quartus II 9.0 SP2. It is implemented on FPGA platform. Due to the similarity in FFT and IFFT algorithms, only concern to FFT is discussed.

2. CNN-Based FFT/IFFT

The basic idea was proposed by Martin Perko et al. [3] and it can be summarized as follows. The DFT formula is

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn}$$  \hspace{1cm} (1)

A decimation-in-time (DIT) radix-4 FFT [1] algorithm is applied in case that N is limited to power of 4 as

$$X[k] = \sum_{r=0}^{N/4-1} x[2r] W_N^{kr} + \sum_{r=0}^{N/4-1} x[2r+1] W_N^{kr}$$

(2)

Let consider an array of N-element inputs; N is limited to power of 4. Utilization of CNN array for N- element FFT or IFFT is shown in Fig. 2. \(L = \log_4 N\) is the number of computational line (levels) with \(N\) cells each.

Where \{x[A], x[B], x[C], x[D]\} are complex input arrays and \{X[A],X[B],X[C],X[D]\} are complex output arrays. The relations between them can be expressed as follows:

$$x[A] = x[A] + W_{2n}x[B] + W_{2n}x[C] + W_{2n}x[D]$$

(3)

$$x[B] = x[A] - W_{2n}x[B] + iW_{2n}x[C] + iW_{2n}x[D]$$

$$x[C] = x[A] + W_{2n}x[B] - W_{2n}x[C] - W_{2n}x[D]$$

$$x[D] = x[A] - W_{2n}x[B] + iW_{2n}x[C] - iW_{2n}x[D]$$

Let consider an array of N-element inputs; N is limited to power of 4. Utilization of CNN array for N-element FFT or IFFT is shown in Fig. 2. \(L = \log_4 N\) is the number of computational line (levels) with \(N\) cells each.

2010 International Symposium on Nonlinear Theory and its Applications
NOLTA2010, Krakow, Poland, September 5-8, 2010
of computational level needed. Each level have \( \frac{N}{4} \) four-terminal operators (or blocks of cells when we using CNN).

Factors \( W_{offs} \) matched with each block cell can be calculated as:

\[
W_{offs}(n,l) = W_{offs}^{n,l}
\]

where \( n \) range from 0 to \( \frac{N}{4} - 1 \) and \( l \) (computation level) range from 0 to \( L-1 \). Fig. 3 shows a 16-FFT using CNN model which has 2 computation lines, 16 complex inputs and 16 complex outputs. Detail of links between output of 1st computation line and input of 2nd computation line of the model can be seen in Table 1 (the outputs and inputs are numbered as in Fig. 3). In a large model, links between two near computation lines can be determined by radix-4 Cooley-Turkey algorithm.

While a normal FFT algorithm calculates one operation at a time, it is seen that CNN-based FFT model performs \( N \log_2 N \) operations simultaneously. It is easy to realize that improvement of computation performance is significant in the case that \( N \) is large.

3. Implementation of CNN-based FFT/IFFT on FPGA

We describe a 16-FFT using CNN model which is developed based on ideas as shown on previous section with hardware description language, VHDL. The chosen tool is Altera’s Quartus II 9.0 SP2. Structure of the proposed model can be seen in Fig. 4. It includes 8 blocks of cells; each block contains 4 pairs of cells and each pair has 2 cells. In a pair of cells, one cell deals with real value and the other does with imaginary value of complex input. Only inputs of cells in a certain block affect states of other cells in the same block.

Cells in the same computation line are synchronized by a source of clock pulse. Characteristic equations for a cell are defined as follow:

\[
\begin{align*}
x_i[k+1] & = \sum_{j=1}^{4} B(i,j) u_j[k] \\
|u_j[k]| & \leq 1 \\
y_i[k+1] & = f(x_i[k]) \\
f(u) & = \frac{1}{2}(|8 + u| - |8 - u|)
\end{align*}
\]

Where \( u_j[k] \) is input value of cell \( j \) (ranges from 1 to 8) in a block cell, \( x_i[k+1] \) is state of cell \( i \) (also range from 1 to 8), \( B(i,j) \) is input control matrix and \( B(i,j) \) is calculated from \( W_{offs} \) as given in Eq.(6). Function \( f(.) \) is usually chosen as a nonlinear function.

Detail of a block cell designed by using ALTERA’s Quartus 9.0 SP2 for FPGA implementation is shown in Fig. 5. It is clear that all of eight cells have the same structure. Inputs of a block of cells are fed inconvolution block of itself, where they are multiplied with respective input control matrix \( B(i,j) \). Sum of the results is state of the cell. Output of each cell is calculated from the state by function \( f(.) \) as reported from previous section. The cells are synchronized by clock pulses. Figure 6 illustrates the implementation on FPGA, we use SRAM on kit DE2 to store sets of input and output data.

4. Result and Discussion

We have created several testbenches to verify the proposed model. Functional simulation of the model has been created using ModelSim-Altera 6.4a. The result shows in Table 2 with the comparison with that of the same input value calculated by normal FFT function using Matlab Software. There, values are converted to complex numbers. The error is due to limit of numerical precision representing parameters for values of inputs, states, etc. In general, it is seen that the model has worked properly as desired.

We have created the same data set as one used in simulation and written it in SRAM, after the implementing the output data set has been verified and the result shows the similarity to simulation result.

The main drawback of this model is in the usage of hardware resource in case of a large model. Generally, a model with \( N \)-element inputs requires \( N \log_2 N \) cells.

About Effectiveness of the CNN-based FFT model, it carries out multiple calculation operations at the same time. Specifically, \( N \log_2 N \) operations can be calculated simultaneously. However, there are two factors limiting the computation performance. Firstly, additional operations are required i.e. calculation and update for state of cells and for outputs. Secondly, there is delay in synchronizing of calculation in cells. However, synchronizing also allow a larger model works properly regardless to the scale of model.

In this realization, the clock pulse is 50 MHz, and it has taken 2 clock pulses to compute a data set, i.e. data set rate is 25x10^6 data set per second (there are 32 inputs in one data set and we used 24 bits to present one input, so data bit rate in input of serial to parallel block can reach up to 32x25=800Mb/s). The result will be very impressive if a parallel input source or serial extra-high speed bit rate is created as in optical OFDM. In case of an expansion to 1024 FFT from this model, data set rate is 10^9 data set/s (in theoretical, data bit rate in STP block can reach 1024x10=20Gb/s).

<table>
<thead>
<tr>
<th>Output of 1st computation line</th>
<th>0</th>
<th>4</th>
<th>8</th>
<th>12</th>
<th>1</th>
<th>5</th>
<th>9</th>
<th>13</th>
<th>2</th>
<th>6</th>
<th>10</th>
<th>14</th>
<th>3</th>
<th>7</th>
<th>11</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input of 2nd computation line</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
</tr>
</tbody>
</table>
\[
B_g = \begin{bmatrix}
1 & 0 & \text{Re}(W_{g,0}^2) & -\text{Im}(W_{g,0}^2) & \text{Re}(W_{g,0}^1) & -\text{Im}(W_{g,0}^1) & \text{Re}(W_{g,0}^3) & -\text{Im}(W_{g,0}^3) \\
0 & 1 & \text{Re}(W_{g,0}^2) & \text{Im}(W_{g,0}^2) & \text{Re}(W_{g,0}^1) & \text{Im}(W_{g,0}^1) & \text{Re}(W_{g,0}^3) & \text{Im}(W_{g,0}^3) \\
1 & 0 & -\text{Re}(W_{g,0}^2) & \text{Im}(W_{g,0}^2) & -\text{Re}(W_{g,0}^1) & \text{Im}(W_{g,0}^1) & -\text{Re}(W_{g,0}^3) & \text{Im}(W_{g,0}^3) \\
0 & 1 & -\text{Im}(W_{g,0}^2) & -\text{Re}(W_{g,0}^2) & -\text{Im}(W_{g,0}^1) & -\text{Re}(W_{g,0}^1) & -\text{Im}(W_{g,0}^3) & -\text{Re}(W_{g,0}^3) \\
1 & 0 & \text{Re}(W_{g,0}^2) & \text{Im}(W_{g,0}^2) & \text{Re}(W_{g,0}^1) & \text{Im}(W_{g,0}^1) & \text{Re}(W_{g,0}^3) & \text{Im}(W_{g,0}^3) \\
0 & 1 & \text{Re}(W_{g,0}^2) & \text{Im}(W_{g,0}^2) & \text{Re}(W_{g,0}^1) & \text{Im}(W_{g,0}^1) & \text{Re}(W_{g,0}^3) & \text{Im}(W_{g,0}^3) \\
1 & 0 & -\text{Re}(W_{g,0}^2) & \text{Im}(W_{g,0}^2) & -\text{Re}(W_{g,0}^1) & \text{Im}(W_{g,0}^1) & -\text{Re}(W_{g,0}^3) & \text{Im}(W_{g,0}^3) \\
0 & 1 & -\text{Im}(W_{g,0}^2) & -\text{Re}(W_{g,0}^2) & -\text{Im}(W_{g,0}^1) & -\text{Re}(W_{g,0}^1) & -\text{Im}(W_{g,0}^3) & -\text{Re}(W_{g,0}^3)
\end{bmatrix}
\]

Figure 3. 16-FFT using CNN model

Figure 4. Block diagram describes structure of the model
5. Conclusion

The paper presents implementation of CNN-based FFT/IFFT on FPGA. The result has shown that implementation of the model gives a potential extra-high speed computation. Furthermore, the proposed model can be developed to larger number of N for various applications.

<table>
<thead>
<tr>
<th>Table 2: Simulation result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>0.9293</td>
</tr>
<tr>
<td>0.3500</td>
</tr>
<tr>
<td>0.1966</td>
</tr>
<tr>
<td>0.2511</td>
</tr>
<tr>
<td>0.6160</td>
</tr>
<tr>
<td>0.4733</td>
</tr>
<tr>
<td>0.3517</td>
</tr>
<tr>
<td>0.8308</td>
</tr>
<tr>
<td>0.5853</td>
</tr>
</tbody>
</table>

Acknowledgments

This work has been supported by Ministry of Science and Technology of Vietnam with the project number DTDL2009G/44. The authors would like to thank Vietnam’s National Foundation for Science and Technology Development (NAFOSTED).

References

Advanced PI/SI/EMI Simulation Technology for High-Speed Electronic Design

- Toward chip/package/board co-design -

Hideki Asai
Shizuoka University
hideasai@sys.eng.shizuoka.ac.jp

Abstract - With the progress of system integration technology, a variety of noise problems, so-called power/signal integrity and EMI issues, have become very serious in the field of chip (LSI), package and board (PCB) design. These problems cause frequently the unexpected behaviors on the today’s high-density and high-speed circuits. Thus, useful remedies are strongly demanded and the development of novel simulation technology has been expected for the short TAT (turn around time) and the cost reduction of a variety of high-performance electronic products.

In this report, the historical overview and the present status of the power/signal integrity problems and electrical simulation technology in the high-speed digital era are described. Furthermore, the future trend, including 3-dimensional and full-wave simulation techniques for PI/SI/EMI design, is suggested for the total solution of chip/package/board co-design.

1. Introduction

With the progress of circuit integration technology, the performance of electronic products has been improved drastically for these years. This fact has required much more skilful design because a variety of noise problems, which have not been considered previously, have become very serious. Noise issues are classified into 3 categories, namely, signal integrity (SI), power integrity (PI) and electromagnetic interference (EMI) [1-3].

In 1990’s, discussions of SI problems such as time delay, reflection and cross talk have been started for the high-speed signal propagation on the interconnects. Furthermore, in 2000’s, a variety of researches have been done actively for the analyses and the remedy for the PI and EMI problems such as simultaneous switching noise, ground bounce and radiation.

For the efficient design with consideration of SI/PI/EMI issues, simulation technologies get a lot of attention. Especially, novel electrical simulation technologies have been strongly demanded for the short TAT and the cost reduction, because these noise effects cause frequently unexpected behaviors and large damages on the high performance electronic products.

It is well-known that SPICE is conventionally one of the most useful tools to simulate the electrical behaviors [4], which is the lumped parameter circuit simulator. However, for more detailed verification of the high-density packaging and high-speed signals, the simulation should be done with the consideration of the 3-dimensional structures of interconnects, power/ground planes and so on.

In this report, first, today’s problems in the chip/package/board co-design are summarized, and the present status and future trend of electrical simulation technology in high-speed digital era are described.

2. PI/SI/EMI and Chip/Package/Board Co-Design

2.1 What is high-speed signal?

“High-speed” means that the signal includes high-frequency components. In such a case, we have to consider the circuit as a distributed parameter system, not as a lumped element system. When the length of the transmission line is long for the wave length of the signal, signal integrity becomes one of the most serious problems. Therefore, the line should be modeled by partial differential equations differently from the ordinary differential equations.

If the signal is sinusoidal wave, we need generally to regard the line as a transmission line in the case that the line length is longer than about 1/10 of the wave length. Therefore, for the signal of 1GHz, the line should be designed as a distributed parameter system when the line is longer than about 30 cm, namely, 1/10 of the wave length, $3\times10^{10}$ cm/10=30 cm. Furthermore, for the line in the dielectrics, the length to be regarded as the distributed parameter system is shorter than 3 cm, since the velocity of electronic signal on the line is proportional to $1/\sqrt{\varepsilon}$, where $\varepsilon$ is permittivity of dielectrics.

2.2 Chip/package/board co-design and PI/SI/EMI

Generally, LSI is fabricated on the silicon, and the chip is packaged and then it is mounted on the printed circuit board (PCB). In other words, circuit design is basically classified to 3 categories, namely, chip, package and board levels as shown in Fig.1.
The circuit design is categorized by analog and digital classes. However, with speedup (high frequency) of circuit operation, the boundary between analog and digital classes would be unclear and thus guarantee of signal quality (signal integrity) is also required for digital signals as well as analog ones.

Fig.1 Power/signal integrity problems.

In 1980’s, noise problem of IC was corresponding to the signal time-delay in the transistor. However, currently, signal propagation on interconnects is much larger problem. Furthermore, according to miniaturization of the circuit, a variety of noises, namely, reflection, cross-talk and IR drop became serious at the chip level. At the package and board levels, ground bounce such as SSN (simultaneous switching noise) is a very serious problem in the high-speed signaling because power/ground lines (planes) are connected each other in the system which is composed of chip, package and board. In order to complete the efficient design for the total system, advanced simulation technology, which can simulate the system in a body, is strongly demanded.

At the board design, typically, the line width is 50-100 [um], and 20-30 [um] at the package design. On the other hand, it is several dozen [nm] at the chip level. These scale differences raise a very serious problem in the 3-D physical simulation of the chip/package/board co-design. Currently, we have no practical tools to simulate the system in a body. In order to reduce the simulation cost, a variety of macro-modeling methods have been proposed. However, usually, some kinds of approximation techniques are used in these methods, thus novel physical simulation methodologies are expected for the detailed verification.

For higher frequency, modeling should be done by using Maxwell’s equations based on electromagnetic theory, and 2-D and 3-D physical simulations should be done. Consequently, in order to simulate exactly the electrical behaviors, hybrid modeling and simulation based on lumped element, distributed parameter systems and electromagnetic theory are required.

In future, LSI-CAD (Computer-Aided Design)/CAE (Computer-Aided Engineering) will be combined with EMC-CAD/CAE. As a result, multi-level simulator should be developed. Furthermore, CAD/CAE system evolves to multi-physic CAD/CAE, which includes a variety of fields, namely, not only LSI and EMC but also thermal and mechanical fields and so on.

3. A Variety of Electrical Expressions

In the multi-level simulator as described above, a variety of electrical expressions are required as shown in Fig.2. Electronic circuits are classified into digital circuits and analog ones. A digital circuit is described by a set of logic functions (Boolean algebra) and an RLC lumped-element analog circuit in which many transistors are also included is formulated by a system of ordinary differential equations (ODEs). Because this type of circuit behaviors can be expressed reasonably as lumped-element circuit equations according to Kirchhoff’s law, these formulations are suitable for the consideration of analog behaviors.

On the other hand, the lines (interconnects) should be modeled as the partial differential equations (PDEs) when they must be treated as transmission lines. Furthermore, expressions based on the Maxwell’s equations are frequently required at the packaging/mounting level since the issues of power integrity and EMI are regarded as electromagnetic field problems in the high frequency domain.

Fig.2 A variety of electrical expressions.

Because the total calculation cost has the trade-off for the simulation accuracy that is severely dependent on the modeling method, modeling and expressions such as ODEs, PDEs and Maxwell’s equations have to be selected comprehensively. Furthermore, we have to use each of them skillfully and appropriately.
4. What Can Spice Do and Not Do?

SPICE (Simulation Program with Integrated Circuit Emphasis)-like circuit simulator for lumped parameter system has been widely used conventionally as a tool for signal integrity analysis of electronic circuits. By using SPICE, DC analysis, AC analysis and transient analysis of a circuit can be done. Although SPICE is effective for timing analysis of an electronic circuit, it spends usually an enormous calculation cost for the transient analysis of the large-scale network because it uses matrix solver.

In the design of high-frequency signaling, circuits must be simulated including the behaviors of interconnects precisely. SPICE possesses transmission line models and a variety of noises on interconnects can be simulated. For these years, power integrity (PI) problem such as ground bounce caused by simultaneous switching noise in the digital part is also serious. For the PI problems, the ground has to be modeled precisely for the exact simulation because the ground line and plane themselves fluctuate [1][3]. In other words, a ground plane must be modeled as 2- or 3-dimensional structure which is composed of a large number of passive RLCG elements. However, a netlist of SPICE becomes enormous and the calculation cost becomes impractical if the ground is modeled in detail. Therefore, some kind or another remedy for this problem is required.

As one of the remedies, model order reduction (MOR) techniques have been studied [5-6]. In MOR-based simulation methods, the large linear part is modeled as an equivalent and small macro-model. As a result, total simulation cost can be reduced drastically. These techniques are available for not only the MNA (Modified Nodal Approach)-based simulations but also electromagnetic field simulations[7]. However, these methods exploit some kind of approximation and are not efficient for the network with lots of ports. Thus, new strategies are strongly demanded.

5. Future Trend and Next Generation Simulators

In this section, recent some results of acceleration techniques are described for large-scale circuit simulations.

5.1 Fast circuit simulation methods

It is well-known that a class of relaxation method is available for the large-scale circuit simulation and a variety of techniques have been proposed in 1980’s [8-9]. Furthermore, the LIM (Latency Insertion Method), which adopts a leap-frog algorithm which is used in FDTD (Finite-Difference Time-Domain) [10] method, has been proposed for the efficient simulation of large-scale RLGC networks [11], and it has been also shown that this method is suitable for the parallel-distributed simulation [12] and the multi-rate latency technique can be exploited [13]. After that, the implementation on the cloud computing system and the GPU (Graphics Processing Unit) has been done. Parallel-distributed LIM on the cloud computing system composed of 32 CPUs can be about 25 times faster than the original LIM [14]. LIM on CUDA with GPU (Geforce GTX295 composed of 240 streaming processors) is more than 30 times faster than the original LIM on the single CPU (Intel Xeon 3.2GHz) [15]. Although LIM is much faster than SPICE-like method, it is not directly applicable to the strongly coupled multi-conductor transmission lines.

In order to cope with this problem, block LIM has been proposed [16], and the parallel-distributed block LIM has been also suggested [17]. Block LIM was used for the fast circuit simulator, SPIDER, as shown in Fig.3.

Fig. 3 Fast circuit simulator SPIDER.

5.2 Fast electromagnetic simulation methods

For these years, 3D electromagnetic simulation methods have been studied for power integrity and EMI issues. FDTD method is simple and suitable for parallel-distributed calculations, thus 3D EMI simulator, BLESS, has been developed for the board design and estimation, through the collaboration between Shizuoka University and SONY, and sufficiently used for the practical PCB design [17]. Furthermore, GPGPU-based FDTD method has been also studied [18], where it is reported that GPGPU-FDTD is about 30 times faster than the conventional FDTD method with single CPU.

FDTD method is a very useful numerical simulation technique, but Courant-Friedrich-Levy (CFL) condition must be satisfied when this method is used, that is to say, the maximum time-step size is limited by the minimum cell size in the simulation.
domain. In order to avoid the constraints from CFL condition, various implicit FDTD methods such as ADI (Alternating-Direction Implicit) -FDTD and HIE (Hybrid Implicit-Explicit) -FDTD methods have been proposed \[20-23\]. We have also proposed the alternating direction explicit (ADE) FDTD method \[22-23\], and the 3-dimensional EMI simulator has been developed as shown in Fig.4.

![3D electromagnetic simulator](image)

**Fig.4** 3D electromagnetic simulator.

### 6. Conclusions

In this report, the historical overview and the present status of the power/signal integrity verification methods have been described. Furthermore, the future trend and the next generation techniques, including 3-dimensional and full-wave techniques, have been suggested for the total solution of chip/package/board co-design.

In future, development of a multi-level simulator for SI/PI/EMI verification is expected for chip/package/board co-design.

### References


Effects of chaotic fluctuation on the Hopfield model for the traveling salesman problem

Ayano Hirahara and Osamu Araki

Graduate School of Science, Tokyo University of Science
Kagurazaka 1-3, Shinjuku-ku, Tokyo, 162-8601 Japan
Email:j1510629@ed.kagu.tus.ac.jp

Abstract—In previous studies, it has been shown that the chaotic or stochastic fluctuation which has a negative and declining autocorrelation contributes to closer reach to the optimum solution in the Hopfield net. In this study, we examine this property from another viewpoint of evaluation. We evaluate the performance by reaching speed to the neighborhood of the optimum solution for a constant learning period. We made the Hopfield net solve the traveling salesman problem with chaotic fluctuations, which are generated from the logistic map and Bernoulli map. The results show that negative autocorrelations of fluctuations may contribute to better performance in both maps, which corresponds to the results of previous studies. In addition, they suggest that even periodic signals can contribute to the performance, but intermittency never does.

1. Introduction

Hopfield neural network [1] has been used for solving combinatorial optimization problems with the theory of minimizing the energy. However, in most of the cases, the solution tends to be dropped into local minimum and cannot get out of it, so that we cannot attain the optimum solution. In order to avoid such situation, many methods in which various fluctuations are supplied to Hopfield neurons have been proposed. In recent years, the chaotic fluctuation helps to get better performance for searching an optimum solution in neural network [2], [3], and [4]. For example, Hayakawa and Sawada showed that the chaotic fluctuation which was generated from logistic map is able to gain the better performance [2]. Furthermore, using the logistic map, Hasegawa et al. insisted that the negative autocorrelation should be the most important factor to improve the performance [5]. In this study, we examine this property from another viewpoint of evaluation. We evaluate the performance by reaching speed to the neighborhood of the optimum solution for a constant learning period. We made the Hopfield net solve the traveling salesman problem (abbr. TSP) with chaotic fluctuations. The fluctuation signal sequence is generated from modified Bernoulli map [6] as well as the logistic map. The results show that negative autocorrelations of fluctuations may contribute to better performance in both maps, which corresponds to the results of previous studies. In addition, they suggest that even periodic signals can contribute to the performance, but intermittency never does.

2. Methods

2.1. Neural Network Model

In Hopfield Network, we aim to minimize an energy function which is defined in the network as follows:

\[ E = A \sum_{i=1}^{N} \left( \sum_{k=1}^{N} x_{ik}(t) - 1 \right)^2 + B \sum_{k=1}^{N} \left( \sum_{i=1}^{N} y_{ik}(t) - 1 \right)^2 \\
+ D \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{N} d_{ij} x_{ik}(t) x_{jk}(t+1) 
\]

(1)

where \( N \) is the number of cities, \( d_{ij} \) is the distance between \( i \)th city and \( j \)th city, and \( A, B, \) and \( D \) are constants. The first and second terms are the constraint terms and the third is the objective term, respectively. In order to decrease this energy, all neurons are updated by the following equation:

\[ x_{ik}(t+1) = f \left[ \sum_{j=1}^{n} \sum_{k=1}^{N} w_{ikj} x_{jk}(t) + \theta_{ij} \right] + \beta z_{ik}(t). \]

(2)

where \( x_{ik}(t) \) is the output of the \((i, k)\)th neuron, \( w_{ikj} \) is a connection weight between the \((i, k)\)th and \((j, l)\)th neurons, \( \theta_{ij} \) is the threshold of the \((i, k)\)th neuron, \( \beta \) is the amplitude of a fluctuation, and \( z_{ik} \) is the fluctuation for the \((i, k)\)th neuron. The \((i, k)\)th neuron represents that \(i\)th city is visited at \(k\)th order. According to Eqs (1) and (2), the connection weight \( w_{ikj} \) and a threshold \( \theta_{ij} \) are shown in following equations:

\[ w_{ikj} = -A \delta_{ij} (1 - \delta_{ik}) + B \delta_{ij} (1 - \delta_{ik}) \\
- D d_{ij} (\delta_{ik+1} + \delta_{i-k-1}). \]

(3)

\[ \theta_{ij} = A + B, \]

(4)

where \( \delta_{ij} = 1 \) when \( i = j \), otherwise \( \delta_{ij} = 0 \).
2.2. Fluctuation

In the previous research, the chaotic fluctuation generated by the logistic map helps neural network to raise the frequency of getting the optimum solution [4]. In this study, we use two kinds of fluctuations generated by two nonlinear dynamical maps: the logistic map and the modified Bernoulli map. The logistic map is shown in the following equation:

\[ z(t + 1) = az(t)(1 - z(t)) \] (5)

where \( a \) is a bifurcation parameter. As \( a \) changes with this range, \( 0 \leq a \leq 4 \), \( z(t + 1) \) behaves either chaotically or not. We focus on the range of \( a = 3.6 \) to \( 4.0 \), which generates chaotic or periodic fluctuations. The modified Bernoulli map [6], which is shown in the following equation:

\[
\begin{cases} 
  z(t) + 2^{a-1}(1 - 2\epsilon)z(t) + \epsilon & (0 \leq z(t) \leq \frac{1}{2}) \\
  z(t) - 2^{a-1}(1 - 2\epsilon)(1 - z(t)) - \epsilon & \left( \frac{1}{2} < z(t) \leq 1 \right) 
\end{cases}
\] (6)

where \( b \) is a bifurcation parameter and as \( b \) changes with this range, \( 0 \leq b \leq 3 \), \( z(t) \) behaves intermittently chaotically or not, and \( \epsilon \) is equal to \( 1.0 \times 10^{-13} \).

2.3. Evaluation of Performance

After we executed the iterations of updates for 100 times in each TSP, we counted the number when the energy will decrease below \( E_0 = 9.0 \). We regard the rate of counts to the total number of trials as the performance in this study.

3. Simulation Results

3.1. Effects of Fluctuations

The energy after learning depends on what kind of fluctuations we choose. We compared the performance with five different fluctuations: uniform and random numbers, chaotic fluctuations from logistic map with parameter \( a = 3.92 \) and \( a = 3.95 \), and those from Bernoulli map with parameter \( b = 0.2 \) and \( b = 2.0 \). 18-city TSP is solved with \( A = 1.0, B = 1.0, D = 1.0, \) and \( \beta = 0.05 \).

Figure 1 shows the relationship between the amplitude of fluctuations and performance. Where the amplitude is 0.05, there are peaks when \( a = 3.95 \) and \( b = 0.2 \). On the other hand, the performance with random fluctuations monotonically decreases. This result shows that chaotic fluctuation promotes the performance better than random fluctuation does.

3.2. Effects of Stochastic Properties

Now, we use the surrogate data method [7] in order to clarify which stochastic property of chaotic fluctuations contributes to raise the performance. We adopted three kinds of data. We call the fluctuations which remains to be produced by logistic map "Original data (abbr. Original)\(^\text{\!*}\)", the one which is displaced randomly "Random shuffled (abbr. RS)\(^\text{\!*}\)", and the one which is shuffled but the frequency spectrum is remained "Fourier transformed (abbr. FT)\(^\text{\!*}\).

Figure 1: The effects of chaotic fluctuations in 18-city TSP.

Figure 2: The effects of modified logistic-fluctuation. 18-city TSP is solved with \( A = 1.0, B = 1.0, D = 1.0, \) and \( \beta = 0.05 \).

Figure 3: The bifurcation diagram of logistic map.

Figure 2 shows that FT fluctuation succeeded secondly. Also especially during the periodical fluctuation \( (3.83 \leq a \leq 3.86) \), the performance gets greater. We are also able to compare Fig. 2 with Fig. 3 to see the performance is better when \( z(t) \) behaves periodically. However, with RS fluct-
tuation, because its autocorrelation is destroyed, the performance holds low level. Therefore the result shows that keeping an autocorrelation makes better performance.

Next, we try the other nonlinear dynamical map, Bernoulli map. First, we show how the performance changes as the bifurcation parameter $b$ increases.

![Figure 4: The effects of modified Bernoulli-fluctuations.](image)

Figure 4 shows that the smaller parameter $b$ is, the better performance is gained. This may be because the intermittency increases as the value of $b$ increases. Now, we also adopt the surrogate data method to the chaotic fluctuations from Bernoulli map to see which stochastic property contributes to gain high performance. RS performed in the same way regardless of parameter $b$. On the other hand, Original and FT show high performance only when $b$ is small.

### 3.3. Autocorrelation of Fluctuations

To analyze the reason why chaotic and FT-surrogate fluctuations are better than RS, we calculated the autocorrelations of the fluctuation signals. Fig.5 shows the autocorrelations of logistic fluctuations when $a = 3.65$, $3.82$, and $3.84$.

![Figure 5: Autocorrelations of various logistic-fluctuations.](image)

When $a = 3.82$, negative and declining autocorrelation clearly emerges. At that time, the performance is also the best.

In the case of modified Bernoulli-fluctuations, we plot the autocorrelations when $b = 0.2$ and $0.8$. When $b = 0.2$, the same property of autocorrelation is clear and the performance is the best.

![Figure 6: Autocorrelations of various Bernoulli-fluctuations.](image)

### 4. Conclusion

We made the Hopfield net solve the TSP with chaotic fluctuations, which are generated from the logistic map and modified Bernoulli map. Then we evaluate the performance by reaching speed to the neighborhood of the optimum solution for a constant period. The results show that the negative autocorrelation of fluctuations is able to increase the speed towards the optimum solution in spite of the kind of maps. In addition, they suggest that even periodic signals can contribute to the performance in some cases, while intermittent signals of modified Bernoulli do not contribute at all. Further analyses are needed to these phenomena. And the relationship between the results and the way of evaluation is also one of our future works.

### References


and Stochastic Noise with Negative Autocorrelation,”


Realizing Ideal Chaotic Dynamics for Combinatorial Optimization
using a Spatiotemporal Filter

Mikio Hasegawa†
†Dept. of Electrical Engineering, Tokyo University of Science
1-14-6 Kudankita, Chiyoda-ku, Tokyo, 102-0073, Japan

Abstract—This paper proposes an optimization algorithm, which utilizes the most stable spatiotemporal chaotic dynamics for solution search in a high dimensional space. Such chaotic dynamics is generated by a FIR filter, which has been applied to the chaotic CDMA in previous researches to minimize the cross-correlation among the sequences. In the proposed method, such filters are introduced at the output of decision functions of combinatorial optimization algorithms to realize an ideal chaotic search, which generates ideally complicated searching dynamics. In this paper, the proposed scheme is applied to two combinatorial optimization approaches, the Hopfield-Tank neural network with additive noise and a heuristic algorithm based on the neighboring solution search, which solve the Traveling Salesman Problems and the Quadratic Assignment Problems. Simulation results show that the proposed approach using the ideal chaotic dynamics simply improves the performance of the chaotic algorithms without searching appropriate parameter values even for large-scale problems.

1. Introduction

Chaotic dynamics have been shown effective for combinatorial optimization problems by many researches [1]–[8]. There are two major approaches using chaotic dynamics to avoid trapping at undesirable local minimum solutions. The first one introduces the chaotic fluctuation to the Hopfield-Tank neural network [9], and the second one drives local search heuristics by the chaotic dynamics. Although the first approach is applicable only to very small toy problems, the second approach can solve much more difficult and large-scale problems by introducing simple neighboring solution search heuristics and has been shown more effective than the conventional heuristic algorithms, such as the stochastic searches and the tabu searches [3, 4, 5].

In the previous researches, effectiveness of such chaotic dynamics for combinatorial optimization has been analyzed, and several important characteristics have been found [6, 8]. In the approach that adds chaotic sequences to each neuron in the Hopfield-Tank neural network solving an optimization problem [7], it has been clarified that the most important factor for high performance of the chaotic noise is a specific autocorrelation of the chaotic dynamics, by an analysis based on the method of surrogate data [8]. This effective chaotic noise has the autocorrelation with a negative value in lag 1 and damped oscillation. Such chaotic dynamics with negative autocorrelation has been also utilized in the chaotic CDMA [10, 11]. In those researches, the chaotic dynamics, whose autocorrelation of the sequences becomes $C(\tau) \approx C \times (-\tau)^r$, $r = -(2 - \sqrt{3})$, is used because the cross-correlation among the sequences becomes smallest by such autocorrelation. In Ref. [12], effects of the sequences, which has negative autocorrelation with damped oscillation, has been applied also to the combinatorial optimization problems, and it has been shown that the noise sequences having negative autocorrelation improves the performance of the neural networks to the same level as those with the chaotic noise. Since such negative autocorrelation noise minimizes cross-correlation, lower cross correlation may be also important for realizing complex spatiotemporal solution search in combinatorial optimization problems.

In the chaotic CDMA [11], such ideal chaotic noise, whose autocorrelation is $C(\tau) \approx C \times (-\tau)^r$, $r = -(2 - \sqrt{3})$ has been generated by a FIR filter. It has been shown that such a filter improves bit error rate of the CDMA communication system for various spreading sequences, such as stochastic sequences and deterministic sequences.

In this paper, such a FIR filter making negative autocorrelation is applied to the combinatorial optimization methods, for realizing ideal spatiotemporal searching dynamics. Our proposed algorithm minimizes cross-correlation among the updating dynamics of neighboring solution search, and realizes ideally complex spatiotemporal searching dynamics. The proposed scheme is applied to two optimization approaches, the Hopfield-Tank neural networks with additive noise and the 2-opt heuristic method, solving the Traveling Salesman Problems (TSPs) and the Quadratic Assignment Problems (QAPs). Effectiveness of the proposed approach is investigated also for the large-scale problems up to 1173-city TSP.

2. Performance of the Optimization Neural Networks with Ideal Spatiotemporal Chaotic Dynamics

First, the effectiveness of the ideal chaotic dynamics is evaluated in the Hopfield-Tank neural network approach. This approach is based on the minimization of the energy
function of the neural networks by asynchronous update of each neuron. However, since the original Hopfield-Tank neural networks stop search at a local minimum, the chaotic noise and other stochastic dynamics has been added to the neurons to avoid trapping at such undesirable states and to achieve much higher performance [7, 8].

In this paper, such a neuronal update function with additive noise is defined as follows,

$$x_i(t + 1) = f[\sum_{j=1}^{N} \sum_{l=1}^{N} w_{ijkl} x_j(t) + \theta_{kl} + \beta z_{ij}(t)],$$

where $x_i(t)$ is the output of the $(i, k)\text{th}$ neuron at time $t$, $w_{ijkl}$ is the connection weight between the $(i, k)\text{th}$ and $(j, l)\text{th}$ neurons, $\theta_{kl}$ is the threshold of the $(i, k)\text{th}$ neuron, $N$ is the number of cities, $z_{ij}(t)$ is a noise sequence added to the $(i, j)\text{th}$ neuron, $\beta$ is the amplitude of the noise, and $f$ is the sigmoidal output function, $f(y) = 1/(1 + \exp(-y/\epsilon))$. The noise sequence used for $z_{ij}(t)$ is normalized to zero mean and unit variance.

To apply this neural network to the TSP, the connection weights $w_{ijkl}$ and the thresholds $\theta_{ij}$ are set as follows,

$$w_{ijkl} = -A[\delta_{ij}(1 - \delta_{kl}) + \delta_{il}(1 - \delta_{kj})] - Bd_{ij}(\delta_{ik+1} + \delta_{i(k-1)}),$$

$$\theta_{ij} = 2A,$$

where $d_{ij}$ is the distance between the cities $i$ and $j$, $A$ and $B$ are the weight of the constraint term (formation of a closed tour) and the objective term (minimization of total tour length), and $\delta_{ij}$ is the Kronecker delta, respectively.

For solving the QAPs, whose objective function is

$$F(p) = \sum_{i=1}^{N} \sum_{j=1}^{N} a_{ij} b_{pi} b_{pj},$$

By the Hopfield-Tank neural networks, the connection weights and the thresholds have to be set as follows,

$$w_{ijkl} = -A[\delta_{ij}(1 - \delta_{kl}) + \delta_{il}(1 - \delta_{kj})] - B a_{ij} b_{kl},$$

$$\theta_{ij} = 2A.$$

Figs. 1 and 2 show the solvable performances of the above neural networks applied to the TSP and the QAP, with various noise sequence for $z_{ij}(t)$, such as the white Gaussian noise, the Chebyshev map chaos with the same dimension for all of neurons, that with different dimension for each neuron, and the logistic map chaos, $z_{ij}(t + 1) = az_{ij}(1 - z_{ij}(t))$, with $a = 3.82, a = 3.92$ and $a = 4$. The abscissa axis is an amplitude of the noise $\beta$ in Eq. (1). The solvable performance on the coordinate is the percentage of successful runs obtaining the optimum solution in 1000 runs with different initial conditions. The successful run obtaining the optimum solution means that the optimum solution state is found at least once in a fixed iteration. The cutoff iterations for each run are set to 1024 for TSP and to 4096 for QAP, respectively. The parameters of the neural networks are $A = 1, B = 1, \epsilon = 0.3$ for the TSP, and $A = 0.35, B = 0.2, \epsilon = 0.075$ for the QAP, respectively. The problems introduced in Figs. 1 and 2 are a 20-city TSP in [8] and a QAP with 12 nodes, Nug12 in QAPLIB [13].

The results in Figs. 1 and 2 show that the neural networks with the logistic map with $a = 3.82$ and $a = 3.92$ perform better than other noise, for both problems. From Fig. 3 showing autocorrelation coefficients of each noise, the sequences having the better performance, the logistic map with $a = 3.82$ and $a = 3.92$, have negative autocorrelation. On the other hand, autocorrelation of the others are almost zero. From these results, it is clear that the negative autocorrelation is very important for improving the performance of the combinatorial optimization algorithm based on the Hopfield-Tank neural networks.
3. Realizing Ideal Spatiotemporal Searching Dynamics by FIR Filter

In order to realize ideal searching dynamics which has negative autocorrelation, this paper proposes a novel scheme to apply the FIR filter to the decision function of the optimization algorithms. In the followings, first such a filter is applied to the Hopfield-Tank neural network approach, and then it is also applied to the 2-opt heuristics for the large-scale TSPs.

3.1. Ideal Spatiotemporal Searching Dynamics for the Hopfield-Tank Neural Networks

In order to make the outputs of the neurons having lower cross-correlation and to realizing ideally complex searching dynamics, the following FIR filter is introduced to make outputs having negative autocorrelation, which have been applied to the chaotic CDMA in Ref. [11],

\[ \hat{f}(t) = \sum_{u=0}^{M} r^u f(t-u). \]  

(7)

By setting \( r = -(2 - \sqrt{3}) \), ideal sequences to minimize the cross correlation have been generated for the chaotic CDMA. This paper introduces such a filter to the output function of each neurons, which can be described as follows, by replacing the output function \( f \) in Eq. (1) with the following equation,

\[ f(y(t)) = 1/(1 + \exp(-\sum_{u=0}^{M} r^u y(t-u)/\epsilon)). \]  

(8)

In the following experiments, \( M = 8 \) for each noise.

The solvable performance of the neural networks with the novel output function generating the ideal spatiotemporal chaotic sequence is shown in Figs. 4 and 5. From these results, we can see that the performances of the neural networks with the noise, whose autocorrelation is almost zero, could be much improved for all of the cases. In Fig. 5 for the QAP, the performance becomes even better than the logistic map chaos with \( a = 3.82 \) or \( a = 3.92 \), that have the best performance when the filter is not applied shown in Fig. 2.

From Figs. 4 and 5, for the noise sequences which already have negative autocorrelation, the logistic map with \( a = 3.82 \) and \( a = 3.92 \), \( r = 0 \) is the best. On the other hand, for other noise sequences having zero autocorrelation, the best value of autocorrelation parameter \( r \) for the FIR filter is around \( -(2 - \sqrt{3}) \), which is the same value as that used to generate optimal sequences to minimize the cross correlation in the chaotic CDMA researches. From this result, such negative autocorrelation may ideally minimize the cross correlation in this asynchronously updated neural network. Minimization of the cross correlation among the neurons makes ideally complex search in the searching space and the performance is much improved.
on the chaotic CDMA [11] have shown that such filtered sequences having autocorrelation $C(\tau) \approx C \times (-r)^{\tau}$ with $r$ around $-(2 - \sqrt{3})$, has the lowest cross-correlation. In the proposed algorithm, such lower cross-correlation makes very complicated searching dynamics, and the performance of the algorithms with such filter can be improved.

Our algorithm is very simple and no need to find the appropriate parameter values carefully. Moreover, it is possible to be applied to various algorithms. Therefore, it should be an interesting future work to apply this proposed approach to various problems and various algorithms.

References


A Method for Solving Very Large Scale TSPs by Chaotic Dynamics

Tohru Ikeguchi†, Shun Motohashi‡, and Takafumi Matsuura†‡

†Department of Information and Computer Sciences, Saitama University
255 Shimo-okuburo, Sakura-ku, Saitama 338-8570, Japan
‡Department of Management Science, Tokyo University of Science
1-3 Kagurazaka, Shinjuku-ku, Tokyo 162-8601, Japan
Email: 1tohru@mail.saitama-u.ac.jp, 2matsuura@ms.kagu.tus.ac.jp

Abstract—To solve traveling salesman problems (TSPs), we have already proposed two chaotic search methods, the Lin-Kernighan (LK) algorithm controlled by the chaotic dynamics (CS-LK) and the stem-and-cycle (S&C) ejection chain method controlled by the chaotic dynamics (CS-SC). The basic concept of the methods is that chaotic dynamics can effectively resolve a local minimum problem intrinsic to the heuristic algorithm. Although the CS-SC shows higher performance than the CS-LK, it is not so easy to apply the CS-SC to large scale instances because this method takes much calculation time to find good results. It is also important to develop an effective algorithm which finds good near-optimal solutions with less calculation time. In this paper, we propose a new method for solving very large scale TSPs with the order of 10^5 cities. In the method, we introduced a hybrid strategy by combining these two chaotic search methods. We used large scale instances form TSPLIB to evaluate the proposed method. We show that the proposed method exhibits very small gaps from the optimal solutions with less calculation time.

1. Introduction

We are often asked to solve various combinatorial optimization problems in our daily life; for example, scheduling, delivery planning, circuit design, computer wiring, and so on. Although these problems are ubiquitous and easy to describe, it is usually hard, if not impossible, to find their optimal solutions. These facts indicate that it is inevitable to design effective algorithms for solving combinatorial problems.

To develop an effective algorithm for the combinatorial optimization problems, the traveling salesman problem (TSP) is often used, because it is one of the most standard combinatorial optimization problems. For an N-city symmetric TSP, the number of all possible tours is (N − 1)!. Thus, the number of tours exponentially diverges if the number of cities increases. It is widely acknowledged that the TSP belongs to a class of NP-hard. It means that it is almost impossible to obtain optimal solutions. Therefore, it is required to develop an effective approximate algorithm for finding near-optimal solutions in a reasonable time frame.

To discover approximate solutions, various local search algorithms have already been proposed, for example, the k-opt algorithm (k = 1, 2, . . . , N), the Or-opt algorithm [1], the Lin-Kernighan (LK) algorithm [2], and the stem-and-cycle (S&C) ejection chain method [3, 4]. However, it is almost impossible to find optimal solutions only by the local search algorithms because of a local minimum problem. During the search, the state gets stuck at local minima.

To resolve the local minimum problem, various metaheuristic strategies have been proposed such as simulated annealing [5], genetic algorithm [6], tabu search [7], chaotic search [8], and so on. Among them, the chaotic search [8–12] shows good performance. In the chaotic search, to avoid local minima, execution of a local search algorithm is controlled by chaotic dynamics. In Refs. [8–12], to generate the chaotic dynamics, a chaotic neural network [13] is used. In the chaotic neural network, the basic element is a chaotic neuron proposed by Aihara et al. [13]. It can reproduce refractoriness, one of the important properties which real nerve cells have: When a neuron has just fired, the firing of this neuron is inhibited for a while by the refractoriness. In Refs. [8–12], execution of the local search algorithm is encoded by firings of the chaotic neuron. If the chaotic neuron fires, the corresponding local search algorithm is executed. Because the firing of the chaotic neuron is inhibited by the refractoriness, frequent firings of the chaotic neuron, or frequent execution of the local search is restricted. Thus, the chaotic search can escape from local minima efficiently.

We have already proposed two chaotic search methods [11, 14, 15]. In the first method, execution of the LK algorithm [2] is controlled by the chaotic dynamics. The LK algorithm [2] is one of the most famous variable depth search methods. As a result, the chaotic search method using the LK algorithm shows solving performance with less than 0.7% gaps from the optimal solution for instances with the order of 10^4 cities and can be applied to large scale instances with the order of 10^5 cities [11].

On the other hand, the S&C ejection chain method [3, 4] is also one of the most effective variable depth search methods. It is reported that the S&C ejection chain method leads to better solutions than the LK algorithm [4]. One of the reasons is that the S&C ejection chain method can explore more diversified solution space, because it introduces an...
S&C structure, which is not a tour. Namely, the S&C ejection chain method can explore unfeasible solution space. However, the S&C ejection chain method also gets stuck at local minima because it is also a greedy algorithm.

Therefore, to resolve the local minimum problem in the S&C ejection chain method, we proposed chaotic search method which controls the S&C ejection chain method by the chaotic dynamics. Although the introduction of the S&C ejection chain method leads to higher performance, it is not so easy to apply the chaotic search method using the S&C ejection chain method to large scale instances, because this method takes much calculation time to find good results. From the viewpoint of application of approximate algorithms to real life problems, it is also important to develop an effective algorithm which finds good near-optimal solutions with less calculation time.

Although there are several strategies of how to reduce the calculation cost, we introduce the following strategy: we combine two chaotic search methods, namely the LK-algorithm-base chaotic search and the S&C-ejection-chain-method-base chaotic search. In the proposed method, both the LK algorithm and S&C ejection chain method are controlled by the chaotic dynamics.

Numerical experiments show that by combining the LK algorithm with the S&C ejection chain method, although calculation costs of the proposed method are less than the chaotic search method using the S&C ejection chain method, its solving performance is improved than the conventional method. The proposed method has high solving ability for very large scale instances such as $10^5$ order.

2. Chaotic search method using both the LK algorithm and S&C ejection chain method

The Lin-Kernighan (LK) algorithm [2] which is one of the most effective local search algorithms for solving the traveling salesman problem (TSP). On the other hand, the stem-and-cycle (S&C) ejection chain method [3, 4] is also one of the most effective local search algorithms, and it shows higher performance than the LK algorithm. Thus, if we apply the chaotic dynamics to the S&C ejection chain method, we can expect that the performance of the chaotic search method is much improved [15]. However, if we only introduce the chaotic search method using the S&C ejection chain method, it takes large amount of computational time to obtain high performance for solving the large scale TSPs.

To find good solutions with less calculation time, in this paper, we introduced a strategy of combination of two chaotic searches: the LK algorithm controlled by the chaotic dynamics [11, 14] and the S&C ejection chain method controlled by the chaotic dynamics [15].

To escape from local minima and to explore better solutions, the chaotic dynamics is introduced in both methods. To realize the chaotic dynamics, we use a chaotic neuron model proposed by Aihara et al. [13]. The chaotic neuron model can reproduce the refractoriness which is one of the important properties of real nerve cells. When a neuron has just fired, the firing of this neuron is inhibited for a while. By controlling executions of the LK algorithm (or S&C ejection chain method) by the firings of chaotic neuron, the chaotic search method can escape from local minima efficiently, because the same improvements are restricted for a while.

In the proposed method, two local search algorithms, the LK algorithm and S&C ejection chain method, are controled by chaotic dynamics. Namely, the LK algorithm controlled by the chaotic dynamics and the S&C ejection chain method controlled by the chaotic dynamics are stochastically selected to execute. To realize this strategy, we introduced a combination rate $p$. When a chaotic neuron is updated, either the LK algorithm or the S&C ejection chain method is selected with probability $1 - p$, and the S&C ejection chain method is selected with probability $p$.

If $p = 0$, the proposed method is the same as the chaotic search method using the LK algorithm [11, 14]. If $p = 1$, the proposed method is the same as the chaotic search method using the S&C ejection chain method [15]. Then, the gain effect is described by the following equations:

\[ \xi_i(t + 1) = \max_j [\beta(t) \Delta_{ij}(t) + \xi_j(t)] \]  
\[ \Delta_{ij}(t) = \begin{cases} \Delta_{ij}^{LK}(t) & \text{with probability } 1 - p \\ \Delta_{ij}^{SC}(t) & \text{with probability } p. \end{cases} \]
\[ \beta(t + 1) = \beta(t) + \frac{q}{1 - p} \sum_{i=1}^{N} |\Delta_{ij}(t)| \]

where $\beta(t)$ is a scaling parameter of the gain effect at time $t$; $\Delta_{ij}^{LK}(t)$ is a gain of the LK algorithm which connects cities $i$ and $j$ (Fig. 1(a)), namely $\Delta_{ij}^{LK}(t) = D_{ij}(t) - D_{ij}^{LK}(t)$, where $D_{ij}(t)$ is a length of a current tour at time $t$ and $D_{ij}^{LK}(t)$ is a length of a new tour if cities $i$ and $j$ are connected by the LK algorithm at time $t$. $\Delta_{ij}^{SC}(t)$ is a gain of the S&C ejection chain method which connects cities $i$ and $j$ (Fig. 1(b)).

Next, the refractory effect is described as follows:

\[ \zeta_i(t + 1) = -\alpha \sum_{d=0}^{r} k^p_r x_i(t - d) + \theta, \]

where $\alpha$ is a scaling parameter of the refractory effect; $k_r$ is a decay parameter; $x_i(t)$ is an output of the $i$th chaotic neuron at time $t$; and $\theta$ is a threshold value. If a neuron has fired in the past, Eq. (4) becomes negative. Therefore, the refractory effect inhibits the firing of the neuron in response to the past firing history.

Finally, the output of the $i$th chaotic neuron at time $t + 1$ is described as follows:

\[ x_i(t + 1) = f(\xi_i(t + 1) + \zeta_i(t + 1)), \]
where \( f(y) = \frac{1}{1 + e^{-y/\alpha}} \). If \( x_i(t) \geq 1/2 \), the \( i \)th chaotic neuron fires at time \( t \), and the LK algorithm (or S&C ejection chain method) which connects cities \( i \) and \( j \) is executed. All neurons are updated asynchronously and randomly. A single iteration is defined to be updates of all neurons.

If \( x_i(t) = 1/2 \), the \( i \)th chaotic neuron fires at time \( t \), and the LK algorithm (or S&C ejection chain method) which connects cities \( i \) and \( j \) is executed. All neurons are updated asynchronously and randomly. A single iteration is defined to be updates of all neurons.

Table 1 shows the results of the proposed methods for the instances with more than 10^5 cities of TSPLIB. We fix the number of trials to 10 for all instances. From the results, the proposed chaotic search methods obtain solutions within less than 1% for all instances. Although the CS-SC and CS-LKSC show higher performance than the CS-LK, the CS-SC takes more running time, particularly for large scale instances. On the other hand, the CS-LKSC can obtain better solutions in shorter time than the CS-SC for all instances. Although running time of the CS-LKSC is almost the same as that of the CS-LK for instances with the order of 10^3 cities, for larger instances, the CS-LKSC takes much calculation time than the CS-LK. Thus, if we apply the CS-LKSC to larger instances, we need to adjust the combination rate \( p \) to a smaller value.

4. Conclusions

In this paper, we proposed a novel chaotic search method which combines the chaotic search method using the Lin-Kernighan algorithm with that using the stem-and-cycle ejection chain method. We used large scale instances from TSPLIB. From the results, it is revealed that the proposed method has higher performance than the conventional chaotic search method. Then, performance of the proposed method does not depend on the size of the instance.

Acknowledgments

The authors wish to thank K. Aihara, Y. Horio, M. Adachi, and M. Hasegawa for their fruitful comments and discussions. The research of T.I. is partially supported by Grant-in-Aid for Scientific Research (B) (No.20300085) from the JSPS.

References


Table 1: Average gaps of the chaotic search methods. CS-LK stands for the chaotic search method using the LK algorithm [11, 14], CS-SC stands for the chaotic search method using the S&C ejection chain method [15] and CS-LKSC stands for the chaotic search method using the LK algorithm and S&C ejection chain method. Bold faces represent the best result.

<table>
<thead>
<tr>
<th>Instance</th>
<th>Type</th>
<th>Gap [%]</th>
<th>Time [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>CS-LK</td>
<td>CS-SC</td>
</tr>
<tr>
<td>brd14051</td>
<td>EUC 2D</td>
<td>0.775</td>
<td>0.726</td>
</tr>
<tr>
<td>d15112</td>
<td>EUC 2D</td>
<td>0.701</td>
<td>0.669</td>
</tr>
<tr>
<td>d18512</td>
<td>EUC 2D</td>
<td>0.770</td>
<td>0.708</td>
</tr>
<tr>
<td>pla33810</td>
<td>CEIL 2D</td>
<td>0.672</td>
<td>0.682</td>
</tr>
<tr>
<td>pla85900</td>
<td>CEIL 2D</td>
<td>0.557</td>
<td>0.580</td>
</tr>
<tr>
<td>rl11849</td>
<td>EUC 2D</td>
<td>0.717</td>
<td>0.657</td>
</tr>
<tr>
<td>usa13509</td>
<td>EUC 2D</td>
<td>0.668</td>
<td>0.621</td>
</tr>
</tbody>
</table>


[16] TSPLIB. available: [http://www.iwr.ub- heidelberg.de/groups/comopt/software/TSPLIB95/](http://www.iwr.ub-heidelberg.de/groups/comopt/software/TSPLIB95/).
A Method for Solving Asymmetric Traveling Salesman Problems
Using Chaotic Exchange Method and Block Shift Operations

Toshihiro TACHIBANA†† and Masaharu ADACHI†∗

†Graduate School of Advanced Science and Technology, Tokyo Denki University
2-2 Kanda–Nishiki–cho, Chiyoda–ku, Tokyo 101–8457, Japan
Email: * 09ude01@ms.dendai.ac.jp, ** adachi@eee.dendai.ac.jp

Abstract—An asymmetric Traveling Salesman Problem is one in which the costs for travel between one city and another are not symmetric. We propose a method for solving such problems based on chaotic search method by Hasegawa et al. It uses a chaotic neural network and tabu search, namely, block shift operations and 2-opt exchange are used as exchanging methods. In block shift operations, we consider several cities as a BLOCK, and the whole BLOCK is exchanged with other city. The proposed method switches two types of exchanging method by using chaotic neurodynamics. The proposed method exhibited better solutions than the method with a fixed ratio of the executing the exchanging methods.

1. Introduction

The Traveling Salesman Problem (TSP) is a famous NP-hard problem. It is a combinatorial optimization problem which appears quite frequently in various fields such as transit and control [1]. For example, vehicle routing, designing of VLSI, boring of circuit boards, X-ray crystal structural analysis, and so on. The TSP suffers from combinatorial explosion: the number of feasible solutions increases as \(n!\) as the number of cities \(n\) increases[1, 2]. In other words, if there are a large number of cities, it is difficult to obtain a good solution to this problem using simple techniques such as the round robin method[2].

Symmetric TSPs have been studied by many researchers. However, in everyday problems of this type, it is rare that the costs are symmetric. Therefore, there are demands for dealing with asymmetric cost problems [3]; however, thus far, asymmetric TSPs have not been extensively studied. In particular, most of the conventional method is applied either to symmetric TSP or asymmetric TSP. Any method that is efficient to both problems does not exist.

Branch and bound method is known as method for solving the asymmetric TSP. However, the branch and bound method requires huge computational costs. It is not practical to solve the asymmetric problem that takes too long computational time. Therefore, in this paper, we propose a method for finding quasi-optimal solution by a metaheuristics approach.

This paper is organized as follows. Section 2 introduces the asymmetric TSP. The chaotic search methods proposed by Hasegawa et al., on which our proposed method is based, is described in Section 3. In Section 4, we describe the proposed method which switches block shift operations and point exchange by using chaotic neurodynamics. In Section 5, results of numerical experiments are shown. Lastly, we present conclusions.

2. Asymmetric traveling salesman problems

The TSP is an optimization problem whose goal is to determine the minimum-cost tour visiting \(n\) cities, under the conditions that one visits every cities only at once [3]. For an \(n\)-city problem, a set of the cities is

\[
V = \{v_1, v_2, \cdots, v_n\}. \tag{1}
\]

A tour \(\sigma\) specifies the order in which these cities are visited. The total cost \(f(\sigma)\) of a tour \(\sigma\) can be evaluated from the costs for travel from city \(j\) to city \(i\), \(d_{ij}(v_i, v_j \in V)\).

\[
f_{TSP}(\sigma) = \sum_{k=1}^{n-1} d_{\sigma(k), \sigma(k+1)} + d_{\sigma(n), \sigma(1)} \tag{2}
\]

The goal of the TSP is to minimize the cost in Eq. (2). The nomenclature for TSPs used in this paper is summarized in Table 1.

<table>
<thead>
<tr>
<th>Table 1: Nomenclature for TSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>(d_{ij}) : Cost to travel from city (i) to city (j)</td>
</tr>
<tr>
<td>(f_{TSP} (\sigma)) : Total cost of a tour</td>
</tr>
<tr>
<td>(\sigma) : The order of cities visited in a tour</td>
</tr>
<tr>
<td>(V) : Set of cities to visit</td>
</tr>
<tr>
<td>(n) : Number of the cities</td>
</tr>
<tr>
<td>(u_{im}) : Internal state of the neuron (im)</td>
</tr>
<tr>
<td>(\theta_{im}) : Threshold of the neuron (im)</td>
</tr>
<tr>
<td>(x_{im}) : Output of the neuron (im)</td>
</tr>
<tr>
<td>(w_{im,jn}) : Synaptic weight from the neuron (jn) to the neuron (im)</td>
</tr>
</tbody>
</table>

A TSP is asymmetric when the cost of travel between two given cities is not symmetric, that is, \(d_{ij} \neq d_{ji}\). In this paper, we do not include cases in which two cities are connected only in one direction.
3. Chaotic search method

The proposed method is based on the chaotic search method [4] which uses a chaotic neural network with tabu search [5]. A feature of this method is that the number of neurons used is equal to the number of cities. We give an outline of the chaotic search method in this section.

3.1. Coding of networks

The updating of the internal state of each neuron is executed by the following set of equations.

\[ f(x) = \frac{1}{1 + e^{-\gamma x}} \]  
\[ \xi_i(t+1) = \max_j \{ \zeta_j(t+1) + \beta \Delta_{ij}(t) \} \]  
\[ \eta_i(t+1) = -W \sum_{k=1}^{N} x_k(t) + W \]  
\[ \zeta_i(t+1) = -\alpha \sum_{d=0}^{t} k_r^d x_i(t-d) + \theta \]  
\[ x_i(t+1) = f(\xi_i(t+1) + \eta_i(t+1) + \zeta_i(t+1)) \]

If \( x_i(t+1) > \frac{1}{2} \), city \( i \) is connected to city \( j \) that gives the maximum in Eq. (4) by using 2-opt exchanges. The nomenclature for neural networks used in this paper is summarized in Table 2.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( k_r )</td>
<td>Decay parameter of the gain effect</td>
</tr>
<tr>
<td>( \alpha )</td>
<td>Scaling parameter of the tabu effect</td>
</tr>
<tr>
<td>( \beta )</td>
<td>Scaling parameter of the gain effect</td>
</tr>
<tr>
<td>( \Delta_{ij} )</td>
<td>Gain of the objective function value offered by the 2-opt exchange which links cities ( i ) and ( j ).</td>
</tr>
<tr>
<td>( \theta, R )</td>
<td>Positive biases</td>
</tr>
</tbody>
</table>

3.2. 2-opt exchange

Two-opt exchange is a basic city exchange method for symmetric TSPs in which any two cities exchange their position in the tour. For example, a tour shown in Fig. 1 (a), can be changed to Fig. 1 (b) by cutting two links and creating two new links.

4. Proposed method

4.1. Overview

We propose a method that is a modification of the chaotic search method described in section 3. Although the proposed method uses the same network and the same algorithm, it differs through the use of two exchange methods switches by chaotic neurodynamics. The two exchange methods are block shift operations and 2-opt exchange in the paper. The block shift operations are described in the next section 4.2. The proposed method for switching between a 2-opt method and block shift operation using chaotic neurodynamics is described in section 4.4.

4.2. Block shift operations

The 2-opt exchange is a fundamental city exchange method for solving TSPs. When a 2-opt exchange is executed, the direction of a part of the tour is reversed by the city exchange. In symmetric TSPs, this reversal does not alter the cost of the tour. However, in an asymmetric TSP, the 2-opt exchange may cause the cost to rise. To overcome this disadvantage of the 2-opt exchange in asymmetric TSPs, the block shift operation has been proposed [6]. This method considers several cities as a BLOCK, and the whole BLOCK is exchanged with another city. During the exchange, the order of the cities in the BLOCK is maintained. We show an example of a block shift operation in Fig. 2. Although a 2-opt exchange could be viewed as a type of block shift operation, we use the term “block shift operation” in this paper only when a block consists of more than one city.
4.3. Bidirectional search

In asymmetric TSPs, the cost of a tour in a direction is different from that of the opposite direction. In symmetric TSPs the cost of a clockwise tour is equal to that of counterclockwise one. For example, in Fig. 3 (a), the cost of the tour of clockwise a-b-c-d-e-g-f-g-h and that of counterclockwise h-g-f-g-e-d-c-b-a are the same (they are “58”). Whereas in asymmetric TSPs, the cost of clockwise tour is not equal to that of counterclockwise one. For example, in Fig. 3 (b), the cost of the clockwise tour a-b-c-d-e-g-f-g-h is equal to “52”, whereas the cost of counterclockwise tour h-g-f-g-e-d-c-b-a is equal to “50”. As shown in the above example, tours generated by the above mentioned operations must be evaluated for both directions.

![Symmetric TSP and Asymmetric TSP](image)

Figure 3: Symmetric and asymmetric TSP by comparing the difference between the cost of visiting direction.

4.4. How to select exchange methods using chaotic neurodynamics

In the proposed method, a switching between two exchange methods are determined by chaotic neurodynamics. It is based on a chaotic search method. The proposed method uses a chaotic neuron in the similar way as it is used in the chaotic search method[4]. The nomenclature for the neuron used in this paper is summarized in Table 5. The updating of the internal state of the neuron is executed by using the following set of equations.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum cost</td>
<td>5620</td>
<td>5620</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Maximum cost</td>
<td>5622</td>
<td>5624</td>
<td>5641</td>
<td>N/A</td>
</tr>
<tr>
<td>Average cost</td>
<td>5620.4</td>
<td>5622.0</td>
<td>5632.3</td>
<td>N/A</td>
</tr>
<tr>
<td>Gap [%]</td>
<td>0.007</td>
<td>0.04</td>
<td>0.2</td>
<td>0.01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum cost</td>
<td>14713</td>
<td>15083</td>
<td>15358</td>
<td>N/A</td>
</tr>
<tr>
<td>Maximum cost</td>
<td>15204</td>
<td>15386</td>
<td>17024</td>
<td>N/A</td>
</tr>
<tr>
<td>Average cost</td>
<td>15083</td>
<td>15254.4</td>
<td>16146.6</td>
<td>N/A</td>
</tr>
<tr>
<td>Gap [%]</td>
<td>3.7</td>
<td>5.8</td>
<td>12.0</td>
<td>1.92</td>
</tr>
</tbody>
</table>

Table 3: Experimental results of proposed and conventional methods for p44.

Table 4: Experimental results of proposed and conventional methods for ry48p.

Table 5: Nomenclature for neuron for selecting exchange methods

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_r$</td>
<td>Decay parameter of the gain effect</td>
</tr>
<tr>
<td>$\alpha_{ex}$</td>
<td>Scaling parameter of the tabu effect</td>
</tr>
<tr>
<td>$\beta$</td>
<td>Scaling parameter of the gain effect</td>
</tr>
<tr>
<td>$\Delta_{ex}(t)$</td>
<td>Gain of the objective function value difference between after the candidate and that of current tour.</td>
</tr>
<tr>
<td>$A, \theta$</td>
<td>Positive biases</td>
</tr>
</tbody>
</table>

\[
\xi_{ex}(t+1) = \beta \Delta_{ex}(t) + A
\]

\[
\zeta_{ex}(t+1) = -\alpha_{ex} \sum_{d=0}^{t} k_r^d x_{ex}(t-d) + \theta
\]

\[
x_{ex}(t+1) = f(\zeta_{ex}(t+1) + \zeta_{ex}(t+1))
\]

Where, the sigmoid function of Eq. (3) is used as $f$. If $x_{ex}(t) > \frac{1}{2}$, the proposed method switches to another exchange method from the current exchange method. In other words, when the current exchange method can reduce costs, the method is continued to use, otherwise, it is switched to the other exchange method.
5. Results of numerical experiments

We numerically evaluated the performance of the proposed method for some benchmark problems of TSPLIB [7], performing 20 trials for each problem. We used asymmetric and symmetric problems. As asymmetric problems, we used ‘br17’, (17-city), ‘p43’, (43-city), ‘ry48p’, (48-city), and ‘rgb443’, (443-city) from TSPLIB. Moreover, as a symmetric problem ‘kroA100’ of 100-city was used for the evaluation. We show results of the performance of the proposed method for ‘p43’, ‘ry48p’, and ‘rgb443’ and ‘kroA100’ in Table 3, 4, 6 and 7, respectively. In these tables, the performances of conventional methods are also shown. Results in these tables shown that the proposed method with dynamic switching between block shift operations and 2-opt exchanges achieves the exact solution more frequently than the conventional methods.

6. Conclusions

In this paper, we have proposed a method for solving an asymmetric TSP using chaotic neural dynamics. The proposed method with dynamic switching between block shift and spot exchange gives better solutions than the chaotic search method [4], although it is not superior to the existing method of using a hybrid genetic algorithm[9].

A future problem is to investigate a method for selecting efficient parameters automatically.

Acknowledgments

This work was supported by KAKENHI (20300085).

Table 6: Experimental results of proposed and conventional methods for rgb443.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum cost</td>
<td>2720</td>
<td>2720</td>
<td>3739</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Maximum cost</td>
<td>2723</td>
<td>2730</td>
<td>3841</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Average cost</td>
<td>2720.4</td>
<td>2722.1</td>
<td>3793.0</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Gap [%]</td>
<td>0.04</td>
<td>0.09</td>
<td>1.16</td>
<td>0.09</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 7: Experimental results of proposed and conventional methods for kroA100.

<table>
<thead>
<tr>
<th>Method</th>
<th>Proposed</th>
<th>Proposed (fixed ratio)</th>
<th>Chaotic Search[4]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum cost</td>
<td>21369</td>
<td>22373</td>
<td>21282</td>
</tr>
<tr>
<td>Maximum cost</td>
<td>22288</td>
<td>22852</td>
<td>22134</td>
</tr>
<tr>
<td>Average cost</td>
<td>21754.8</td>
<td>22690.6</td>
<td>21440.8</td>
</tr>
<tr>
<td>Gap [%]</td>
<td>2.2</td>
<td>6.6</td>
<td>0.7</td>
</tr>
</tbody>
</table>

References

[7] TSPLIB (http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/)
Double-Assignment Method Driven by Chaotic Neurodynamics for Quadratic Assignment Problems

Kazuaki Shibata† and Yoshihiko Horio†

†Graduate School of Engineering, Tokyo Denki University
Chiyoda-ku, Tokyo, 101-8457, Japan
Email: 09kme26@ms.dendai.ac.jp

Abstract—A quadratic assignment problem (QAP) is one of the NP-hard combinatorial optimization problems. Local search methods such as the 2-opt method are used to solve the QAPs. However, they are usually trapped in the local minima. In order to solve this problem, heuristic methods such as the tabu search have been proposed. In particular, the exponential chaotic tabu search shows excellent performance. On the other hand, we proposed a double-assignment method which searches sub-optimal solutions through a solution space including infeasible solutions. We have shown that the proposed method is superior to the 2-opt algorithm in solving the QAPs. In the double-assignment method, an infeasible solution is first constructed from the initial solution by assigning two elements to one index. Then, a feasible solution is composed from the infeasible solution.

In this paper, we improve the double-assignment method for the QAPs by introducing chaotic neurodynamics. We show numerical simulation results comparing the performance of the improved method and the original double-assignment method.

1. Introduction

A quadratic assignment problem (QAP) [1] is one of the NP-hard combinatorial optimization problems. Local search methods such as the 2-opt method are used to solve the QAPs. However, they are usually trapped in the local minima. In order to solve this problem, heuristic methods such as the tabu search have been proposed [2]. In particular, the exponential chaotic tabu search [3] shows excellent performance. In the exponential chaotic tabu search, the chaotic neurodynamics from a chaotic neural network drive the 2-opt algorithm. Our final goal is to apply the Lin-Kernighan algorithm [4], which is an effective local search method for traveling salesman problems (TSPs) [5], to the exponential chaotic tabu search for the QAPs instead of the 2-opt algorithm. To this end, we have proposed a double-assignment method based on the Lin-Kernighan algorithm [6]. In addition, we have shown that the double-assignment method has higher performance than the 2-opt method in solving the QAPs [6]. On the other hand, the Lin-Kernighan algorithm was driven by the chaotic dynamics to have good performance in solving the TSPs [7].
3. Double-Assignment Method

The double-assignment method [6] searches sub-optimal solutions through a solution space including infeasible solutions. First, this method constructs an infeasible solution by assigning two plants to one city. Then, we reconstruct a feasible solution by reassigning one of the two plants to an empty city. The procedure of the double-assignment method is as follows.

**Step 0**: Let $m = 0$ and $n = 0$, where $m$ is the total number of runs in Step 1 and Step 2, and $n$ is the number of runs in Step 3.

**Step 1**: This step is shown in Fig. 1. We first choose a plant $i$ at random. Then, we choose a city $j$ from the cities to which the plant $i$ is not assigned. Next, we assign the plant $i$ to the city $j$. After that, we make $m = 1$. Moreover, the city to which the city $i$ was originally assigned is denoted as $y_i$.

**Step 2**: We increment $m$ by 1, that is, $m = m + 1$. We choose a city $y_m$ among the cities which have only one plant. We choose the city $x_m$ that gives the minimum increase in the cost from the cities excluding the cities $x_1$ to $x_{m-1}$ and $y_1$ to $y_m$. Then, we reassign the plant in the city $y_m$ to the city $x_m$. If $m < N/2$, we repeat **Step 2**, otherwise we proceed to **Step 3**. **Step 2** is illustrated in Fig. 2.

**Step 3**: We increment $n$ by 1 as $n = n + 1$. We choose a plant from the plants assigned to the cities $x_1$ to $x_m$, which gives the minimum increase in the cost when it is assigned to the city $y_n$. Then, we assign the chosen plant to the city $y_n$. Moreover, we denote the city where the chosen city resided as $z_n$. If $n = m$, we complete the algorithm, otherwise we repeat **Step 3**.

Changes of the cost in **Step 1** and **Step 2** can be written by Eqs. (5) and (6), respectively. In Eqs. (5) and (6), $\Delta F_{ij}^1$ is the increase of the cost when the plant in the city $i$ is assigned to the city $j$ in **Step 1**, and $\Delta F_{ij}^2$ is that in **Step 2**, respectively. In these equations, $p(i)$ and $p(j)$ are the plants which are assigned to the city $i$ and city $j$, respectively. Moreover, $p_1(i)$ is the plant assigned to the city $i$ at the beginning, and $p_2(i)$ is the plant newly assigned to the city $i$.

\[
\Delta F_{ij}^1 = -\left(a_{ij}b_{p(i)p(j)} + a_{ji}b_{p(j)p(i)}\right)
+ \sum_{k=1, k\neq i, j}^{N} \left(b_{p(i)p(k)}\right) \left(a_{kj} - a_{kj}\right)
+ \sum_{k=1, k\neq i, j}^{N} \left(b_{p(j)p(i)}\right) \left(a_{jk} - a_{jk}\right) \tag{5}
\]

\[
\Delta F_{ij}^2 = -\left(a_{ij}b_{p(i)p(j)} + a_{ji}b_{p(j)p(i)}\right)
+ \sum_{k=1, k\neq i, j}^{N} \left(b_{p(i)p(k)}\right) \left(a_{kj} - a_{kj}\right)
+ \sum_{k=1, k\neq i, j}^{N} \left(b_{p(j)p(i)}\right) \left(a_{jk} - a_{jk}\right) \tag{6}
\]

Furthermore, the increase of the cost in **Step 3** is given in Eq. (7). In Eq. (7), $\Delta F_{ij}^3$ represents the change in the cost when the plant $i$ is assigned to the city $j$ in **Step 3**. Moreover, $l$ is the city to which the plant $i$ was originally assigned. Moreover, $p(j)$ and $p(l)$ are the plants assigned to the city $j$ and the city $l$, respectively.

\[
\Delta F_{ij}^3 = a_{ij}b_{p(i)} + a_{ji}b_{p(j)}
+ \sum_{k=1, k\neq i, j}^{N} \left(b_{p(i)k}\right) \left(a_{kj} - a_{kj}\right)
+ \sum_{k=1, k\neq i, j}^{N} \left(b_{p(j)k}\right) \left(a_{jk} - a_{jk}\right) \tag{7}
\]
Step 2 of the proposed method (size-7 QAP, \( m = 1 \)). If the city II is randomly chosen, we calculate the increases in the cost when the plant 3, which is assigned to the city II, is temporarily assigned to the cities III, IV, V and VII, respectively. If the increase in the cost is the lowest when the plant 3 is temporarily assigned to the city VII, we eventually assign the plant 3 to the city VII. We do not assign the plant 4 to the city II which is already assigned to city VII, as in Step 1.

4. Double-Assignment Method Driven by Chaotic Neurodynamics

In this section, we propose the double-assignment method driven by the chaotic neurodynamics. We use a chaotic neural network constructed by the chaotic neurons given by

\[
\begin{align*}
\zeta_{ij}(t+1) &= k_r \zeta_{ij}(t) - ax_{ij}(t) + R \quad (8) \\
\xi_{ij}(t+1) &= \beta(F^p_{ij}(t) - F^f_{ij}(t)) \quad (9) \\
x_{ij}(t+1) &= f(\zeta_{ij}(t+1) + \xi_{ij}(t+1)) \quad (10)
\end{align*}
\]

where, \( \zeta_{ij}(t+1) \) is the refractory effect, \( \xi_{ij}(t+1) \) is the gain effect, \( x_{ij}(t+1) \) is the output of the \((i, j)\)th neuron, \( \beta \) is a scaling parameter of the gain effect, \( k_r \) is a decay parameter of the refractory effect \((0 < k_r < 1)\), \( \alpha \) is a scaling parameter of the refractory effect \((\alpha > 0)\), \( R \) is an external bias, and \( f(y) \) is a sigmoidal output function of the neuron \((f(y) = 1/(1 + e^{-3/5y}))\). \( F^p_{ij}(t) \) represents the cost at time \( t \), and \( F^f_{ij}(t) \) is the cost of the assignment of the plant \( i \) to the city \( j \) through the double-assignment method.

In the proposed method, we asynchronously update the neural network as follows. To update the \((i, j)\)th neuron, \( \zeta_{ij}(t+1) \) and \( \xi_{ij}(t+1) \) are calculated. To obtain \( \zeta_{ij}(t+1) \), the double-assignment method which assigns the plant \( i \) to the city \( j \) is executed. This temporal solution is denoted as \( P' \). Then, if \( x_{ij}(t+1) > \frac{1}{2} \), we update the current solution to \( P'' \). Here, \( F^p_{ij}(t) - F^f_{ij}(t) \) is normalized by \( a_M b_M \), where \( a_M = \max(a_{ij}) \) and \( b_M = \max(b_{ij}) \). The order of neuron updates are \((1, 1) \rightarrow (1, 2) \rightarrow \cdots \rightarrow (1, N) \rightarrow (2, 1) \rightarrow (2, 2) \rightarrow \cdots \rightarrow (2, N) \rightarrow \cdots \rightarrow (N, N-1) \rightarrow (N, N) \), for the size \( N \) problem. When the \((N, N)\)th neuron was updated, one iteration is completed.

5. Simulation Results

We compare the performance of the proposed method with the chaotic neurodynamics and the original double-assignment method through numerical simulations. In the original double-assignment method without chaotic dynamics, we first try all the combinations of the indices and elements for assignments. Then, we finally update the cur-
rent solution using the combination that gives the minimum cost. We define the above one update as one iteration with the original double-assignment method. Moreover, we use 5,000 iterations for one trial, and 30 trials are done for both methods.

Tables 1 and 2 show the average gaps from the optimal solutions, and the frequency of the optimum solution (OS), respectively. Table 3 shows the parameters for the chaotic neural network used in the proposed method.

From Table 1, the proposed method with the chaotic dynamics obtains better solutions for size-20 problems than the original double-assignment method except for Tai20b. However, Table 2 shows the original method is better than the proposed method for size-30 problems except for Lipa30a and Lipa30b.

6. Conclusions

We have proposed the double-assignment method driven by the chaotic neurodynamics. The proposed method was shown to have better performance than the conventional method for the QAPs of size of 20. However, the improvements have not been obtained for size-30 QAPs except for Lipa30a and Lipa30b. Therefore, we will decipher the reason for the above results, and further improve the proposed method.

Table 1: The average gap from the optimum solution, and the number of the optimum solution, (OS), obtained during 30 trials for the size-20 QAPs

<table>
<thead>
<tr>
<th>Instance</th>
<th>Original Method</th>
<th>Proposed Method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Average Gap</td>
<td>Average Gap</td>
</tr>
<tr>
<td>Nug20</td>
<td>1.08949</td>
<td>0.21530</td>
</tr>
<tr>
<td>Tai20a</td>
<td>1.75716</td>
<td>0.93555</td>
</tr>
<tr>
<td>Tai20b</td>
<td>0.32451</td>
<td>0.63842</td>
</tr>
<tr>
<td>Had20</td>
<td>0.49889</td>
<td>0.27930</td>
</tr>
<tr>
<td>Rou20</td>
<td>1.42614</td>
<td>0.90682</td>
</tr>
<tr>
<td>Scr20</td>
<td>8.68429</td>
<td>1.60926</td>
</tr>
<tr>
<td>Chr20a</td>
<td>8.75000</td>
<td>1.17701</td>
</tr>
<tr>
<td>Chr20b</td>
<td>6.54482</td>
<td>2.18741</td>
</tr>
<tr>
<td>Chr20c</td>
<td>9.26319</td>
<td>0.15981</td>
</tr>
<tr>
<td>Lipa20a</td>
<td>1.01819</td>
<td>0.00000</td>
</tr>
<tr>
<td>Lipa20b</td>
<td>0.87076</td>
<td>0.00000</td>
</tr>
</tbody>
</table>

Table 2: The average gap from the optimum solution, and the number of the optimum solution, (OS), obtained during 30 trials for the size-30 QAPs

<table>
<thead>
<tr>
<th>Instance</th>
<th>Original Method</th>
<th>Proposed Method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Average Gap</td>
<td>Average Gap</td>
</tr>
<tr>
<td>Nug30</td>
<td>1.59482</td>
<td>1.68408</td>
</tr>
<tr>
<td>Tai30a</td>
<td>2.62372</td>
<td>3.41508</td>
</tr>
<tr>
<td>Tai30b</td>
<td>0.30766</td>
<td>1.30766</td>
</tr>
<tr>
<td>Lipa30a</td>
<td>1.11575</td>
<td>0.98750</td>
</tr>
<tr>
<td>Lipa30b</td>
<td>5.36006</td>
<td>1.83971</td>
</tr>
</tbody>
</table>

Table 3: Parameter values for the chaotic neuron.

<table>
<thead>
<tr>
<th>Instance</th>
<th>$\alpha$</th>
<th>$\beta$</th>
<th>$k$</th>
<th>$R$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nug20</td>
<td>0.7</td>
<td>0.7</td>
<td>0.4</td>
<td>0.1</td>
</tr>
<tr>
<td>Tai20a</td>
<td>1.0</td>
<td>0.7</td>
<td>0.7</td>
<td>0.1</td>
</tr>
<tr>
<td>Tai20b</td>
<td>1.0</td>
<td>1.0</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>Had20</td>
<td>0.1</td>
<td>0.7</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>Rou20</td>
<td>0.1</td>
<td>1.0</td>
<td>0.7</td>
<td>0.1</td>
</tr>
<tr>
<td>Scr20</td>
<td>1.0</td>
<td>1.0</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>Chr20a</td>
<td>1.0</td>
<td>0.7</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>Chr20b</td>
<td>1.0</td>
<td>0.7</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>Chr20c</td>
<td>1.0</td>
<td>0.7</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>Lipa20a</td>
<td>1.0</td>
<td>1.0</td>
<td>0.7</td>
<td>0.1</td>
</tr>
<tr>
<td>Lipa20b</td>
<td>1.0</td>
<td>1.0</td>
<td>0.7</td>
<td>0.1</td>
</tr>
<tr>
<td>Nug30</td>
<td>0.7</td>
<td>0.7</td>
<td>0.4</td>
<td>0.1</td>
</tr>
<tr>
<td>Tai30a</td>
<td>1.0</td>
<td>0.7</td>
<td>0.7</td>
<td>0.1</td>
</tr>
<tr>
<td>Tai30b</td>
<td>1.0</td>
<td>1.0</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>Lipa30a</td>
<td>1.0</td>
<td>1.0</td>
<td>0.7</td>
<td>0.1</td>
</tr>
<tr>
<td>Lipa30b</td>
<td>1.0</td>
<td>1.0</td>
<td>0.7</td>
<td>0.1</td>
</tr>
</tbody>
</table>

Acknowledgement

This work was supported by Kakenhi (20300085).

References


Spectral Analysis and Dynamical Behavior of Complex Networks

Kohshi Okumura and Ljiljana Trajković
Simon Fraser University
Vancouver, British Columbia, Canada
Email: o.kohshi@gmail.com, ljilja@cs.sfu.ca

Abstract—In this paper, we employ spectral graph theory as a tool for analyzing the Internet topology. We show its importance in understanding dynamical behavior of complex networks. We also provide an overview of various approaches dealing with synchronization in complex networks.

1. Introduction

A variety of complex networks has been identified in real life. Many have universal characteristics such as small-world [1] and scale-free [2] topologies.

Analysis of complex networks often relies on discovering spectral properties of graphs capturing their topology. Such analysis is based on constructing matrices describing the network connectivity. Both the well-known adjacency matrix (also called Kirchhoff matrix) and variants of the Laplacian matrix (including normalized Laplacian and signless Laplacian matrices) of graphs capturing network structure have been employed in such analysis.

We describe analysis of large datasets collected from the Internet over several years. Spectral analysis of graphs constructed from these datasets confirms the existence of power-laws and was used to identify historical trends in the development of the network. Spectral analysis of the associated graphs also reveals historical trends in the clustering of network nodes and their connectivity. These connectivity and clustering properties of the network may be further analyzed by examining element values of the corresponding eigenvectors.

Dynamics in complex networks has recently attracted considerable research interest stimulated by the study of synchronization in systems with multiple oscillators. In 1970s, analysis of network dynamics was related to electrical networks. In this paper, we provide an overview of analysis of network dynamics with particular attention to synchronization.

2. The Internet Topology

Analyzing the Internet topology using randomly generated graphs, where routers are represented by vertices and transmission lines by edges, has been widely replaced by mining data that capture information about Internet Autonomous Systems and by exploring properties of associated graphs on the AS-level. The Route Views data [3] and RIPE [4] datasets collected from Border Gateway Protocols (BGP) routing tables have been extensively used by the research community [5]–[7]. The discovery of power-laws and spectral properties of the Internet topology indicates a complex underlying network infrastructure.

Analysis of the collected datasets indicates that the Internet topology is characterized by the presence of various power-laws observed when considering a node degree vs. node rank, a node degree frequency vs. degree, and a number of nodes within a number of hops vs. number of hops [5], [6]. Some of these early conclusions were subsequently revised by considering a more complete AS-level representation of the Internet topology [7], [8]. These extended maps have heavy tailed or highly variable degree distributions and only the distribution tales have the power-law property. It has been observed that the power-law exponents associated with Internet topology have not substantially changed over the years in spite of the Internet exponential growth [9]–[11]. Power-laws also appear in the eigenvalues of the adjacency matrix and the normalized Laplacian matrix vs. the order of the eigenvalues. They also show invariance regardless of the exponential growth of the Internet.

While various power-law exponents associated with the Internet topology have remained similar over the years, indicating that the power-laws do not capture every property of a graph and are only one measure used to characterize the Internet, spectral analysis of both the adjacency matrix and the normalized Laplacian matrix of the associated graphs reveals new historical trends in the clustering of AS nodes and their connectivity. The eigenvectors corresponding to the largest eigenvalues of the normalized Laplacian matrix have been used to identify clusters of AS nodes with certain characteristics [9]. Spectral analysis was employed to analyze the Route Views and RIPE datasets in order to find distinct clustering features of the Internet AS nodes [12]. For example, the connectivity graphs of these datasets indicate visible changes in the clustering of AS nodes and the AS connectivity over the period of five years [10]. Clusters of AS nodes can be also identified based on the eigenvectors corresponding to the second smallest and the largest eigenvalue of the adjacency matrix and the normalized Laplacian matrix [11]. The connectivity and clustering properties of the Internet topology can be further analyzed by examining element values of the corresponding eigenvectors.
3. Dynamics in Complex Networks

Earlier analysis of network dynamics addressed regular networks (nodes have equal degree) [13], [14], [15], where synchronism in the lattice, ladder, and ring networks were discussed and the conditions of complete synchronism were derived. In these papers, each node contained a Van der Pol oscillator and was connected to other nodes by resistors or inductors. In 1990’s, the complex phenomena in networks with chaotic circuits was intensively analyzed [16]–[19], [18], [19]. In electrical systems, star-connected oscillators [20] and the ring coupling of chaotic circuits were analyzed [21]. Coupled oscillators networks were also analyzed in the context of cellular neural networks [22].

3.1. Synchronization in Small-World Networks

Small-world networks have two main properties: small average distance $\bar{D}$ and high clustering. Some use only the first property as the definition of small-world networks. The answer to the question whether or not synchronization is easily achieved on a network with small-world property is somewhat surprising: The small-world property does not generally guarantee synchronization in the network [23].

3.2. Synchronization in Scale-Free Networks

Scale-free networks are characterized by the power-law connectivity distribution of certain network variables such as, for example, $P(k) \propto k^{-\gamma}$, where $P(k)$ the probability distribution function and $k$ is the node degree of the network. The smaller the parameter $\gamma$, the more the network becomes heterogeneous in its connectivity distribution and, accordingly, the average network distance decreases. However, when the average network distance becomes smaller, synchronization is more difficult to achieve. This result was explained by considering the load (information) on center nodes (hubs), of a network [24].

3.3. Synchronization in Complex Networks

In recent years, dynamical behavior of complex networks has been of particular interest. Each node in a complex network contains an oscillator or a dynamical system that generates periodic or chaotic oscillations. The network topology is represented by a Laplacian matrix $L(G)$, which is symmetric and has a single zero eigenvalue for a connected network.

The number of edges incident to a node in an undirected graph is called the degree of the node. Two nodes are called adjacent if they are connected by a link. The network can be represented by the adjacency matrix $A(G)$:

$$A(i, j) = \begin{cases} 1 & \text{if } i \text{ and } j \text{ are adjacent} \\ 0 & \text{otherwise} \end{cases}$$

A diagonal matrix $D(G)$ associated with $A(G)$, with row-sums of $A(G)$ as the diagonal elements, indicates the connectivity degree of each node. The Laplacian matrix is defined as $L(G) = D(G) - A(G)$. The eigenvalues of $L(G)$ are closely related to certain graph invariants. The spectrum of $L(G)$ is the collection of all eigenvalues and contains 0 for every connected graph component.

The general synchronization condition for complex networks with a large number of oscillators was derived in [25]. We provide here a brief overview of the employed analytical method.

3.3.1. Master Stability Equation and Master Stability Function

We consider a network with $N$ nodes and assume that each network node is governed by a self-oscillatory autonomous system with $m$ variables. For example, $m = 2$...
in case of the Van der Pol oscillator and \( m = 3 \) for the Lorenz system. We assume that the oscillators are identical with identical coupling to other oscillators.

- We first formulate the state equation of the network by introducing Laplacian matrix. For very large dimensional spaces, the direct product offers a convenient expression. In order to show the overall coupling, a constant \( \sigma \) is introduced.

Let \( \mathbf{x} \) be the \( m \)-dimensional vector of state variables of the \( i \)-th node. Let \( \mathbf{F}(\mathbf{x}) \) be the isolated (uncoupled) dynamics for each node and \( \mathbf{H}: \mathbb{R}^m \to \mathbb{R}^m \) be a coupling function. The dynamics of node \( i \) can be expressed as:

\[
\dot{x}_i = \mathbf{F}(\mathbf{x}_i) + \sigma \sum_{j=1, j \neq i}^{N} G_{ij} \mathbf{H}(\mathbf{x}_j),
\]

where \( \sigma \) is a coupling strength.

We define matrices \( \mathbf{x} = (x_1, x_2, \ldots, x_N) \), \( \mathbf{F}(\mathbf{x}) = [\mathbf{F}(x_1), \mathbf{F}(x_2), \ldots, \mathbf{F}(x_N)] \), and \( \mathbf{H}(\mathbf{x}) = [\mathbf{H}(x_1), \mathbf{H}(x_2), \ldots, \mathbf{H}(x_N)] \). Let \( \mathbf{G} \) be the \( N \times N \) matrix of coupling coefficients \( G_{ij} \). Note that \( \mathbf{G} = -\mathbf{L}(\mathbf{G}) \).

The dynamics of the network is described as:

\[
\dot{x} = \mathbf{F}(\mathbf{x}) + \sigma \mathbf{G} \otimes \mathbf{H}(\mathbf{x}),
\]

where \( \otimes \) is the direct product.

- We then seek to find the periodic solutions of the state equation (1).

- We derive the variational equation from the periodic steady-state in order to investigate the stability of synchronized steady-state or periodic solution

\[
\dot{\xi} = [\mathbf{I}_N \otimes \mathbf{D}\mathbf{F} + \sigma \mathbf{G} \otimes \mathbf{D}\mathbf{H}] \xi,
\]

where \( \xi^i \) are variations on node \( i \) and \( \xi = (\xi^1, \xi^2, \ldots, \xi^N) \).

The variational equation becomes the linear differential equation with periodic coefficients combined with the Laplacian matrix. By using an appropriate linear transformation, the variational equation can be divided in separate blocks, each block corresponding to an eigenvalue \( \gamma_k (k = 0, \ldots, N - 1) \), where \( N \) is the number of nodes:

\[
\hat{\xi}_k = [\mathbf{D}\mathbf{F} + \sigma \gamma_k \mathbf{D}\mathbf{H}] \hat{\xi}_k.
\]

Each separate block equation is called the master stability equation.

- From the variational equation, we compute the maximum Lyapunov exponents \( \lambda_{\text{max}} \) called the master stability function. If \( \lambda_{\text{max}} \) is negative, the corresponding periodic steady-state is stable and the variations die out. The stability investigation is the extension of the 2nd order nonlinear differential equation such as the Duffing’s equation [26].

Factor \( \alpha \equiv \sigma \gamma_k \), defined as the product of \( \gamma_k \) and the overall strength of coupling parameter \( \sigma \), is a measure used to express the coupling strength. The stability plots of \( \lambda_{\text{max}} \) vs. \( \alpha \) (generic coupling factor for nonlinear function and output function at each node) are used to define stability regions. The oscillatory systems such as periodic oscillators have a master stability function that has \( \lambda_{\text{max}} < 0 \) over the interval \((\alpha_{\text{min}}, \alpha_{\text{max}})\) in these stability plots. The generic requirement for the synchronous state to be stable is given by \( \sigma \gamma_k \in (\alpha_{\text{min}}, \alpha_{\text{max}}) \) for each \( k \). This requirement can be equivalently written as \( \lambda_{\text{max}}/\lambda_1 < \alpha_{\text{max}}/\alpha_{\text{min}} \), where \( \lambda_1 \) and \( \lambda_{\text{max}} \) are the second smallest and the largest eigenvalues, respectively [23]. The left-hand side of the inequality is determined solely by the Laplacian matrix while the right-hand side is defined by the master stability function. Hence, we can analyze the stability of synchronization and network dynamics by only observing the network topology.

3.3.2. The Internet Dynamics

In case of Internet graphs, the Laplacian matrix has distinct real eigenvalues \( \lambda_k (k = 0, \ldots, N - 1) \), where \( N \) is the number of network nodes [27]. The behavior of the nodes is governed by network transport protocols and queuing algorithms. Hence, the network dynamics can be described by using the fluid-model of the Transport Control Protocol (TCP) combined with the Random Early Detection (RED) queuing algorithm [28]:

\[
\frac{dw(t)}{dt} = \frac{1}{r(t)} - \frac{w(t)}{2r(t)} p(t - r(t))
\]

\[
\frac{dq(t)}{dt} = N \frac{w(t)}{r(t)} - C
\]

\[
\frac{dx(t)}{dt} = C \ln(1 - \alpha)(x(t) - q(t))
\]

\[
p_s(t) = \begin{cases} 
0 & 0 \leq x(t) < x_{\text{min}} \\
(x(t) - x_{\text{min}}) & x_{\text{min}} \leq x(t) \leq x_{\text{max}} \\
\frac{1}{2} x_{\text{max}} & x_{\text{max}} < x(t) \leq 2x_{\text{max}} \\
1 & 2x_{\text{max}} \leq x(t) 
\end{cases}
\]

\[
p(t) = kp_s(t) \quad \text{and} \quad r(t) = \frac{q(t)}{C} + R_0,
\]

where \( w(t) \) = averaged instantaneous window size (in packets) of the TCP sources

\( \lambda(t) \) = round trip time

\( q(t) \) = averaged instantaneous queue length (in packets)

\( \lambda(t) \) = filtered queue length after removal of short bursts

\( \alpha \) = marking probability

\( \alpha \) = filter resolution (0 < \alpha < 1)

\( \kappa \) = a proportionality constant dependent on the implementation of the RED algorithm

\( x_{\text{max}} \) = maximum threshold of \( x(t) \)
\[ x_{\text{min}} = \text{minimum threshold of } x(t) \]
\[ p_{\text{max}} = \text{maximum threshold of } p(t) \]
\[ R_0 = \text{propagation delay} \]
\[ C = \text{bottleneck bandwidth in packets/second} \]
\[ B = \text{maximum physical queue length} \]

4. Conclusions and Future Work

In this paper, we consider numerous new aspects of the dynamics of complex networks and we do not necessarily restrict our attention to classical small-world and scale-free networks. Included in this analysis are also many electrical networks such as regular networks. One of the problems that we plan to address is universal quantification using differential equations combined with graph theory. In various applications of complex networks, it is essential to deal with dynamics of complex networks with the weights imposed on network nodes and edges. Furthermore, we plan to develop effective methods for obtaining synchronous solutions of nonlinear equations with higher dimensions.

Acknowledgments

The authors would like to thank Laxmi Subedi for generating MATLAB figures used in this paper. This research was supported by the NSERC Discovery Grant 216844-07, the NSERC RTI Grant 330480-06, and the BCNET.

References


Abstract—Deflection routing is a mechanism to route packets. Traditional shortest path routing uses only the static topological information as input, whereas deflection routing takes into account the dynamic queue length to route a packet. In the simplest form of deflection routing, a packet being dropped due to queue buffer overflow is “rescued” and is re-routed to some other links.

Effectively, deflection routing can reduces the rate of packet drops and allows a network to carry more packets without the need of additional bandwidth. Although this is a favorable characteristic, deflection routing can lead to unstable deflecting behavior under some congestion scenario. Typically, cascading deflection can be formed. Thus, we are motivated to study deflection routing when it is operated near the point of congestion. In particular, we aim to investigating how close to the point of congestion deflection routing can operate without winding up being unstable, and how to narrower this gap as much as possible. Some observed complex dynamic behavior of deflection routing will also be reported.
Scale-Free Property of Affordable Neural Network

Yoshifumi Nishio and Yoko Uwate

Dept. of Electrical and Electronic Engineering, Tokushima University
2-1 Minami-Josanjima, Tokushima, Japan
Email: [uwate, nishio]@ee.tokushima-u.ac.jp

Abstract—We have recently proposed a novel neural network structure called an Affordable Neural Network (AfNN), in which affordable neurons of the hidden layer are considered as the elements responsible for the robustness property as is observed in human brain function. We have confirmed that the AfNN gains good performance both of the generalization ability and the learning ability. In this study, we focus on the firing frequency of neurons in the hidden layer and the amount of weight changes during the learning process. By computer simulations, we confirm that the AfNN generates scale-free properties.

1. Introduction

Since the scale-free networks were discovered by Barabasi et al. [1], studies assessing the influence of this property on the efficiency of networks have been carried out in various fields [2]-[4]. One way of how to characterize the difference between random and scale-free networks is by means of the distribution of the number of links which a node has. From Fig. 1, where we contrast the two network types, it is evident that in the scale-free network, although most nodes only have few connections, some nodes act as highly connected hubs. This distinction is captured in a more quantitative way by the distribution of the number of links vs. the number of nodes, as shown in Fig. 2. Random networks display a bell-shaped curve, implying that most nodes have the same number of links, and no highly connected nodes (see Fig. 2(a)). Scale-free networks, in contrast, often have many nodes with a few links only, whereas quite a few hubs exist that have a large number of links. Mathematically, scale-free networks are characterized by power law distributions (Fig. 2(b)). Because scale-rules emerge in many areas and disciplines of science (e.g. engineering, economics, social sciences and so on), we expect that also in the development of the science of complex networks, they will play an important role.

In a previous study on artificial neural networks, we proposed a new network structure with affordable neurons in the hidden layer, for efficient BP-learning [5]. We christened this network “Affordable Neural Network (AfNN).” In this network, some extra neurons are inserted into the hidden layer. By computer simulations [5], the AfNN has been confirmed to achieve an improved performance over conventional networks for BP-learning, in terms of speed of convergence and of learning efficiency. Moreover, we have investigated the performance of the AfNN for noise-polluted input data. We found that the AfNN is able to generate noise-cleaned outputs, which leads to the conclusion that the AfNN has the generalization property. Furthermore, the AfNN has a kind of durability, because the AfNN still performs well even if some of neurons in the hidden layer are damaged after learning process. We have confirmed that the AfNN can operates with keeping its efficiency against damaging neurons [6]. However, we believe that many advantageous characteristics of the AfNN are yet to be unveiled, and we also believe that the operation of the AfNN embodies important general features of the BP-learning process.

In this article, we investigate the scale-free property of the AfNN, which can be regarded as a complex network with time-varying connections, in order to clarify the relationship between the network topology and its information processing ability. First, we investigate the characteristics of the firing frequency of neurons in the hidden layer and the amount of weight changes during the learning process. Next, we introduce the scale-rule selection of affordable neurons and its effect is examined by computer simulations.
2. Affordable Neural Network (AfNN)

In [5], we introduced the AfNN to reflect important properties of the brain. During BP-learning, not all of the neurons in the hidden layer are used at every updating: some of the neurons are selected for the learning and the rest of the neurons are deactivated. The underlying network model is sketched in Fig. 3. The affordable neurons are selected by random every updating time (see Fig. 4). The definition of the affordable neurons is described as follows.

- The output of affordable neurons is set to zero.
- The weight vectors connected to the affordable neurons are not updated.

Figure 3: Affordable neural network.

Figure 4: Random selection of affordable neurons.

3. Pattern Recognition by Random Selection

In this study, we use the batch BP-learning algorithm. The batch BP-learning algorithm can be expressed similarly to the standard BP-learning algorithm [7], with the difference lying in the timing of the weights updating.

As the first task, we consider the pattern recognition where 10 numeric characters (Fig. 5) are fed into the neural network for recognition.

In this case, the number of neurons in the input layer is 25, and we choose 100 hidden layer neurons. The number of neurons in the output layer is 10. The network learns these 10 numeric characters with 100000 iterations and the error curve converges to almost zero. The network parameters are fixed to $\eta$ (learning rate)=0.05 and $\xi$ (proportionality factor)=0.002. After learning process, the recognition rate is investigated by using the set of 10 patterns shifted 1 bit from each original pattern. We confirm that the learned networks can obtain around 70 to 90 percent recognition rate.

Next, we investigate the following characteristics of the AfNN:

1. Firing frequency for each neuron in hidden layer.
2. Amount of weight changes between input/hidden and hidden/output layers.

Figure 6 shows the simulation result of the frequency of firing neurons in the hidden layer by using multi trial. The threshold of firing is set to 0.5. This graph is shown with double logarithmic plot. From Fig. 6, the graph curve almost follows the straight line which means this graph has some scale-free properties.

Figure 5: Pattern recognition.

Figure 6: Frequency distribution for firing neurons in hidden layer.

Figure 7 shows the frequency distribution for total adjustment of weight vectors between input and hidden layers. In this case, the graph curves does not have any rules.

While in the case of he total adjustment of weight vectors between hidden and output layers as shown in Fig. 8, the graph curve shows the straight line.

Figure 7: Frequency distribution for total adjustment of weight vectors between input and hidden layers.

Figure 8: Frequency distribution for total adjustment of weight vectors between hidden and output layers.
The total adjustment of weight vectors between input and hidden layers corresponds with the input pattern deeply. Then, the scale-free property does not occur for the total adjustment of weight vectors between input and hidden layers.

Figure 7: Frequency distribution for total adjustment of weight vectors between input and hidden layers.

Figure 8: Frequency distribution for total adjustment of weight vectors between input and hidden layers.

4. Scale-Rule Selection of Affordable Neurons [7]

In this section, we introduce the simulation results of past study in [7]. The scale-rule selection of affordable neurons has been investigated. Our scale-rule selection procedure is described in terms of a parameter denoted by a vector $S$. The dimension $s$ of $S$ equals the number of neurons present in the hidden layer; each component of $S$ corresponds to one single neuron indexed by $i$. The values of the components are evaluated in each update by

$$S_i = \text{random}() / i^2$$

where $\text{random}()$ means the uniform random function producing values from 0.0 to 1.0. This implies that the neuron with the highest index will generally have a small value, whereas the first neuron will – unless the random function states something different – have a larger entry. Note that the values of the entries follow a power-law distribution. Using these values, we select in each update the set of active neurons according to the values of $S$. From the $s$ neurons in the hidden layer, exactly the $k$ neurons with the smallest entries are chosen as the affordable neurons. Figure 9 illustrates this scale-rule selection of the affordable neurons, where the number of the neurons in the hidden layer is set to be 100 and the number of affordable neurons is 20, 40 and 60, respectively. Our simulations will be based on 100000 updates. In this Figure, the horizontal axis indicates the neuron number, whereas the vertical axis displays the number of times the corresponding neuron was in the set of operating neurons. By this Figure it is confirmed that the operation time decreases gradually with the neuron number of the hidden layer. Furthermore, histograms of the number of neurons that have a given operation time are shown in Fig. 10. The resemblance with the scale-free distribution of Fig. 2(b) is evident, although, when inspected in details, the distribution is not of a simple power law type. For comparison let us also consider random selection of affordable neurons.

Figure 9: Scale-rule selection (Hidden: 100).

Figure 10: Distribution of scale-rule selection (Affordable neuron: 40, Hidden: 100).

For our simulations we want to teach our network to generate typical time series of the skew tent map. To this end, the network is trained – using time series of the tent map – to output, starting from given initial conditions, the same
time series as the tent map would have generated.

The skew tent map and an example of time series are shown in Fig. 11. The length of chaotic time series is set to 50 steps; the size of the set of learning patterns is 20. In our approach, this requires the network to have 50 nodes in the input and the output layers. Each data is inputted to each node in the input layer. We carried out BP-learning by using the following parameters. The parameter of the learning rate and the inertia rate are fixed at $\eta = 0.05$ and $\zeta = 0.02$, respectively. The initial values of the weights are chosen between $-1.0$ and $1.0$ at random. The learning time is set to 5000.

Figure 11: Skew tent map.

We investigate the learning efficiency as the average of the total error between the output and the desired target, when the network structure of the hidden layer is changed. The “Average Error $E_{ave}$” for this learning example is defined by the following equation.

$$E_{ave} = \frac{1}{P} \sum_{p=1}^{P} \left\{ \frac{1}{2} (r_p - o_p)^2 \right\}$$

We consider the case that the hidden layer consists of 100 neurons. The number of the affordable neurons is varied from 10 to 70. The results of this simulation are shown in Fig. 12, where the horizontal axes are the number of the affordable neurons and the vertical axes are $E_{ave}$ for the pattern learning. From this Figure, we can confirm that the scale-rule selection method achieves a better performance if compare to the random selection. It is also seen that the difference between the errors of the scale-rule and the random selection networks increases with the number of affordable neurons. Even when the number of affordable neurons becomes large, the scale-rule selection network continues to show good learning ability. From this result, we can conclude that the scale-rule selection method of affordable neurons could play an important role for learning processes, in particular in biological systems.

5. Conclusions

In this study, we have investigated the characteristics of the firing frequency of neurons in the hidden layer and the amount of weight changes during the learning process. By computer simulations, we have confirmed that the AfNN could generate scale-free properties.

Acknowledgment

This work was partly supported by JSPS Grant-in-Aid for Scientific Research 22500203.

References


Dynamical AD Converter by Saito’s Rotation Map in Functional Cellular Neural Sensor Network

Mamoru Tanaka and Toshimichi Saito

Department of Electrical and Electronics Engineering, Sophia University
7-1 Kioi-cho, Chiyoda-ku, Tokyo 102-8554, Japan
Email: mamoru.tanaka@gmail.com
EECE Dept., Hosei University, Tokyo, 184-8584.

Abstract—This paper describes dynamical AD Converter in which each spiking neuron was proposed by Saito’s Rotation Map (SRM) in a general functional Cellular Neural Sensor Network. In general, a Cellular Neural Network (CNN) is defined as a local connected circuit which has continuous state variables \( x \in \mathbb{R}^n \). The importance is in that the piece-wise linear function of the CNN has a linear region \( |x| \leq 1 \) for \( x \in \mathbb{R} \) because the linear region can be quantized from the continuous variable \( x \) to the multi-level quantized variable \( f(x) \) by each 1-bit \( \Sigma \delta \) modulator which is corresponding to a spiking neuron model. The Saito’s Rotation Map (SRM) is considered as one of the efficient \( \Sigma \delta \) modulators. The firing rate is corresponding to the information. The SRM is used as local dynamics for the nonlinear function \( f(x) \) and the global dynamics by A-and B-templates are used by many image processing.

1. Introduction

In general, a semiconductor imaging device includes an array of many pixel cells for capturing image. Each pixel includes a circuit which has a photo detector, a floating diffusion region, a transfer transistor between the photo detector and the floating diffusion. The Bayer pattern pixels are read. The current flows from the top of the pixel column to a reference potential (e.g. ground) thereby producing a voltage on the column. Each of the color columns is connected to a read-out circuitry wherein the pixel values are read. Here, the column bus resistance of the pixel array is increasing. Finally, these voltages are sampled and digitized to generate digital pixel values. The readout result is large DC analog voltage difference between top and bottom readout signal paths for a given column. This analog DC voltage difference is mainly due to the resistance of the pixel output bus. A high conventional resolution AD converter is used after the analog reading. The digital outputs are transferred to a digital image processing. In future, it is very important to build 3D imaging device in which spiking pulses are read by directly converting from photo information in each pixel like a human retina system.

The \( \Sigma \Delta \) modulation (SDM) [5] [6] [8] is a technique for converting the analog signal into the digital pulse sequence. So, it is widely used by analog-to-digital converter (ADC). The main character of the SDM is high original signal reproducibility by the noise shaping characteristic. The quantization noise is distributed to the high frequency area by the over sampling technology. Moreover, if the outputs of the digital pulse signals are added, the DC information can be obtained as the average of sum of the pulse sequence, and the analog signal of the input is reconstructed by a low-pass filter.

The CNN[7] has been applied to the image processing such as image compression, filtering, and the pattern recognition, etc. The CNN with symmetrical A-template has a stable equilibrium point and if the center parameter \( a_i \) of the A-template is larger than 1, the equilibrium point is in \( \pm 1 \) saturation region corresponding to the black and the white of the image. In a word, global dynamics of the CNN in which each cell uses the piece-wise linear function can be to generate the halftoning image. However, the halftoning images are too low resolution for image processing. It is very important to construct high resolution and high intensity digital images from real analog images.

We have paid attention to the CNN in which each cell is build as SDM and in which local connection can be used to generate the multi-quantization in the linear region for piece-wise linear function and and to construct high resolution intensity digital image from real analog image[2] [1]. Of course, it is not necessary to have local connection if the output digital image is same as the input digital image with same resolution as D-D conversion. It is very important to get the high resolution digital image from more high resolution image like analog image by local connection defined by Gaussian A-template with standard deviation \( \sigma > 0 \). For the SDM type of CNN, the halftoning images produced from each cell connected in its neighborhood are added dynamically to the direction of the time sequence to construct the equivalent multi-level quantization function \( f(x) \). The quantization noise in
space can be decreased by the space integration effect of C-template and by over sampling effect. That is, a good quality halftoning image is generated in each time interval of CNN dynamics and the original image can be re-constructed through a low Gaussian filter for the sum of many halftone images[2].

But, the disadvantage of SDM type of CNN is in that the local and global dynamics corresponding to the generation of multi-quantization and image processing by A-templates have synchronization and it is difficult to perform many image processing with different A-templates.

This paper describes dynamical AD Converter in which each spiking neuron was proposed by Saito’s Rotation Map (SRM) [3] in a general functional neural sensor network. Many image processing by templates can be generated by local and global dynamics corresponding to the high resolution AD conversion and to many templates image processing which proposed by CNN researchers, respectively. In local dynamics by the SRM, the area intensity method is used to have independent high resolution digital pixel image to generate the sum of ±1 halftone sub pixels image at each pixel area corresponding to the rate of black and white area. When the size of the area elements of each pixel of the image is different, the different color images are generated. That is, In the area intensity method, there are sub pixels called area elements in each pixel. In CNN, on/off states of these area elements will be controlled by internal spiking neuron cells in each pixel according to the SRM. The proposed method is a hierarchical dynamics in which the space dynamics is divided into global and local dynamics in the CNN with SRM. The global dynamics executes the image processing based on the CNN state equation. On the other hand, the SRM local dynamics generates high resolution and high area intensity of each pixel according to the CNN state equation as a SDM with trapping window(WADC)[3]. Conventionally the digital image for 8-bit SDM generates up to minimum 256 steps from analog image. If the rotation map with a window by SRM is used, the trapping window effect by the SRM will work as encoding source for analog-to-digital CNN sensor device with higher resolution by local dynamics in more than 256 steps equivalently.

2. Dynamical Area Intensity Method

The quality of the image depends on both of intensity and resolution for displays or printers. The area intensity method expresses image as the rate of number for white and black sub elements for each pixel area. By local dynamics of each cell in SDM type of CNN, the sum of on/off states of these sub elements will be controlled to have equilibrium point in each pixel or to have pulse density average approaching to a given input pixel image. The SDM type of CNN has periodical pulses such that the average number of logical "1" pulses in a period approaches to a given input pixel DC intensity by dynamics using cells in its neighborhood.

Let \( M \times N \) be the number of pixels, then each pixel position is defined by \((I,J), I=1,2,\cdots, M; J=1,2,\cdots, N\). There are off-area element of black and on-area element of white in each pixel. The sum of the on and off area is assumed to be \( S_{\text{on}}(I,J) \) and \( S_{\text{on}}(I,J) = S(I,J) - S_{\text{on}}(I,J) \). Respectively, the sum is corresponding to the pixel intensity. That is, the gray level is described by:

\[
v = \frac{S_{\text{on}}(I,J)}{S_{\text{on}}(I,J) + S_{\text{on}}(I,J)} = \frac{S_{\text{on}}(I,J)}{S(I,J)} \quad (1)
\]

The pulse density is almost proportional to the area ratio. In the area in pixel \((I,J)\), there are sub internal cells \( P(I,J) = \{ C_{ij} | i = 1,2,\cdots, p; j = 1,2,\cdots, q \} \) of \( p \times q \). And the area element is assumed to be \( \{ S(i,j) | i = 1,2,\cdots, p; j = 1,2,\cdots, q \} \). Also, \( S(k,l) \) is assumed to be the area element for which cell \( C_{kl} \in P(I,J) \) handles. At this time, the pixel area intensity \( S_{\pm}(I,J) = S_{\text{on}}(I,J) - S_{\text{on}}(I,J) \) with the sign is controlled by on/off of cell \( C_{kl} \in P(I,J) \) and is described by

\[
S_{\pm}(I,J)(t) = \sum_{C_{kl} \in P(I,J)} S(k,l) \text{sign}(x_{kl}(t)) \quad (2)
\]

All area elements are same, the number of cells increases. And, two or more state patterns of on/off with the same sum area exist and the problem such as taking time to process is caused. However, large different area for sub elements generate area noise for printer device.

As shown in Fig 1, the hierarchical CNN with SDMs has global dynamics and local dynamics. Global dynamics executes the image processing functionally according to the design templates as:

\[
x_{IJ}(t+1) = \sum_{C(K,L) \in N(I,J)} A(I,J;K,L)y_{\pm}(K,L)(t)
+ \sum_{C(K,L) \in N(I,J)} B(I,J;K,L)u_{KL} + T(3)
\]

where \( u_{KL} \) is an analog input, and \( y_{\pm}(I,J)(t) \) is a multi-quantized function regularized by \([-1,+1]\] and described by

\[
y_{\pm}(I,J)(t) = \frac{S_{\pm}(I,J)(t)}{S(I,J)} \quad (4)
\]

Local dynamics is SDM to get the pixel intensity for the state variable input \( x_{IJ}(t+1) \) made by the amount
of pixel area intensity $S_k(I, J)$ . Internal each sub cell $C_{ij} \in P(I, J)$ operates asynchronously by

$$
\sum_{C_{ij} \in P(I, J)} x_{ij}(\bar{t} + 1) = x_{ij}(\bar{t}) + w_{i,j,k,l} f(x_{kl}(\bar{t})) + x_{IJ}(t + 1)
$$

where $f(x_{kl}(\bar{t}))$ which is a nonlinear quantization function and the $w_{i,j,k,l}$ are described by

$$
f(x_{kl}(\bar{t})) = \frac{S(k,l)}{S(I,J)} \text{sign}(x_{kl}(\bar{t}))
$$

$$
w(i,j; k,l) = \begin{cases}
-1 & \text{for } (k,l) \neq (i,j) \\
1 + \frac{h}{S(I,J)} & \text{for } (k,l) = (i,j)
\end{cases}
$$

where $h$ is controlling parameter. It is assumed $A(I,J; I,J) = 0$ now. The A-template is designed from GF by :

$$
\sum_{C(K,L) \in N, (I,J)} \tilde{A}(I,J; K,L)S(K,L) + \xi S(I,J) = 1
$$

$$
A(I,J; K,L) = \tilde{A}(I,J; K,L)S(K,L)
$$

3. Saito’s Rotation Map(SRM)

The basic ADC with trapping window (WADC) by the SRM encodes a dc input into a binary output sequence and the trapping window extracts an available part of the output sequence automatically. Using the available part, the decoder provides an estimation by a fraction with variable denominator and realizes higher resolution.

Figure 1 shows the WADC by the SRM. The WADC encodes a dc analog input $u$ to binary digital output sequence $\{y_n\}$ via a one-bit quantizer $Q$. Let $x_n \in \mathbb{R}$ ($n = \bar{t}$) be the internal state at discrete time $n$ and let $l$

be the upper limit of the output code length. Now, we consider the case that the number of sub elements in each pixel is only one here for a display not a printer. As an initial value $x_0$ is given, the WADC is updated as the following like a SDM.

$$
x_{n+1} = f(x_n) \equiv x_n - Q(x_n) + u \quad \text{for } 0 \leq n < l
$$

$$
y_n = Q(x_n) = \begin{cases}
1 & \text{for } x_n \geq 0 \\
0 & \text{for } x_n < 0.
\end{cases}
$$

We then prepare the trapping window $W$ that extracts the trapping time $p$ [3]:

$$
x_p \in W \equiv [0, 2\epsilon], x_n \notin W \text{ for } 0 < n < p
$$

where $\epsilon$ is a small positive parameter and $\epsilon < u < 1 - \epsilon$. If the trapping window does not exist for window interval parameter ($\epsilon = 0$ ), it is to be the well-known single loop SDM. After the trapping time $p$, the output $y_n$ ($n > p$) is to be unavailable. Using the available output sequence $\{y_0, \cdots, y_{p-1}\}$ with length $p$, our decoder produces an estimate $\tilde{u}$.

$$
\tilde{u} = 1 \sum_{n=0}^{p-1} y_n, u - \tilde{u} = \frac{1}{p}(x_p - x_0)
$$

If the state $x_n$ is not trapped into the window by time $l$, the update is terminated at time $l$ and $p = l$ is used in Equation (11). The WADC can estimate the input $u$ by a fraction with variable denominator. The WADC is characterized by two parameters $\epsilon$ and $l$. Figure 2 (Top) shows the basic conversion characteristics of the WADC. Figure 2 (Bottom) shows the basic conversion characteristics for $l = 8$. Figure 2 (Top) suggests that the WADC generates higher resolution than SDM.
4. Experiment Result

The simulation result shows the image of the reconstruction by the WADC type of CNN which generates the input image of 8 bit gray scale image from high resolution 16 bit medical image of $512 \times 512$ instead of analog image. And, we show the result for PSNR of the output image as shown in Fig.4. We can confirm that the PSNR by the proposal method that the proposal method has better original image reproducibility to give a good quality image of the reconstruction.

5. Conclusion

In this paper, we proposed the hierarchical dynamics CNN which has pipelining global and local dynamics. The proposal system can realize higher resolution by using WADC as $\Sigma\Delta$ modulation in the local dynamics. The WADC is one of spiking neurons. The high resolution and high intensity (about 70dB PSNR) of a image can be generated according to the SRM. The system will be used as cellular neural sensor network in which each spiking neuron in a pixel generates digital pulses from photo sensor to its corresponding column lines in terms of digital 1-bit digital pulse sequence. In future, our CNN will be used in 3D sensor imaging device.

Acknowledgments

This research is supported by the fund of Open Research Center Project from MEXT of Japanese Government (2007-2011).

References

FCC-Compliant Operation of Low-Rate UWB Impulse Radio Applying Multiple Pulses per Bit

Tamás Krébesz†, Géza Kolumbán†, Chi K. Tse‡ and Francis C. M. Lau‡

†Faculty of Information Technology, Pázmány Péter Catholic University
Budapest, Hungary
Email: {krebesz,kolumban}@itk.ppke.hu
‡Dept. of Electronic and Information Engineering, The Hong Kong Polytechnic University
Hung Hom, Hong Kong SAR, China
Email: {encktse,encmlau}@polyu.edu.hk

Abstract— Starting from the FCC Regulations and the IEEE UWB standard this contribution derives the required specifications for the blocks of UWB transceivers. It shows that if CMOS technology is used then the coverage of UWB impulse radio is limited by the low supply voltage. To overcome this limit more than one UWB carrier pulse is used to keep transmitted energy per bit high enough but to reduce the required voltage swing below the supply voltage. The optimum value of delay to be set between two successive UWB pulses is determined and the effect of frequency detuning is studied.

1. Introduction

Based on the already approved FCC Regulations and the IEEE Standard, the paper derives the system parameters for the Low-Rate (LR) Ultra-WideBand (UWB) Impulse Radio (IR) transceivers and their circuits. The key parameter is the peak pulse amplitude that determines the voltage swing in the UWB transceiver circuits. It is shown that the peak pulse amplitude allowed by the FCC Regulations cannot be exploited by the handheld CMOS devices due to the low supply voltage. A solution has been proposed to overcome this problem where one bit information is transmitted by a burst of UWB carrier pulses. This contribution determines the optimum value of delay to be set between two successive UWB pulses.

2. Allocated Frequency Band and Bandwidth for UWB Radio

The frequency band allocated to handheld UWB radio devices goes from 3.1 GHz up to 10.6 GHz. The UWB bandwidth is defined by the frequency band that is bounded by the frequencies $f_L > f_t$ where the power spectrum of radiated signal is 10 dB below the peak value.

By definition, the fractional bandwidth is given by

$$BW_{frac} = \frac{f_u - f_t}{f_u + f_t}$$

(1)

A UWB transmitter is an intentional radiator that has (i) a fractional bandwidth $BW_{frac} \geq 20\%$, or (ii) a UWB bandwidth $f_u - f_L \geq 500$ MHz regardless of the fractional bandwidth.

The only IEEE Standard already approved for the UWB impulse radio was elaborated by the IEEE 802 LAN/MAN Standards Committee as an amendment to IEEE 802.15.4-2006 in 2007 [2].

3. Carrier of UWB IR Devices

Because of its easy implementation with CMOS technology [3], easy mathematical handling and IEEE Standard 802.15.4a [2] compliance the frequency-shifted gaussian pulse is considered here as UWB IR carrier pulse:

$$g(t) = p(t) \cos(\omega_c t) = \sqrt{\frac{2Z_0 E_b}{k \sqrt{\pi} u_B}} \exp\left(-\frac{t^2}{2u_B^2}\right) \cos(\omega_c t)$$

(2)

where $p(t)$ is the lowpass gaussian envelope, $f_c = \omega_c/2\pi$ is the center frequency of the gaussian pulse, $Z_0$ is the characteristic impedance over which $E_b$ is measured and $u_B$ is determined by the required 10-dB RF bandwidth $2f_B$ of UWB wavelet

$$u_B = \frac{1}{2\pi f_B \sqrt{\log_10(e)}}$$

(3)

To increase the radio coverage, one bit information is transmitted by a burst of UWB carrier pulses [3]. Parameter $k$ in (2) gives the number of UWB pulses used to carry one bit information. In the remaining part of the contribution, $g(t)$ is referred to as UWB carrier pulse.

The peak pulse amplitude is obtained from (2) as

$$V_{peak} = \sqrt{\frac{2Z_0 E_b}{k \sqrt{\pi} u_B}}$$

(4)

The idea of effective pulse width, introduced in spectrum analysis [4], is used to characterize the UWB pulse duration

$$\tau_{eff} = \int_{-\infty}^{\infty} \frac{p(t)}{V_{peak}} dt = \sqrt{2\pi u_B} = \frac{1}{f_B \sqrt{2\pi \log_{10}(e)}}$$

(5)

4. Derivation of Peak Pulse Amplitude

4.1. Interpretation of FCC Peak Power Limit

According to the FCC Regulations [1], the peak power level of UWB emission has to be measured within a 50-MHz bandwidth centered on the frequency at which the highest radiated emission occurs.

The frequency-shifted gaussian pulse (2) achieves its highest emission at the carrier frequency $\omega_c$. Let an isotropic radiator be used at the UWB transmitter and consider an RF bandpass filter characterized by its impulse response $h(t)$.

The test configuration defined in the FCC Regulations is shown in Fig. 1 where the bandwidth and center frequency of the RF
The RF bandpass filter are $RBW_{50}^{FCC} = 50$ MHz and $\omega_C$, respectively. The RF bandpass filter is excited by the UWB pulse $g(t)$. To establish the relationship among the FCC Regulations, $V_{peak}$ and $E_b$, the peak power at the filter output has to be found.

Let the frequency-shifted Gaussian pulse $g(t)$ in Fig. 1 be expressed as the product of three terms

$$g(t) = \tau_{eff} V_{peak} p(t) \cos(\omega_C t)$$

(6)

where $\tau_{eff}$ and $V_{peak}$ are defined by (5) and (4), respectively, and $p(t)$ is a gaussian function can be easily expressed from (2) as

$$p(t) = \frac{2Z_0 E_b}{k \sqrt{\pi} u_b} \exp \left( \frac{t^2}{2u_b^2} \right)$$

(7)

Consider the subcircuit included in the dashed box in Fig. 1. To get the simplest model, the RF bandpass filter is substituted by its lowpass equivalent [5] as shown in Fig. 3 where

$$h_\ell(t) = h_1(t) + jh_2(t)$$

denotes the complex impulse response of the RF bandpass filter.

The only duty of the RF bandpass filter depicted in Fig. 1 is to limit the bandwidth of $g(t)$ according to the FCC Regulations, consequently, even an ideal RF bandpass filter can be used. Then $h_2(t) = 0$ and the complex impulse response of lowpass equivalent of RF bandpass filter takes the form [5]

$$h_\ell(t) = h_1(t) = 4B \text{sinc}(RBW_{50}^{FCC} t)$$

(8)

The choice of an ideal RF bandpass filter does not restrict the validity of the lowpass equivalent model but simplifies it considerably because the two blocks characterized by the impulse response $h_2(t)$ in Fig. 3 can be canceled.

The lowpass equivalent can be simplified further if the circuits included in the dashed and dotted boxes in Fig. 3 are merged. Consider the dashed box first that contains two multipliers, a lowpass filter and an amplifier with a gain of 2. The equivalent transfer function of these circuits is equal to 1. Similarly, the equivalent transfer function of the circuits that are involved in the dotted box is equal to 0.

The derived lowpass equivalent model of the peak power calculation is shown in Fig. 2, where the complex impulse response of the lowpass filter is given by (8). Note, the cutoff frequency of equivalent lowpass filter is 25 MHz, the half of the RF bandwidth of RF bandpass filter specified in the FCC Regulations.

The two models shown in Figs. 1 and 2 are equivalent in the sense that their inputs $p(t)$ and outputs $y(t)$ are identical. The peak power limited by the FCC Regulations can also be determined by means of the lowpass equivalent model. Then the relationship among the FCC regulations and the peak pulse amplitude $V_{peak}$ and UWB bit energy $E_b$, must known parameters for the circuit designers and system engineers, respectively, can be established.

Figure 2: Lowpass equivalent of the peak power level calculation.

The excitation $p(t)$, applied to the input of lowpass equivalent model, is a gaussian nascent function that implements a delta function provided that $\sqrt{2}\pi u_b \to 0$. From an engineering point of view this condition is satisfied when the bandwidth of the gaussian nascent function $p(t)$ is much larger than that of the equivalent lowpass filter. This condition is always satisfied in UWB IR systems since $u_b \gg RBW_{50}^{FCC}/2$.

If the excitation $p(t)$ in Fig. 2 can be considered as a unit impulse function then the output of the lowpass equivalent filter is nothing else as its impulse response $h_\ell(t)/2$ and the output of the RF bandpass filter defined by the FCC Regulations is obtained as

$$y(t) = \frac{\tau_{eff} V_{peak}}{2} h_\ell(t) \cos(\omega_C t)$$

$$= RBW_{50}^{FCC} \tau_{eff} V_{peak} \text{sinc}(RBW_{50}^{FCC} t) \cos(\omega_C t)$$

(9)

Note, except a weighting factor $\tau_{eff}/V_{peak}$, $y(t)$ is equal to the impulse response of the filter defined in the FCC Regulations.

Let the RF bandpass filter be terminated by $Z_0$ ohms. The peak power is measured at $t = 0 s$ and is obtained as

$$P_{peak}^{FCC} \equiv (0 \text{ dBm EIRP}) = \frac{y(0)^2}{Z_0} = (RBW_{50}^{FCC} \tau_{eff})^2 \frac{V_{peak}^2}{Z_0}$$

(10)

5. Derivation of Specification for UWB Circuits

Each UWB transceiver contains many circuits from the transmit power amplifier to the low-noise preamplifier. To develop these circuits, the voltage swings caused by the UWB carrier pulse and the specification for their frequency responses have to be known.

5.1. An Important Property of UWB Circuits

The model shown in Fig. 2 highlight a very important and unique property of UWB circuits that cannot be neglected. The conventional communication circuits almost always operate in steady-state, the transient responses of the circuits are generally neglected. The situation is very different in UWB impulse radio where extremely short pulses are used as carriers. Since the bandwidth of UWB pulses is large compared to that of the systems or circuits being excited by the UWB pulse, the excitation may be considered as a unit impulse function. The response of the excited circuit is equal to its impulse response, consequently, the transient response of excited circuit cannot be neglected.

5.2. Required Peak Pulse Amplitude

The peak pulse amplitude determines the linearity requirements and the required supply voltage that is crucial in handheld and mobile LR-WPAN/WLAN applications.

Equation (10) establishes the relationship between the FCC Regulations and the peak pulse amplitude. The paper considers only low-rate UWB systems which are peak power limited [3].

The supply voltage of low-cost, low-power CMOS SoC UWB radio systems is less than 1.5 V. The low supply voltage limits
the maximum attainable peak-to-peak output voltage swing at the power amplifier output in about 1 V. Therefore, the large peak pulse amplitude allowed by the FCC Regulations cannot be exploited. The low attainable peak pulse amplitude results in a low $E_b$ and, consequently, in a very short radio coverage.

This observation has a serious consequence. The LR UWB IR devices cannot exploit, even theoretically, the FCC peak power limit.

### 5.3. Reduction of peak pulse amplitude

In the low-rate UWB IR systems the required peak voltage amplitude may be reduced considerably while keeping $E_b$ high enough if more than one UWB carrier pulse is used to transmit one bit information. This solution is shown in Fig. 4 where 5 UWB carrier pulses are used to transmit one bit information. Recall, parameter $k$ appearing in (2) and (4) was introduced to specify the number of UWB pulses used to carry one bit information.

Consider a LR UWB IR device where $k = 5$. Because of the handheld application, let the peak pulse amplitude be limited in 0.5 V as shown in Fig. 4. Note, the time delay $t_{\text{delay}}$ elapsed between two successive UWB pulses is a free design parameter that can be exploited to optimize the parameters of UWB transmitter. Let the effect of time delay be studied first.

![Figure 3: Substitution of the RF bandpass filter by its lowpass equivalent.](image)

![Figure 4: Transmission of one bit using five UWB carrier pulses in a burst.](image)

![Figure 5: FCC filter output when $k = 5$ and $t_{\text{delay}}/\tau_{\text{eff}} = 8$. The 1-mW FCC peak power limit is shown by dashed curve.](image)

![Figure 5: FCC filter output when $k = 5$ and $t_{\text{delay}}/\tau_{\text{eff}} = 8$. The 1-mW FCC peak power limit is shown by dashed curve.](image)

To find the optimum UWB transmitter configuration, the relationship among the (i) peak pulse amplitude, (ii) time delay between the successive UWB carrier pulses and (iii) number of pulses used to transmit one bit information has to be found.

Let us consider the maximal value of peak pulse amplitude that satisfy the FCC Regulations. Let $V_{\text{p,peak}}^i$ denote this amplitude. To get the maximal coverage $V_{\text{p,peak}}^i$ should be maximized. Unfortunately, in many CMOS implementation $V_{\text{p,peak}}^i$ is limited by the supply voltage.

The relationship among these effects can be observed in Fig. 7 where $V_{\text{p,peak}}^i$ is plotted against the number $k$ of pulses used to carry one bit information and where the parameter is the normalized de-
Figure 6: FCC filter output when $k = 5$ and $t_{\text{delay}}/\tau_{\text{eff}} = 2$. The 1-mW FCC peak power limit is shown by dashed curve.

Figure 7: The peak pulse amplitudes of the UWB carrier pulses that belong to the 1-mW FCC peak power limit. The parameter is the normalized time delay $t_{\text{delay}}/\tau_{\text{eff}}$ elapsed between the successive UWB carrier pulses. Its values are 2, 3, 4, 6 and 8 from the bottom to the top.

lay elapsed between two successive UWB pulses. The number $k$ of UWB carrier pulses can take only integer values but in order to get an easy-to-use figure, the values belonging to the same normalized delay are connected by solid curves. Note, if $k \geq 5$ then $V_{\text{peak}}^1 mW$ is almost independent of $k$, but heavily depends on $t_{\text{delay}}/\tau_{\text{eff}}$.

Let us consider the case when $V_{\text{peak}}^1 mW$ is not limited by the supply voltage and $k \geq 5$. The radio coverage depends on $V_{\text{peak}}^1 mW$, the higher the $V_{\text{peak}}^1 mW$ the larger the coverage. According to Fig. 7, the larger coverage is achieved by $t_{\text{delay}}/\tau_{\text{eff}} = 8$, where $V_{\text{peak}}^1 mW$ achieves its maximal value.

When $V_{\text{peak}}^1 mW$ is limited by the supply voltage then the duration of UWB carrier burst becomes a free parameter. Assume that $V_{\text{peak}}^1 mW$ is limited in 0.5 V. Figure 7 shows that $t_{\text{delay}}/\tau_{\text{eff}}$ should not be below 3 otherwise the FCC peak power limit is not met.

6. Sensitivity to detuning of center frequencies

The center frequency of FCC filter used to check the peak power limit has to be "centered on the frequency at which the highest radiated emission occurs" [1]. The center frequency of a UWB carrier burst is equal to the center frequency $\omega_{C}$ of the frequency-shifted gaussian pulses (2).

The UWB carrier pulses are generated by CMOS digital circuits [3], the FCC peak power limit is checked by an LC filter. Both $\omega_{C}$ and the center frequency of an LC filter may deviate from their nominal value. The sensitivity to frequency detuning has to be determined.

Consider a UWB carrier burst where $k = 5$ pulses are used to carry one bit information. Let the normalized delay between two consecutive UWB pulses set to 2. The effect of detuning is plotted in Fig. 8 where the center frequency, that is equal to 4 GHz, is varied. Note, there is no need for extra precautions since the maximum peak power emission occurs when $\omega_{C}$ coincides with the center frequency of the FCC filter.

Figure 8: Effect of center frequency detuning on the maximum peak power emission. The frequency axis shows the detuning in MHz, the center frequency is 4 GHz.

7. Conclusions

The low rate UWB impulse radio is peak power limited. Starting from the FCC peak power limit, the paper derived the maximal peak pulse amplitude that determines the coverage of UWB radio. Since the coverage of UWB IR using one carrier pulse is too short, not a single but a burst of UWB impulses is used to carry one bit information. The relationship among the UWB burst parameters and peak pulse amplitude has been determined, the effect of time delay elapsed between two consecutive UWB pulses has been revealed. The sensitivity of frequency detuning has been shown.

8. Acknowledgments

This work has been financed by the Hungarian-Chinese Inter-governmental S&T Cooperation Programme. T. Krébesz’s participation was supported by the Mecenatura Fund, National Office for Research and Technology (NKTH), Hungary.

References


Radio Coverage Extension of the FCC-Compliant Low-Rate UWB Networking Devices

Géza Kolumbán†, Tamás Krébesz‡, Francis C. M. Lau‡ and Chi K. Tse‡

†Faculty of Information Technology, Pázmány Péter Catholic University
Hung Hom, Hong Kong SAR, China
Email: [kolumban,krebesz]@itk.ppke.hu
‡Dept. of Electronic and Information Engineering, The Hong Kong Polytechnic University
Hung Hom, Hong Kong SAR, China
Email: [encmlau, encktse]@polyu.edu.hk

Abstract—In UWB systems three factors have to be considered to design a radio link: (i) peak- and (ii) average power limits defined by FCC Regulations and (iii) low supply voltage available in handheld cheap devices. An exact mathematical model for the interpretation of FCC Regulations is derived. The radio coverage of UWB networking devices is determined and a chaos-based approach for the coverage extension is proposed.

1. Introduction

Ultra-WideBand (UWB) technology allows the reuse of frequency bands occupied by narrowband wireless systems by keeping the peak- and average power levels at such low values regulated by the Federal Communications Commission (FCC, USA) [1] that does not cause any harmful interference. This contribution interprets the FCC Regulations to clarify the main aspect that is usually overlooked, namely that the power limits have to be checked not over the transmitted UWB signal but at the output of a bandpass filter specified in the FCC Regulations.

In 2007 IEEE 802.15.4a Task Group approved a standard [2] for the Low-Rate (LR) UWB applications. The standard focuses mainly on the Impulse Radio (IR) approach but the application of chirp signal or chaotic signal as carriers are also permitted in UWB applications.

LR UWB Impulse Radio transceivers have been designed and built at MIT, USA [7], and in the framework of two large European projects called PULSERS [3] and EUWB [4]. The results of field tests have shown that the sensitivity of the built receivers is too low, the coverage of implemented UWB IR systems is only a few meters. Therefore an increase in bit energy that increases the coverage is essential.

Starting from the FCC Regulations, the paper introduces a novel approach for the link budget calculation to reveal the reason of low coverage. Since the ultra short duration of UWB carrier pulses is responsible for the low coverage of UWB impulse radio the UWB carrier-based approach is proposed here where the duration of UWB carrier is increased considerably and the ultra-wide bandwidth is assured by a chaotic signal. The chaotic carrier-based approach allows to increase the bit energy even by 20 dB that extends the radio coverage considerably.

2. Preventing Interference in Conventional Systems

To maximize sensitivity, conventional radio systems use narrowband receivers that can be modeled by a cascade connection of a bandpass channel filter and a demodulator.

2.1. Classes of UWB Carriers

Two classes of UWB carriers can be distinguished considering the two kinds of interferences caused: (i) UWB impulse radio with very low duty cycle and (ii) UWB carrier-based radio. In the former case the average level of corruption is negligible but the huge peaks interrupt the operation of synchronized subsystems at the victim receiver. In the latter case no huge peaks occur, the UWB signal is a wideband signal with almost uniform psd that increases noise level at the demodulator input of victim receiver.

2.2. FCC Power Limits

The FCC Regulations consider both kinds of interference but say nothing about the type of UWB carrier. They limit both the peak- and average powers of UWB carrier:

1. “There is a limit on the peak level of the emissions contained within a 50-MHz bandwidth centered on the frequency at which the highest radiated emission occurs . . . . That limit is 0 dBm EIRP.”

2. The average “radiated emissions . . . shall not exceed” −41.3 dBm EIRP “when measured using a resolution bandwidth of 1 MHz” over the frequency band of 3.1 GHz to 10.6 GHz. “The RMS average measurement is based on the use of a spectrum analyzer with a resolution bandwidth of 1 MHz, an RMS detector, and a 1 ms or less averaging time.”

Note, FCC Regulations gives not only the limits but also instructions how the power limits have to be measured. FCC Regulations have been derived from the model of a narrowband receiver. Observe, neither the FCC peak nor average power limits are directly applied to the modulated UWB signals, instead, they give limits on the outputs of the two specified FCC bandpass filters.

3. Carrier used by UWB IR Devices

Both narrowband and wideband UWB IR devices have been defined in the IEEE Standard 802.15.4a–2007 [2], the bandwidth of the former and latter devices are about 499.2 MHz and 1.3312 GHz, respectively. The UWB IR carrier is a bandpass signal that can be decomposed into a lowpass envelope and a sinusoidal carrier. The standard does not specify the exact shape of envelope.

Because of its easy implementability with CMOS [7] and easy mathematical handling, the frequency-shifted gaussian pulse is considered here as a UWB IR carrier

\[ g(t) = p(t) \cos(2\pi f_c t) = V_{peak} \exp\left(-\frac{t^2}{2\sigma^2}\right) \cos(2\pi f_c t) \]  

(1)
where \( p(t) \) is the lowpass gaussian envelope, \( f_c \) is the center frequency of UWB carrier and \( V_{peak} \) is the pulse peak amplitude

\[
V_{peak} = \sqrt{\frac{2Z_0 E_b}{\sqrt{\pi} u_B}} \tag{2}
\]

In (2), \( Z_0 \) denotes the characteristic impedance over which the energy per bit \( E_b \) is measured and \( u_B \) is determined by the 10-dB RF bandwidth \( 2f_b \) of UWB IR carrier

\[
u_B = \frac{1}{2\pi f_b \sqrt{\log_{10}(e)}} \tag{3}
\]

Due to the gaussian envelope, the UWB carrier decreases rapidly as a function of time. Effective pulse width, introduced in spectrum analysis [5], is used to define UWB pulse duration

\[
\tau_{eff} = \int_{-\infty}^{\infty} \frac{p(t)}{V_{peak}} dt = \sqrt{2\pi} u_B = \frac{1}{f_b \sqrt{2\pi \log_{10}(e)}} \tag{4}
\]

Since only the narrowband UWB IR devices are feasible today, only that one is considered here where \( \tau_{eff} = 2.43 \) ns.

4. Attainable Coverage of UWB IR Devices

The coverage of a radio device depends on the energy per bit. The higher the \( E_b \), the larger the coverage.

4.1. Limits on Energy per Bit

\( E_b \) is limited by three issues: (i) FCC peak and (ii) FCC average power limits, and the (iii) low supply voltage of handheld devices.

4.1.1. FCC Peak Power Limit

The peak emission defined at the output of a bandpass filter having a bandwidth of \( \mathrm{RBW}_{\text{UWB}} = 50 \) MHz shall not exceed 1 mW. Let this bandpass filter be referred to as the FCC filter.

To express the relationship between the envelope \( p(t) \) of the UWB IR carrier and the FCC peak power limit, a lowpass equivalent model for the interpretation of FCC peak power limit has been developed. In the equivalent model, depicted in Fig. 1, the lowpass equivalent of the FCC filter is driven by the envelope of UWB IR carrier. The cutoff frequency of lowpass equivalent filter is equal to the half of the FCC filter bandwidth, that is, \( \mathrm{RBW}_{\text{UWB}}/2 = 25 \) MHz.

\[
E_{\text{peak}} = \sqrt{\frac{2Z_0 E_b}{\pi u_B}} \tag{5}
\]

4.1.2. FCC Average Power Limit

The average power level of UWB emission has to be measured by a spectrum analyzer with a resolution bandwidth of 1 MHz, an RMS detector, and a video filter that has 1 ms or less averaging time. The lowpass equivalent model of the FCC average power limit measurement is depicted in Fig. 2. The Rms filter is equal to the half of the FCC filter bandwidth, that is, \( \mathrm{RBW}_{\text{UWB}}/2 = 500 \) kHz.

\[
P_{\text{avg}} = \frac{E_{\text{peak}}}{2Z_0} \tag{6}
\]

Figure 2: Lowpass equivalent model of FCC average power limit calculation.

According to the FCC Regulations the average power \( P_{\text{avg}} \) may not exceed \(-41.3 \) dBm EIRP

\[
P_{\text{avg}}^\text{FCC} = -41.3 \text{ dBm} = \frac{E_b}{2Z_0} 1.06 \mathrm{RBW_{UWB}} \frac{\tau_{eff}^2}{T_{\text{bin}}} \tag{6}
\]

where \( T_{\text{bin}} \) is the bit duration and constant of 1.06 appears since not ideal but gaussian filters are used in spectrum analyzers [5].

4.1.3. Supply Voltage Limit on Pulse Peak Amplitude

The supply voltage of low-power, handheld UWB transceivers is less than 1.5 V. In the low-cost applications, considered here, an LC-transformer cannot be used to increase the signal amplitude. Consequently, the low supply voltage limits the maximum attainable peak-to-peak voltage swing at the transmitter output in 1 V. The supply voltage limit on \( E_b \) is obtained from (2) as

\[
E_b = \frac{\sqrt{2} \nu_b^2 V_{\text{peak}}^2}{2Z_0} \tag{7}
\]

4.2. Link Budget Calculation

In conventional telecommunications systems, where the carrier waveform fills up the entire bit duration, the link budget calculation relies on the signal-to-noise ratio. This approach cannot be used in the UWB IR systems where the duty cycle is extremely low and the peak-to-average power ratio is extremely high. Instead, the link budget calculation must rely entirely on \( E_b \).

4.2.1. Attainable Excess Link Margin

Let \( (E_b/N_0)^{[\text{FCC}]} \) and \( (E_b/N_0)^{[\text{DE}\text{M}]} \) denote the UWB signal energy per bit-to-noise power spectral density ratio available at the transmitter and required by the demodulator, respectively. The attainable excess link margin is

\[
\text{ELM}_{\text{attain}} = (E_b/N_0)^{[\text{FCC}]} - (E_b/N_0)^{[\text{DE}\text{M}]} \tag{8}
\]

Since \( N_0 = -174 \) dBm | 1 Hz, the first term in (8) is determined by the maximum attainable \( E_b \). As shown in Sec. 4.1, \( E_b \) is limited by the FCC Regulations and the supply voltage.

The effects of the three limits on \( E_b \) have been calculated from (5), (6) and (7) and are plotted in Fig. 3 where \( (E_b/N_0)^{[\text{FCC}]} \) is plotted against the data rate. The solid and dashed curves give the limits imposed by the FCC Regulations and the supply voltage, respectively. Two important conclusions may be drawn: (i) the low-rate UWB IR devices are peak power limited while the high-rate systems are average power limited and (ii) the low supply voltage of
handheld UWB IR devices prevents even the exploitation of FCC peak power limit, a further loss in maximum attainable $E_b$ occurs.

![Graph showing PL and BER vs. Data Rate](image)

Figure 3: Determination of excess link margin. Solid and dashed curves show $(E_b/N_0)_d$ limited by the FCC Regulations and 1.5-V supply voltage, respectively. Dashed-dotted curve shows the typical value of $(E_b/N_0)$ required by a noncoherent UWB demodulator at $BER = 10^{-3}$.

The noncoherent UWB IR demodulators, the only feasible implementation for the low-cost CMOS receivers requires $(E_b/N_0)_d(\text{DEM}) \approx 14$ dB to assure a BER of $10^{-3}$ [6]. This value is plotted by dash-dotted curve in Fig. 3.

Figure 3 gives the excess link margin graphically for a specified data rate as the distance measured between the curves of $(E_b/N_0)_d(T_x)$ and $(E_b/N_0)_d(\text{DEM})$. As examples, consider two narrowband UWB IR systems with the data rates of 10 kbps and 10 Mbps. As shown by arrow "a" $E_{\text{LM} \text{total}} = 76.3$ dB in the former case and the coverage is limited by the supply voltage. Arrow "b" gives $E_{\text{LM} \text{total}} = 73.2$ dB for the high-rate system where the coverage is limited by the FCC average power limit.

4.2.2. UWB IR Device Related Parameters

The $E_{\text{LM} \text{total}}$ has to cover the (i) path loss, (ii) receiver noise contribution and the (iii) effects of transmit and receive antennas

$$a_{CH} + N_{F_{Rx}} + I_{LOS} - G_{T_x} - \eta_{\text{fig}}(f) - G_{R_x} - \eta_{\text{ant}}(f) \tag{9}$$

where $a_{CH}$ is the path loss including multipath, $N_{F_{Rx}}$, and $I_{LOS}$ denote the overall noise figure and implementation loss, respectively, of the UWB receiver. The gain and frequency dependence of transmit and receive antennas are accounted by the $G$ and $\eta_{\text{ant}}(f)$ terms. Note, each parameter has to be substituted in decibels in (9).

The parameters of built UWB impulse radio receivers have been reported recently in [7]. Only low- or zero-gain antennas can be used in handheld applications. Let the frequency dependence of antenna be neglected in our calculation. Then the sum of UWB IR device related parameters is about 14 dB.

4.2.3. Path Loss

The path loss is a random variable, its mean depends on both the distance of transmit and receive antennas and the frequency. The Channel Modeling Subgroup of IEEE 802.15.4a adopted the "power-law" model and decomposed the path loss into four terms

$$a_{CH} = P_L d + 10 n \log_{10} (d/d_0) + 20(\kappa + 1) \log_{10} (f/f_0) + S \tag{10}$$

where $P_L$ is the path loss at the reference distance $d_0 = 1$ m and reference frequency $f_0 = 5$ GHz, $n$ is the path loss exponent, $\kappa$ describes the frequency dependence of path loss and $S$ is a random variable with zero mean. The distance $d$ and frequency $f$ have to be substituted in meters and GHz, respectively.

4.3. Coverage of UWB IR Devices

To get the radio coverage, first the UWB IR device related parameters have to be deduced from the attainable $E_{\text{LM}}$ plotted in Fig. 3. Then substituting $S = 0$ into (10), the mean coverage can be calculated.

Consider a typical low-rate application where the data rate is less than 75 kbps. Assuming Non-Line-Of-Sight (NLOS) propagation (10) gives 2 m and 1.4 m for the mean value of coverage in indoor residential and office applications, respectively. These attainable link distances are so short that the implementation of WLAN systems with UWB impulse radio devices is not feasible.

To solve the problem, $(E_b/N_0)_d(\text{DEM})$ must be increased considerably.

There are two solutions to the problem: the energy per bit can be increased by using (i) more than one UWB carrier pulse to transmit one bit information or by applying (ii) UWB carrier-based approach. The latter approach is considered here.

5. Improving UWB Radio Coverage using the UWB Carrier-Based Approach

The very short pulse duration and the 1.5-V supply voltage limit are responsible for the low $E_b$ in UWB impulse radio. In the UWB carrier-based approach an inherently wideband wavelet is used as UWB carrier, consequently, the duration of UWB wavelet can be increased considerably. Recall, the larger the wavelet duration, the larger the $E_b$ and the UWB radio coverage.

5.1. Generation of UWB Carrier

The block diagram of UWB carrier generation is shown in Fig. 4 where a Bernoulli shift map is used to provide a discrete-time chaotic signal which is converted into an analog waveform by a Zero-order Hold (ZoH) circuit. Then an FM modulator generates an ultra-wideband signal with constant envelope. Finally, a switch is used to form the UWB carrier wavelets with duration of $T_{ch}$ and repetition rate of $T_{\text{bin}}$.

![Diagram showing UWB carrier generation](image)

Figure 4: Generation of a constant envelope UWB carrier.

As shown in Fig. 5, a smooth spectrum satisfying the FCC Regulations is obtained with an adequate choice of circuit parameters. The duration of UWB carrier wavelets has been chosen to 300 ns in order to permit the generated chaotic signal to cover the entire state space of Bernoulli shift generator. If so then the spectrum of each UWB carrier wavelet becomes identical.

5.2. Exploitation of FCC Peak Power Limit

As shown in Fig. 3, the low-rate UWB IR devices are peak power limited. Unfortunately, the handheld devices even cannot exploit the FCC peak power limit because their voltage swing at the transmitter output is limited by the low supply voltage.

The use of UWB carrier based approach increases $E_b$ in two ways: (i) increases the duration of carrier wavelet (note, $\tau_{ch} = 2.43$ ns $<< T_{ch} = 300$ ns) and, simultaneously, (ii) reduces...

- 324 -
the wavelet amplitude that belongs to the FCC peak power limit. The reduced amplitude remains below the supply voltage limit. Recall, the FCC peak power limit says nothing about the power or amplitude of the UWB carrier, instead, it limits the peak power at the output of the FCC bandpass filter. The relationship between the parameters of UWB carrier (amplitude, duration, shape) and peak power output of FCC bandpass filter is not straightforward, an analytical expression has not yet been found.

Consider a constant envelope UWB carrier generated by the block diagram of Fig. 4, characterized by the spectrum shown in Fig. 5 and having a duration of 300 ns. Recall, the bandwidth of FCC filter is 50 MHz. The response of the FCC bandpass filter to the chaos-based UWB carrier wavelet is shown in Fig. 6. Both the transient and steady-state responses can be observed in the FCC filter output that carries an AM in spite of the fact that the input is a constant envelope signal. The amplitude of UWB carrier remains below 0.22 V; consequently, the low supply voltage of the handheld UWB device does not limits $E_b$ anymore. Note, the UWB carrier-based approach allows to increase the wavelet duration considerably and also permits to fully exploit the FCC peak power limit in handheld applications.

5.3. Enhancement in Attainable Energy per Bit

To show the effectiveness of the technique proposed here the attainable energy per bits of two narrowband handheld UWB systems offering data rate of 75 kbps have been compared. In the reference solution the UWB impulse radio approach is used where 1.5-V supply voltage limits the attainable $E_b$. Due to the low $E_b$, UWB IR devices offer an unacceptable short radio coverage. The relative enhancement in $E_b$ is plotted in Fig. 7 as a function of the UWB carrier wavelet duration. A considerable enhancement can be achieved, for example, the improvements in $E_b$ are 16.2 dB and 20 dB when the durations of UWB carrier wavelet are 300 ns and 800 ns, respectively. According to (10), this large improvement significantly increases the coverage of UWB radio devices.

6. Conclusion

The coverage of UWB impulse radio devices is limited by the FCC Regulations because they restrict the attainable energy per bit. The low supply voltage of handheld UWB IR devices limits even further the attainable $E_b$. Deriving a novel approach for the link budget calculation this paper has shown that the maximum coverage of UWB IR devices becomes less than 2 m in indoor applications under NLOS propagation conditions. That short coverage prevents the application of the UWB impulse radio approach in WLAN applications.

Since the extremely short pulse duration is responsible for the low $E_b$ in UWB impulse radio, the use of FM modulated chaotic waveforms with relatively long duration is proposed here as a UWB carrier. A 16-dB and 20-dB improvements in $E_b$ have been achieved with the UWB carrier durations of 300 ns and 800 ns, respectively. That huge increase in energy per bit considerably improves the coverage of UWB radio devices.

7. Acknowledgments

This work has been financed by the Hungarian-French Intergovernmental S&T Cooperation Programme. T. Krébesz’s participation was supported by the Mecenatura Fund, National Office for Research and Technology (NKTH), Hungary.

References

Application of Open-Plus-Closed-Loop Control to Secure Communications Using Chaos Masking

Yusuke Yoshida and Takaya Miyano

Department of Micro System Technology, Ritsumeikan University, 1-1-1 Noji-Higashi, Kusatsu, Shiga 525-8577, JAPAN
Email: tmiyano@se.ritsumei.ac.jp

Abstract—We have applied the open-plus-closed-loop control method, recently devised by Grosu et al., to secure communications using chaos masking. In our method, a message is treated as part of the parameter mismatch between the chaotic oscillators installed on a drive and a response system. In the drive system, a message is added to the state variable of the chaotic oscillator as dynamical noise and thus the time-integrated message is encrypted by the chaotic signal. In the response system, the message is decrypted by subtracting the chaotic carrying signal reproduced by chaotic synchronization using the open-plus-closed-loop control method from the received signal, followed by differentiation with respect to time. We show our experiment for the encryption and decryption of an actual speech signal to assess the performance of our method.

1. Introduction

Chaos synchronization is one of important discoveries in nonlinear science, which has stimulated the development of applications of chaos [1, 2]. An outstanding example is secure communications using chaos masking, which was discovered by Cuomo, Oppenheim and Strogatz [3, 4]. Their method relies on the fact that a couple of identical chaotic oscillators sharing a particular state variable can synchronize with each other because of negativity of the conditional Lyapunov exponents. It has recently been applied to a commercial communications network system with optical devices as the chaotic oscillators [5]. Advanced analysis concerning chaos-based communications have recently been performed, for instance, in [6, 7]. However, Cuomo-Oppenheim's method has two weak points. One is negativity of the conditional Lyapunov exponent and the other is parameter mismatch between chaotic oscillators that is unavoidable in actual communications systems. Because of negative conditional Lyapunov exponents, which are required for the synchronous state of the oscillators to be dynamically stable, we are obliged to have a narrow choice of the dynamics governing the oscillators and the mode of coupling between the oscillators. This allows eavesdroppers to identify the dynamics. Parameter mismatch causes a technical problem of the need to precisely tune the chaotic oscillators installed on the drive and response systems.

We have recently proposed an alternative method for secure communications using chaos masking that is free from not only the conditional Lyapunov exponent but also the precise tuning of the chaotic oscillators [8]. Our method is based on chaotic synchronization using the open-plus-closed-loop (OPCL) coupling recently devised by Grosu and co-workers [9]–[13]. In our method, a message is treated as part of the parameter mismatch between the chaotic oscillators and added to a state variable of the chaotic oscillator of the drive system as dynamical noise. In this manner, the time-integrated message is encrypted by the chaotic signal. In the response system, the message is retrieved by subtracting the chaotic carrying signal reproduced by the OPCL control method from the received signal, followed by differentiation with respect to time. The parameter mismatch as well as the parameter of the chaotic oscillators are used as the keys to synchronize the chaotic oscillators of the drive and response systems, which may improve the security of communications.

In this paper, we demonstrate our experiment for the secure communications of an actual speech signal to test the utility of our method. In section 2, we provide a short summary of the mathematics of our method. In section 3, we show the procedure and experimental system for our experiment and show results. In sections 4 and 5, we discuss our results and make conclusions.

2. Theory

The main points of the OPCL control method are as follows. For details, see [9, 10]. Let us consider making $y(t) \in \mathbb{R}^d$ synchronous with $x(t) \in \mathbb{R}^d$ whose time evolutions are governed by $\dot{x} = F(x) + \Delta F(x)$ and $\dot{y} = F(y) + C_1(x) + C_2(x, y)$, where $y(t) - x(t) \to 0$ as $t \to \infty$. $\Delta F$ denotes parameter mismatch. The open-loop coupling $C_1$ and the closed-loop coupling $C_2$ are given as $C_1(x) = \dot{x} - F(x)$ and $C_2(x, y) = -$.
\[ [H - DF(x)](y - x), \] respectively, where \( DF(x) \) is the Jacobian matrix \((\in M_d)\) of \( F \) evaluated at \( x \) and \( H \((\in M_d)\) is a Hurwitz matrix whose eigenvalues have a negative real part. Complete synchronization of \( y \) with \( x \) is realized when introducing both \( C_1 \) and \( C_2 \). Let us assume that \( x \) is close to \( y \) as a consequence of the open-loop coupling \( C_1 \). Then, \( F(y) \) is written as \( F(y) \approx F(x) + DF(x)(y - x) \) using Taylor’s expansion and the time evolution of the difference \( e = y - x \) is subject to \( \dot{e} = H(y - x) = He \). The negative real part of the eigenvalues of \( H \) leads to \( e \to 0 \) as \( t \to \infty \), despite the parameter mismatch \( \Delta F(x) \). The synchronization of \( y \) with \( x \) is thus achieved irrespectively of the conditional Lyapunov exponent.

We next show a concrete procedure for our method on the basis of the OPCL control method using chaotic oscillators subject to the Sprott equations \([10, 14]\). Let us express the state variables of a drive system as \( x_1, x_2, x_3 \in \mathbb{R} \) and the corresponding variables of a response system as \( y_1, y_2, y_3, y_4 \in \mathbb{R} \). A message signal \( m(t) \in \mathbb{R} \) is encrypted using

\[
\begin{align*}
\dot{x}_1 &= -(k + \Delta k)(x_2 + m), \\
\dot{x}_2 &= x_1 + x_3, \\
\dot{x}_3 &= x_1 + (x_2 + m)^2 - x_3,
\end{align*}
\] (1)-(3)

where \( k \) is a parameter and \( \Delta k \) is parameter mismatch. The message is continuously input into the equations as dynamical noise to \( x_2 \) as if \( m \) were part of the parameter mismatch. Thus, \( m \) is mixed into the carrying signal \( x_2 \) and numerically integrated. We make the magnitude of \( m \) sufficiently smaller than that of the carrying signal, i.e., \( \| m \| \ll \| x_2 \| \), not to seriously deform the message. The response system receives the \( x_2 \) signal that contains the integrated message. The message can be decrypted using

\[
\begin{align*}
\dot{y}_1 &= -ky_2 - \Delta kx_2, \\
\dot{y}_2 &= y_1 + y_3, \\
\dot{y}_3 &= y_1 + y_2^2 - y_3 - (1 + 2x_2)(y_2 - x_2),
\end{align*}
\] (4)-(6)

where we use the Hurwitz matrix \( H \) given in \([10]\).

\[
H = \begin{pmatrix} 0 & -k & 0 \\ 1 & 0 & 1 \\ 1 & -1 & -1 \end{pmatrix}
\] (7)

Because \( y_2 \) is synchronized with the chaotic component of the received signal, subtraction of \( y_2 \) from \( x_2 \) and the subsequent differentiation of \( x_2 - y_2 \) with respect to time enable the retrieval of the original message \( m \). The parameter mismatch \( \Delta k \) must be exactly known to achieve the synchronization of the chaotic oscillator in the response system with that in the drive system. Accordingly, \( \Delta k \) is a key for decrypting the message.

3. Numerical Experiments

We conducted an experiment to test our idea. In this experiment, a speech signal “Yes, we can.” articulated by one of the authors (Y.Y.) was used as a message. Our experimental procedures are as follows. The speech signal was acquired in the WAV format using a microphone of high quality (Shure SM81) and a sound processing system (Yamaha MW8CX) under a sampling frequency of 11.025 [kHz] and a quantization precision of 8 bits. The file format of the message was converted to the TEXT format. Subsequently, the numerical calculation for the encryption and decryption of the message was performed using the method described in the preceding section on a personal computer. The file format of the message after encryption and decryption was again converted to the WAV format. Then, we listened to the original, encrypted and decrypted messages with a speaker system and compared their sound qualities to assess the performance of our method for secure communications.

The parameter and parameter mismatch of the Sprott oscillators were set to \( k = 0.225 \) and \( \Delta k = 0.0225 \) (10 % parameter mismatch), respectively. In the encryption and decryption of the message, the fourth-order Runge-Kutta method was applied to numerically integrate the equations with the time width equivalent to 11.025 [kHz]. After discarding the initial transient part of \( x_1, x_2, x_3 \), the speech signal was added to \( x_2 \) as dynamical noise in the drive system. Then, the \( x_2 \) signal containing the speech signal was transmitted to the response system. The maximum amplitude of the speech signal relative to that of the carrying signal was adjusted to be \( \| m \| / \| x_2 \| \approx 10^{-4} \).

Figure 1 displays the original speech signal “Yes, we can”. The chaotic signal \( x_2 \) containing the time-integrated message generated by the drive system and the chaotic signal \( y_2 \) retrieved using the OPCL control method by the response system are shown in Figs. 2 and 3, respectively. There was no discernible difference in between \( x_2 \) containing the message and \( y_2 \). That is, the encrypted signal sounded like noise containing no audible message. The message decrypted by the response system is shown in Fig. 4. For comparison, we show the same part of the speech signal before and after encryption and decryption in Figs. 5 and 6, respectively. The decrypted message is substantially similar to the original message shown in Fig. 1, although it sounded slightly noisy comparing with the original message. These observations indicate that our method for secure communications works as expected.
4. Discussion

In our method, the message is treated as dynamical noise added to a state variable of the chaotic oscillator of the drive system when encrypting a message. The variable should be selected in terms of the exclusiveness of a parameter mismatch in multiplying it with the variable. Thus, the message is viewed as part of the parameter mismatch. This is in contrast to Cuomo-Oppenheim’s method in which a message is superimposed as additive (observational) noise on a carrying signal when transmitted to the response system. The message is decrypted using chaotic synchronization by OPCL coupling between the chaotic oscillators in drive and response systems having parameter mismatch. The parameter mismatch is an indispensable key for decryption. The response system cannot retrieve the message from the received signal without knowledge of the parameter mismatch. When the chaotic oscillators have multiple parameters and parameter mismatches, we can achieve multiplex encryption and decryption of multiple messages. This has recently been shown using chaotic oscillators subject to the Lorenz equations by the authors [8].

Another benefit of our method is also brought about by the use of OPCL coupling. It makes our method free of the negativity of the conditional Lyapunov exponents indispensable for the stability of the synchronization manifold. In fact, we have not considered the conditional Lyapunov exponents in the present experiment using the Sprott oscillators. This allows us a wide selection of the dynamics governing the chaotic oscillators installed on drive and response systems.

5. Conclusion

We have shown the applicability of the OPCL control method to chaos-based communications through our experiment for the encryption and decryption of the speech signal. Our method provides two benefits that
are absent in Cuomo-Oppenheim-Strogatz’s method. One is freedom from negativity of the conditional Lyapunov exponents in selecting the dynamics governing the chaotic oscillators. The other is the use of parameter mismatch between the oscillators as the key for encryption and decryption. However, our recent work has revealed a weak point of our method that a message cannot be entirely masked by a carrying signal in the frequency domain [8], which remains to be studied as an open problem.

Acknowledgment

We appreciate Mr. Shuji Nakamura and Mr. Tatsuya Kunioka for their technical support.

References


A constructive approach for the design of finite time self-synchronizing coupled systems with unknown inputs

Gilles Millérioux†

†Nancy University
Research Center for Automatic Control of Nancy (CRAN UMR 7039)
2 rue Jean Lamour, 54519 Vandoeuvre-les-Nancy, France
Email: gilles.millerioux@esstinn.uhp-nancy.fr

Abstract—In this paper, we are interested in the problem of synchronization of coupled dynamical systems. The coupling under consideration is unidirectional and corresponds to a drive-response configuration. The drive system is supposed to be subjected to unknown inputs. It is provided a systematic methodology for selecting suitable drive variables and for designing an appropriate response system so that a finite time self-synchronizing is achieved. The approach is based on the notion of flatness, a notion borrowed from control theory.

1. Introduction

Driving a dynamical system has been an important subject of research for a long time. Driving a system by another means that both systems are coupled so that the behavior of the second one is dependent on the behavior of the first one but the converse does not hold. The first system is called the drive system while the second one is called the response. The driving is often referred to as unidirectional coupling and distinguishes from the bidirectional coupling. The coupling is made through the drive variables which consist of one or several outputs variables of the drive system. In the years 1990, works of Hubler [6] have been shown that driving systems with aperiodic signals could induce some interesting behaviors like nonlinear resonances or stimulation of particular modes. The idea has been extended to chaotic signals and originates from the pioneering works of Pecora and Carroll [5].

Among numerous definitions of synchronization (see [2] for an complete list), self-synchronization in a drive-response configuration has drawn much attention. By self-synchronization, it is meant an identical behavior of the drive and the response which is achieved without any external control. The main issue in self-synchronization is not only the selection of appropriate outputs of the drive to guarantee a given convergence behavior as asymptotical, finite-time, robustness against parameter mismatch or disturbances, as well as the design of a suitable structure for the response. Several examples and different situations can be borrowed from the engineering area. For instance, we can be interested in synchronizing two oscillators for communication purpose. The drive may consist of a modulator in a communication setup while the response may be a PLL (Phase Locked-Loop). In such a case, the drive signal is imposed while the structure of the response must be suitable designed to guarantee the phase synchronization with good filtering properties. Another example concerns symmetric cryptography. In such a context, the drive consists of the generator delivering a complex sequence used to conceal the information called the plaintext. The response consists of the decipher which not only must be synchronized in finite-time with the drive but also must be designed so that the plaintext can be properly recovered. The drive variable is nothing else but the ciphertext which is conveyed through the public channel. For a typical class of ciphers, the synchronization must be guaranteed without external control on the decipher. Indeed synchronization flags may be forbidden for throughput purpose. In other words, finite-time self-synchronization must be ensured.

Numerous techniques proposed so far in the literature to guarantee self-synchronization of autonomous dynamical systems resort to state reconstruction approaches involving for example observers. In such a case, the corresponding required property to guarantee synchronization is observability. A more complex self-synchronization issue arises when the drive system is non autonomous, that is forced by an input, and when such an input is unknown to the response. In such a case, a so-called unknown input observer must be used. It is the typical situation encountered in the aforementioned symmetric ciphers where the plaintext plays the role of the unknown input. It is also the case when a drive system is subjected to unknown disturbances. It turns out that unknown input finite-time self-synchronization is an issue which has not been deeply addressed.

In this paper, we propose a methodology leading to a systematic and constructive design of finite time self-synchronizing coupled systems with unknown inputs. Both issues, namely, the selection of suitable drive (output) variables and the design of the response system, are investigated. Analysis approaches have already been
suggested in the literature. By analysis, it is meant that given a specific output, it is possible to check whether the finite-time self-synchronization can be achieved. For example, a condition which applies for switched discrete-time systems has been proposed in [4]. On the other hand, design purpose, that is the issue of selecting a priori suitable outputs to achieve finite-time self-synchronization is a much more intricate problem. Actually it is an open problem in the general case. Very few works have addressed such an issue. See the work [3] for an exception dealing with continuous linear systems and a polynomial matrices-based approach. Based on the notion of flatness, a notion borrowed from control theory, we propose here a state space approach for discrete-time linear systems with the hope that an extension can be carried out for some classes of nonlinear systems.

The outline of this paper is the following. In Section 2, the problem of finite-time self-synchronization is stated in the general case. In Section 3 the problems of the selection of appropriate drive variable and of the design of the response which must ensure a finite-time self-synchronization are solved for discrete-time linear systems. An illustrative example is provided in Section 4. Finally Section 5 is devoted to some concluding remarks addressing the possible extension to nonlinear systems.

2. Problem statement in the general case

We are interested in a drive-response setup where the drive part is described by

\[
\begin{align*}
  x_{k+1} &= f(x_k, m_k) \\
  y_k &= h(x_k, m_k)
\end{align*}
\]

(1)

where \( f \) is the input function, \( f \) is the state-transition function, \( h \) is the output function and \( y_k \) is the drive (output) variable ensuring the coupling with the response. The response part admits the following generic equations

\[
\begin{align*}
  \hat{x}_{k+r+1} &= \tilde{f}(\hat{x}_{k+r}, y_{k+r}, \ldots, y_{k+r'}) \\
  \hat{m}_{k+r} &= \tilde{h}(\hat{x}_{k+r}, y_{k+r}, \ldots, y_{k+r'})
\end{align*}
\]

(2)

where \( r \) and \( r' \) are integers and where \( \tilde{h} \) must have the following property:

\[
\hat{m}_{k+r} = m_k \text{ if } \hat{x}_{k+r} = x_k
\]

(3)

where \( U \) is a non empty set of initial conditions and \( \| \cdot \| \) denotes the Euclidean norm.

Firstly, since the coupling is only unidirectional - from the drive to the response - \( x_k \) cannot depend on \( \hat{x}_k \). As a result, for all \( k > k_f \), when (4) applies, that is when \( x_k \) and \( \hat{x}_{k+r} \) are equal after a finite number of iterations, the consideration of (2) leads to the fact that \( x_k \) and \( \hat{x}_k \), up to a delay \( r \), are both expressed as a function, denoted \( F \), which depends exclusively on a finite number of delayed outputs \( y_k \), that is

\[
x_k = \hat{x}_{k+r} = F(y_{k-M}, \ldots, y_{k-M'}) \quad \forall k > k_f
\]

(5)

where \( M \) and \( M' \) are integers.

Besides, after substituting the expression of \( \hat{x}_{k+r} \) into the second equation of (2) and taking into account (3), it turns out that \( m_k \) and \( \hat{m}_{k+r} \) are equal and both of them can be expressed as a function, denoted \( G \), which depends also exclusively on a finite number of delayed outputs \( y_k \), that is

\[
m_k = \hat{m}_{k+r} = G(y_{k-N}, \ldots, y_{k-N'}) \quad \forall k > k_f
\]

(6)

where \( N \) and \( N' \) are integers. Let us point out that (6) is nothing else but the input/output model of (1). Such a relation provides a way to recover the unknown input \( m_k \).

The property that the state vector \( x_k \) and the input \( m_k \) of the dynamical system (1) can be expressed exclusively as a function of the delayed outputs is called flatness. For more details about flatness, the reader may refer to [1]. It is important stressing that, likewise observability, all dynamical systems haven’t got this property. The output \( y_k \) corresponding to a suitable function \( h \) which yields the relations (5) for \( x_k \) and (6) for \( m_k \) is called the flat output.

Remark 1 The relation (5) reflects that a flat system is necessary observable insofar as the state vector \( x_k \) is expressed by means of a function of the output only.

Equation (5) and (6) allow to rewrite the response (2) in the strictly equivalent form whenever \( k > k_f \)

\[
\begin{align*}
  \hat{x}_{k+r} &= F(y_{k-M}, \ldots, y_{k-M'}) \\
  \hat{m}_{k+r} &= G(y_{k-N}, \ldots, y_{k-N'})
\end{align*}
\]

(7)

The previous developments allow us to state the following proposition.

Proposition 1 If a dynamical system at the drive side is flat, it is always possible to select an output \( y_k \) called flat output and to design a response system so that, not only a self-synchronization in finite time is achieved, but also so that the unknown input \( m_k \) of (1) can be recovered in finite time. The equations describing the response are given by (7) or by (2) for a recursive equivalent form.

The purpose of this paper is to provide a systematic methodology based on a state space approach to select the flat outputs and to design the response system for the special class of discrete-time linear systems with the hope that an extension can be carried out for some classes of nonlinear systems.

Definition 1 A finite time self-synchronization fulfills

\[
\exists k_f < \infty, \forall k_0 \in U, \forall k > k_f \quad \text{and} \quad \forall m_k \| x_k - \hat{x}_{k+r} \| = 0
\]

(4)
3. Main result

3.1. Background on control theory

Throughout this section, when classical linear control theory results are mentioned, proofs are not incorporated. Let us consider the state space representation of a Single Input Single Output linear system:

\[
\begin{align*}
    x_{k+1} &= Ax_k + Bm_k \\
    y_k &= Cx_k + Dm_k
\end{align*}
\]

(8)

with \( x_k \in \mathbb{R}^n \), \( m_k \in \mathbb{R} \) and \( y_k \in \mathbb{R} \).

The corresponding input/output model of (8) reads:

\[
y_{k+n} + \ldots + a_1y_{k+1} + a_0y_k = \beta_n m_{k+n} + \ldots + \beta_1 m_{k+1} + \beta_0 m_k
\]

(9)

where the \( a_i \)'s are the coefficients of the characteristic polynomial of \( A \) which is by definition \( \varphi(A) = \det(A - \lambda I) \) (\( I \) stands for the identity matrix of dimension \( n \) and \( \det \) is the determinant). Equation (9) can be obtained by working out the transfer function \( H(z) = Y(z)/M(z) \) of (8) which is given by

\[
H(z) = C(zI - A)^{-1}B + D = \frac{\beta_n z^n + \ldots + \beta_1 z + \beta_0}{z^n + \ldots + a_1z + a_0}
\]

(10)

then considering \( z \) as the shift operator in the time domain.

**Proposition 2** The system (8) is observable if and only if rank \( Q_o = n \) with

\[
Q_o = [CA^T \ldots (CA^{n-1})^T]^T
\]

**Proposition 3** [1] The system (8) is flat if and only if it controllable, that is, rank \( Q_c = n \) with

\[
Q_c = [B AB \ldots A^{n-1}B]
\]

3.2. Selecting a flat output for the drive

We wish to state a condition on the space space model (8) which guarantees that \( y_k \) is a flat output.

**Proposition 4** The output \( y_k \) of (8) is flat whenever the pair \((C,D)\) fulfills

\[
D = D^*, \; C = C^* T^{-1} \quad \text{with} \quad T \quad \text{solution of}
\]

\[
\begin{align*}
    TA^* &= AT \\
    B &= TB^*
\end{align*}
\]

with

\[
A^* = \begin{bmatrix}
    -a_{n-1} & 1 & 0 & \cdots & 0 \\
    -a_{n-2} & 0 & 1 & \cdots & 0 \\
    \vdots & \vdots & \ddots & \vdots & \vdots \\
    -a_1 & 0 & 0 & \cdots & 1 \\
    -a_0 & 0 & 0 & \cdots & 0
  \end{bmatrix}, \quad
B^* = \begin{bmatrix}
    \beta_{n-1} - \beta_n a_{n-1} \\
    \vdots \\
    \beta_i - \beta_n a_i \\
    \vdots \\
    \beta_0 - \beta_n a_0
  \end{bmatrix}
\]

\[
C^* = \begin{bmatrix}
    1 & 0 & \cdots & 0 & 0
  \end{bmatrix}, \quad
D^* = \beta_n
\]

(12)

and all the \( \beta_i \)'s are zero but one denoted \( \beta_p \), \( p \in \{0, \ldots, n\} \).

**Proof**

Firstly, the observable canonical form

\[
\begin{align*}
x_{k+1} &= A^* x_k + B^* m_k \\
y_k &= C^* x_k + D^* m_k
\end{align*}
\]

(13)

with \( A^*, B^*, C^* \) and \( D^* \) defined by (12) and the state space representation (8), related one another according to

\[
A^* = T^{-1}AT \\
B^* = T^{-1}B \\
C^* = CT \\
D^* = D
\]

(14)

from which (11) are deduced, have the same input/output model (9). \( T \) is the similarity transform matrix and is invertible by definition. To prove such a correspondence, it suffices to work out the transfer function \( H'(z) = C'(zI - A')^{-1}B' + D' \) to realize that is the same as (10) and then considering again the variable \( z \) as a shift operator.

Since it is assumed that only one term \( \beta_i \) with \( i = p \) (\( p \in \{0, \ldots, n\} \)) is different from zero, (9) reduces to

\[
\sum_{j=0}^{p-1} a_j y_{k+j} + y_{k+n} = \beta_p m_{k+p}
\]

(15)

Consequently, the first condition for a flat output, that is the input must be expressed as a finite number of delayed outputs \( y_k \), is fulfilled.

Secondly, let us iterate (8) and lump together the iterates in the following matrix form

\[
\begin{bmatrix}
y_k \\
y_{k+1} \\
\vdots \\
y_{k+n-1}
\end{bmatrix} - \Gamma 
\begin{bmatrix}
m_k \\
m_{k+1} \\
\vdots \\
m_{k+n-1}
\end{bmatrix} = Q_o x_k = 0
\]

(16)

with

\[
\Gamma = \begin{bmatrix}
    D & 0 & \ldots & 0 \\
    CB & D & \ldots & 0 \\
    \vdots & \vdots & \ddots & \vdots \\
    CA^{n-1}B & \ldots & \ldots & D
  \end{bmatrix}, \quad
Q_o = \begin{bmatrix}
    C \\
    CA \\
    \vdots \\
    CA^{n-1}
  \end{bmatrix}
\]

Notice that the matrix \( Q_o \) is invertible since by construction, (8) is observable and precisely, \( Q_o \) is the observability matrix of (8). As a result, one gets

\[
x_k = Q_o^{-1} \begin{bmatrix}
y_k \\
y_{k+1} \\
\vdots \\
y_{k+n-1}
\end{bmatrix} - \Gamma 
\begin{bmatrix}
m_k \\
m_{k+1} \\
\vdots \\
m_{k+n-1}
\end{bmatrix}
\]

(17)

Since \( m_k \) and its iterates depend exclusively on a finite number of delayed outputs \( y_k \) regarding (15), so does the state vector \( x_k \) of (17). As a consequence, the second
condition required for a flat output is fulfilled. That completes the proof.

Based on Proposition 1, several important remarks can be made.

Remark 2 The tractability of this result lies in that the unknown $T$, which enables to compute $C$, can be easily obtained since (11) are mere linear matrix equalities to be solved.

Remark 3 Given $A^\ast$ and $B^\ast$, the solution $T$ of (11) is not unique.

Remark 4 The whole uncountable set of flat outputs of (8) corresponds to the set of all triplets $(p, \beta_p, T)$ with $p \in \{1, \ldots, n\}, \beta_p \in \mathbb{R}$ and $T$ solutions of (11).

3.3. Design of the response

On one hand, from (15), we infer that the function $G$ of (7) fulfills:

\[
\dot{m}_{k+r} = m_k = \beta_p^{-1} \cdot (\sum_{j=0}^{n-1} a_{j} y_{k-j} + y_{k-n})
\]

where $r$ is the delay introduced for causality sake. Next, substituting $m_k$ given by (18) and its iterates into (17) and replacing $x_k$ by $\dot{x}_{k+r}$ gives explicitly the function $F$ of (7).

4. Example

We consider the following drive system with state space model

\[
\begin{align*}
\dot{x}_{k+1} &= \begin{bmatrix} 1 & 2 \\ -1 & 3 \end{bmatrix} x_k + \begin{bmatrix} 1 \\ -1 \end{bmatrix} m_k \\
y_k &= C x_k + D m_k
\end{align*}
\]

We aim to find an appropriate output $y_k$ and so suitable matrices $C$ and $D$ as well as designing a response system so that the resulting drive-response configuration has the finite-time self-synchronization property.

The controllability matrix $Q_c$ is

\[
Q_c = \begin{bmatrix} 1 & -1 \\ -1 & -4 \end{bmatrix}
\]

The system is controllable since rank $Q_c = n = 2$. According to Proposition 3, it is flat. As a result, we can check for a flat output $y_k$. The characteristic polynomial of $A$ is $\phi_l(A) = det(AI - A) = A^2 - 4A + 5$. Thus, the matrix $A^\ast$ of the observable canonical form reads

\[
A^\ast = \begin{bmatrix} 4 & 1 \\ -5 & 0 \end{bmatrix}
\]

We set arbitrarily $p = 1$ and $\beta_1 = 1$. It is recalled that according to Proposition 4, all the other coefficients $\beta_i$ with $i \neq p$, here $\beta_0$ and $\beta_2$, must be zero. Thus $B^\ast$ reads

\[
B^\ast = \begin{bmatrix} 0 \\
0
\end{bmatrix}
\]

To find out $C$ and $D$ which ensures a flat output, we must consequently solve (11). One gets

\[
T = \begin{bmatrix} 1 & 0 \\ -1 & 0 \end{bmatrix}, \quad C = [0, -1], \quad D = 0
\]

Following Section 3.3, the corresponding response system which ensures a finite-time self-synchronization with $C$ and $D$ previously obtained yields, after basic matrices manipulations:

\[
\dot{m}_{k+1} = y_{k+1} - 4y_k + 5y_{k-1}
\]

\[
\dot{x}_{k+1} = \begin{bmatrix} y_k - 5y_{k-1} \\ -y_k
\end{bmatrix}
\]

Let us notice that a delay $r = 1$ has been introduced for causality sake.

5. Concluding remarks

In this paper, we have provided a systematic methodology for achieving a finite time self-synchronization between two unidirectional coupled systems. Both issues, namely the selection of suitable drive variables and the design of an appropriate response system, have been addressed. The approach is based on state space models and the notion of flatness. Whether the method can be extended to nonlinear systems is an interesting but difficult matter. Indeed, the key idea of the present paper lies in that we are able to find out an equivalence between two objects: a general state space model on one hand and a canonical state space model on the other hand. Owing to the equivalence, the property of flatness for the canonical form induces the same property for the general state space model. The trick lies in that characterizing flatness for the canonical form is straightforward. As a result, if we wish to extend the approach for nonlinear systems, we must find out canonical forms, also called normal forms, characterizing flat systems. Such an issue deserves further works.

References


CA-based Stream Cryptography with Variable-Length Key

Masaki Igarashi, Masayuki Ikebe and Junichi Motohisa

Graduate School of Information Science and Technology, Hokaido University
14 Nishi, 9 Kita, Sapporo, 060-0814 Japan
Email: igarashi@impulse.ist.hokudai.ac.jp, ikebe, motohisa@ist.hokudai.ac.jp

Abstract—Cellular automata (CA) stream cryptography with a variable-length key was developed. CA are used to design secret key cryptography systems based on a one-time pad cipher. To change the connection of cells of one-dimensional CA which generate pseudo-random number sequences (PNS), the state of the cells can be shifted and the key length of the cryptography can also be controlled. Arranging combinations and mixing CA-chaos sequences in time and space conceals a feature of the CA-chaos sequences, and their frequency characteristics are similar to those of white noise. Moreover, pseudo-random-numbers are extracted with consideration of spatial direction and the random numbers are selected by the internal state of the cipher system for internal-state-recovery attacks. The cipher system was simulated, and the encryption/decryption throughput was 64 bit/cycle.

1. Introduction

In a chaos-stream cipher, a chaos generator combining a one-way function and is used as a random number generator [1]. Usually, a logistic map is used for the chaos generator and the chaos value is binalized as a one-way function. However, a symbolic dynamics attack [2], which analyzes a chaos generating mechanism from the long PNS, is proposed for this type of chaos cipher.

In Cellular Automata is used as random sequence generator [3]. Gutowitz uses Cellular Automata as discrete dynamical system to add complexity of the cryptosystem [4]. But none of these schemes has been able to withstand the modern attacks developed out of the cryptanalysis techniques. Rules of radius \( r = 1 \) and 2 for non-uniform 1-D CA have been also proposed for high quality random numbers generation [5]. However, peculiar patterns with the CA rules appear near the cell. Here, we proposed combining and mixing the CA chaos sequences to conceal the peculiar patterns, and developed a cipher system using one-dimensional CA with a variable-length key.

Figure 1 is a block diagram of a cellular automaton cipher system. It consists of a converter with a one-way function machine and a cellular automaton chaos generator corresponding to a variable-length key. Details are given below.

2. Variable-Length Key of CA Cryptography

Here, we describe our method of creating the variable-length key by changing connections between cells. In conventional CA cell connections, an initial value is given to the one-dimensional cell array connected as a ring, and the cell array generates chaos by a appropriate rule. Here, we consider making the connection of the cells variable.

A bypass is added to the cell array with connection as a cross point, and it is made the shape of a small ring (Fig. 2). When the same initial values were given to the conventional cellular connection and the connection with a shifted cross point, different types of chaos were generated. In this way, a variable-length key can be made by shifting the knot of a bypass (the key length bit width is determined by the position the knot). Moreover, since the number of cells is not changed, the width of the obtained binary sequence does not change, and it becomes possible to conceal the composition of the chaos-generating mechanism from the outside.

Figure 2: Implementation of variable-length key on the CA.
3. Proposed Random Number Generator with CA Chaos

A chaos-generating mechanism is described by temporal differentiation, which represents correlation of a time series. This is applied also to cellular automata. Correlation of a time series is also represented for cellular automata by the cell-interaction rule. Therefore, when the generated chaos value is used as it is, there is a weakness that the initial chaos-generating value will be found from a correlation of a time series. Particular patterns of 1-D CA also become the weakness for searching internal state. Here, a converter that took safety into consideration is realized using composition and a mixture of chaos and XORing.

The following three elements are applied to our random number generator.

- Two chaos sequences are combined.
- Mixing the chaos sequences on the time-space and compressing them.
- Generating two binary sequences from the compressed chaos and selecting one sequence at the time of generation of a cryptogram.

First, two one-dimensional cell arrays are prepared and each bit of two generated chaos sequences are compounded by XORing (Fig. 3 and Eqs. 1-2). Here, we used rule[30 and 86]. CA-state $C^t_i$ consists of a set of each cell state $c^t_{i,k}$, $t$ and $k$ denote cell transition and cell position.

$$C^t_{30} = \{c_{30,0}^t, c_{30,1}^t, \ldots, c_{30,64}^t\}$$

$$C^t_{36} = \{c_{36,0}^t, c_{36,1}^t, \ldots, c_{36,64}^t\}$$

$$R^t_0 = C^t_{30} \oplus C^t_{36} = \{c_{30,0}^t \oplus c_{36,0}^t, \ldots, c_{30,64}^t \oplus c_{36,64}^t\}$$

Next, a synthetic chaos sequence is arranged in a form suitable for mixing and compressing on a time axis (Fig. 4 and Eq. 3). For 128-bit width, the chaos sequence bends at 90 degrees and is generated with a length of 64 bits so that the length and the width become 64 bits. Then, the 128x64-bit sequence is separately overlapped with the two sequences of 64x64 bits. Each bit of these overlapping sequences is combined by XORing. By this conversion, a 128x64-bit series compresses to 64x64 bits.

$$R^t_1 = \begin{pmatrix}
    r_{0,63}^t & \cdots & r_{0,63}^{t+63} \\
    \vdots & \ddots & \vdots \\
    r_{0,63}^t & \cdots & r_{0,63}^{t+63}
\end{pmatrix}
\oplus
\begin{pmatrix}
    r_{0,64}^t & \cdots & r_{0,127}^t \\
    \vdots & \ddots & \vdots \\
    r_{0,64}^t & \cdots & r_{0,127}^t
\end{pmatrix}
\quad (3)
$$

Figure 5 and equations 4-5 show operation flow of the final stage processing. The generated 64x64-bit sequence $R_i^t$ is divided into two binary patterns. Two 64-bit patterns ($A^t$ and $B^t$) are extracted from each pattern by $Mask_A$ and $Mask_B$. Separately, from the 64x64-bit sequence $R^t_i$, 64 bits on a diagonal line are extracted and a parity operation is performed on the bits. From the results of the parity operation $Parity^t$, a 64-bit pattern is selected from $A^t$ and $B^t$ at the time of encryption. Since the 64 bits on the diagonal line and the result of the parity operation $Parity^t$ are concealed from the outside, it cannot be confirmed from the outside whether $A^t$ and $B^t$ was selected.

$$A^t = R^t_1 \cdot Mask_A$$

$$B^t = R^t_1 \cdot Mask_B$$

$$Parity^t = \begin{pmatrix}
    r_{1,0,0}^t \oplus r_{1,1,1}^t \oplus \cdots \oplus r_{1,63,63}^t \\
    O^t \oplus r_{1,0,0}^t \oplus r_{1,1,1}^t \oplus \cdots \oplus r_{1,63,63}^t
\end{pmatrix}$$

$$t = 0$$

$$t > 0$$

$$R^t = Sel(Parity^t, A^t, B^t)$$

$$Sel(Parity^t, A^t, B^t) = \begin{cases}
    A^t(Parity^t = 1) \\
    B^t(Parity^t = 0)
\end{cases}$$

-335-
Figure 5: Operation flow of PNS generation

Figure 6: Proposed cipher system

4. Operation of Our Cipher System

Proposed cipher system is shown in Fig. 6. After the key setup is completed, the random number generation starts. Under processing, a chaos sequence is output for every cycle. After 128 operation cycles are completed, a 64-bit random number is generated for every cycle. First 64 operation cycles are used for a stable chaos generation. The 64-bit source data are enciphered at each cycle. After 192 operation cycles, the previous 64-bit cryptogram unit is also used for selecting the 64-bit random number with the 64-bit-parity operation on the diagonal line of a 64x64-bit-chaos pattern, and the system continues generating 64-bit random numbers. The random number generation of a proposal cipher system also takes into consideration the previous 64-bit cryptogram (i.e., that used for 64-bit selection). Therefore, if the data enciphered differs even if it uses the same key, a different random number will be generated. In decryption, the data is recovered in the same procedure using the same instrument and same key as encryption. Specification of our system is shown in Table 1.

5. Simulation Results

The results are shown in Figs. 7 and 8. Figure 7 show Fourier spectrum of chaos patterns. In the conventional CA sequence (rule [86]), particular patterns are confirmed and its frequency characteristic has also particular ones. How-

Table 1: Specification of proposed cipher system

<table>
<thead>
<tr>
<th>Method</th>
<th>CA based chaos stream cipher</th>
</tr>
</thead>
<tbody>
<tr>
<td>CA form</td>
<td>one dimensional, uniformed</td>
</tr>
<tr>
<td>rules</td>
<td>30, 86</td>
</tr>
<tr>
<td>Length of key</td>
<td>$32 \leq K_{0x}, K_{1x}, K_{0y}, K_{1y} \leq 64$</td>
</tr>
<tr>
<td>Throuthput</td>
<td>64 bit/cycle</td>
</tr>
</tbody>
</table>
ever, in the combined CA sequence, particular ones are reduced considerably and its frequency characteristic is close to white noise. Moreover, in the mixed CA sequence, its characteristics are improved.

Figure 8 shows results of our system operation. We confirmed that the encryption/decryption were performed normally and the difference patterns were generated with the initial images, even if the same key was used.

6. Consideration of Cipher Attack

In the proposed cipher system, two chaos sequences are combined. Here, the separation problem of chaos can be given to attackers. Moreover, in mixing and compressing of the chaos sequence, the chaos-generated values generated at different times are used. Since the data mixture on space time is applied to our system, this operation reduces the correlation of PNS[6]. In extracting the random number, a number with spatial distance is extracted and the internal state is concealed by selecting a number using an internal state.

There are four types of attacks that are assumed to be possible against the proposed system:

- Symbol dynamics attack
- Output prediction attack
- Internal state recovery attack
- Key recovery attack

The symbol dynamics attack searches for the sequence based on time series correlation of the binalized chaos value. In this type of attack, the approximate value of a higher bit is predicted in the state before binalization, and a brute force attack is applied to the lower bit. However, since a chaos of CA does not have distinct higher and lower bits, this attack cannot predict the approximate chaos value. Moreover, the attack cannot predict the sequence without time series correlation. Our system is protected from this type of attack by negating the time series correlation.

The remaining three types of attacks have relativity each other, and in our system, unless the internal state recovers, output prediction and key recovery cannot be performed. Therefore, recovery of the internal state passes through three processes:

- Recovery of selected PNS
- Decision of variable-length key positions
- Decision of an internal state

Each process is interlocking and each factor of the process is concealed. Each one requires calculation of $2^{64}$, $2^{50}$ and $4^{4096}$ respectively. Therefore, if there is an internal state recovery attack against our system, computational complexity will increase more than it would with a brute force attack.

7. Conclusion

We proposed a method for CA-based stream cryptography with variable-length key and evaluated the cipher system. To change the connection of cells of one-dimensional CA, we controlled the key length of the cryptography. Arranging combinations and mixing CA-chaos sequences in time and space, we concealed a feature of the CA-chaos sequences, and we improved their frequency characteristics, which became similar to those of white noise. Moreover, using selection of PNSs are applied to our system, we protected the internal state of the system from the attacks. The cipher system was simulated, we confirmed that the encryption/decryption throughput was 64 bit/cycle.

References


Abstract—The progressive miniaturization of electronics components makes today feasible to exploit the nanoscale level. At this scale, phenomena peculiar of the quantum world arise, which can be exploited to realize innovative and potentially breakthrough devices. In this paper, the dynamic behavior of one such device, the Casimir nonlinear oscillator, is analyzed. It is shown that, due to the nonlinear nature of the Casimir force, the oscillator can exhibit Smale horseshoes, and the mechanism leading to their birth is revealed.

1. Introduction

The progressive reduction in size of electronic devices, that in about 50 years has covered the interval from centimeters to microns, is now undergoing a further dramatic descent, from micro to nano-scale. At this scale, the laws of physics are quantum mechanical in nature, and new amazing phenomena, which are unexpected from a classical perspective, emerge.

An important prediction of quantum electrodynamics (QED) is the existence of irreducible fluctuations of the electromagnetic field even in vacuum. These fluctuations are responsible of van der Waals forces between atoms, and of Casimir forces, i.e. interactions between electrically neutral and highly conductive metals [1].

The boundary conditions imposed on the electromagnetic fields by the presence of metallic surfaces lead to a spatial redistribution of the mode density with respect to free space, creating a spatial gradient of the zero-point energy density and hence a net force between the metals [2]. Apart from its intrinsic relevance from the point of view of theoretical physics, the Casimir effect has recently attracted considerable attention for its possible engineer applications. Because boundary conditions can be tailored, this raises the interesting possibility of designing QED forces for specific applications, exploiting the fascinating idea to use the vacuum\(^1\) as a device itself. In this optic, nano-electrometers, actuators, resonators and nonlinear oscillators have been realized and are under investigation [3, 4, 5, 6, 7].

The Casimir forces are inherently mesoscopic, since they can acquire significant values when the separation between the metallic surfaces is reduced to less than 100 nm, and nonlinear in nature. On the one hand, the mesoscopic nature allows a classical description of the dynamic behavior of the aforementioned devices. On the other hand, the nonlinear nature suggests the possible emergence of complex nonlinear behaviors. While there is vast experimental literature about hysteretic response and bistability of nonlinear oscillators in quantum optics, solid-state physics, mechanics, and electronics, it was only in [7] that the experimental observation of such phenomena caused by QED effects was given.

In this paper, we study the dynamical behavior of the Casimir nonlinear oscillator in the weakly damped, weakly forced regime. We show that, for some range of the parameters, the system has a homoclinic loop. By using the method of Melnikov, we prove that, under the effect of a periodic forcing, the system can exhibit transverse homoclinic orbits, and thus Smale horseshoes.

2. The Casimir nonlinear oscillator

A simple model of the Casimir oscillator is shown in figure 1. It is composed of a metallic plate (thick grey line), free to rotate about two torsional rods (black dot), subjects to the momentum generated by the nonlinear Casimir force, which arises from the interaction with a fixed metallic sphere of radius \(R\) placed at a distance \(z\). The oscillator is excited by the application of a voltage to an electrode fixed under the plate. The choice of the spherical shape for one of the interacting surfaces is justified to avoid alignment problems.

For this arrangement, the Casimir force takes the value

\[
F_C = \frac{\pi^3 h c R}{360 z^3}
\]

(1)

where \(h\) is the Planck constant/\(2\pi\), and \(c\) is the speed of light.

So far, we ignore the dissipation and the forcing. At the equilibrium distance \(z = d\), the momentum generated by the Casimir force \(M_C = F_C h\) is balanced by the restoring elastic torque \(M = -\alpha \theta\), inducing a rotation \(\theta = \theta_0\),

\[
-\alpha \theta_0 + \frac{\pi^3 h c R}{360 a^3} = 0
\]

(2)
where \( \alpha \) is the torsional spring constant. For small oscillations, \( z \sim d - d\theta \), and following [7], the Casimir force \( F_C(z) \) is Taylor expanded about \( d \) up to \( z^3 \). The potential energy of the system can be easily calculated giving

\[
V(\theta) = \frac{\alpha (\theta + \theta_0)^2}{2} - F_C(d) b \theta + \frac{F_C'(d)b^2}{2} \theta^2
\]

\[
- \frac{F_C''(d)b^3}{6} \theta^3 + \frac{F_C'''(d)b^4}{24} \theta^4
\]

(3)

where \( F_C(d), F_C'(d), F_C''(d), F_C'''(d) \) are the Casimir force, and its first, second and third derivatives evaluated at a distance \( d \), respectively. The potential \( V(\theta) \) has a local minimum in the origin and two local maxima at

\[
\theta_\pm = \frac{3}{F_C''(d)b^3} \left( \frac{F_C'''(d)b^4}{2} \right) \pm \sqrt{\left( \frac{F_C'''(d)b^4}{6} \right) \left( \alpha + F_C'(d)b^2 \right)}.
\]

(4)

and decreases unbounded for both \( \theta < \theta_- \) and \( \theta > \theta_+ \). Therefore the system has a neutrally stable equilibrium in the origin (a center), and two unstable equilibria at \( \theta_\pm \), which are of saddle type. The Lagrange equation of motion is

\[
\ddot{\theta} = \lambda \theta + \mu \theta^2 + \nu \theta^3
\]

(5)

where

\[
\lambda = \left[ \alpha_0 + \frac{F_C'(d)b^2}{I} \right]; \quad \mu = \frac{F_C''(d)b^3}{2I}; \quad \nu = \frac{F_C'''(d)b^4}{6I}.
\]

(6)

Here, \( I \) is the moment of inertia of the plate, and \( \omega_0^2 = \alpha/I \) is the fundamental frequency of the oscillator. To simplify eq. (5), we introduce a new variable \( \phi = \theta - \theta_- \), and rewrite (5) as a system of first order ODEs

\[
\begin{align*}
\dot{\phi} &= J \\
\dot{J} &= \nu \phi (\phi + \theta_+) (\phi + \theta_+ - \theta_-).
\end{align*}
\]

(7)

Then we introduce new parameters

\[
\rho = \nu(2\theta_+ - \theta_-); \quad \sigma = \nu \theta_+(\theta_+ - \theta_-)
\]

(8)

and perform a linear change of coordinates

\[
(\phi, J, t) \rightarrow \left( \sqrt{\frac{\sigma}{\rho}}, \sqrt{\frac{\sigma}{\rho}}, \frac{1}{\sqrt{\sigma}}t \right).
\]

(9)

which reduces eq. (7) to

\[
\begin{align*}
\dot{\phi} &= J \\
\dot{J} &= \phi^3 + \xi \phi^2 + \phi
\end{align*}
\]

(10)

where \( \xi = \rho/\sqrt{\sigma} \).

Eq. (10) presents the advantage to depend on the parameter \( \xi \), which adsorbs the three previously defined parameters \( \lambda, \mu, \nu \). Moreover, we have shifted one saddle to the origin, the other to \( \phi_+ \) and the center to \( \phi_- \), where

\[
\phi_\pm = \frac{1}{2} \left( -\xi \pm \sqrt{\xi^2 - 4} \right).
\]

(11)

3. Homoclinic Orbit

In this section we show that, for certain values of the parameter \( \xi \), system (10) possesses a homoclinic orbit through the origin, surrounding a region filled with periodic orbits. System (10) has the Hamiltonian

\[
H(\phi, J) = \frac{J^2}{2} - \frac{\phi^4}{4} - \frac{\xi \phi^3}{3} - \frac{\phi^2}{2}.
\]

(12)

the level sets \( H(\phi, J) = E \) define the trajectories of (10). For the orbit passing through the origin we have \( H(0,0) = 0 \), which implies

\[
J = \pm \sqrt{\phi^2 \left( \frac{\phi^2}{2} + \frac{2\xi \phi}{3} + 1 \right)}.
\]

(13)

This curve intersects the \( \phi \)-axis in three points, \( \phi = 0 \), and

\[
\phi_\pm = \frac{2 \xi}{3} \pm \frac{4 \xi^2}{9} \sqrt{\frac{9}{2} \phi - 2}.
\]

(14)

provided \( \xi > 3 \sqrt{2}/2 \). Introducing the positive determination of (13) in the first of (10) we obtain

\[
\phi = \sqrt{\phi^2 \left( \frac{\phi^2}{2} + \frac{2\xi \phi}{3} + 1 \right)}.
\]

(15)

By separation of variables, eq. (15) can be integrated from 0 to \( t \) in terms of elementary functions, since it has repeated roots. Choosing the initial conditions as \( (\phi(0), J(0)) = (\phi_+, 0) \) we have

\[
\phi(t) = \frac{3}{2\xi - 3 \sqrt{2} \tanh \frac{tK}{2}}\left( \tanh \frac{tK}{2} - 1 \right).
\]

(16)
where \( K = 2 \text{ atanh} \left( \frac{\sqrt{2}}{2} \phi \right) \). Computing the derivative and with some algebraic manipulations we obtain
\[
J(t) = \frac{12 \xi \sinh(t - K) - 18 \sqrt{2} \cosh(t - K)}{\left( 2 \xi + 2 \xi \cosh(t - K) - 3 \sqrt{2} \sinh(t - K) \right)^2}. \quad (17)
\]

It is readily seen that
\[
\lim_{t \to \pm \infty} (\phi(t), J(t)) = (0, 0)
\]
(18)

which matches the requirement for \((\phi(t), J(t))\) to be a homoclinic loop through the origin.

Next we observe that for \( \xi > 3 \sqrt{2}/2 \), eqs (11) and (14) imply \( \phi_+ < \phi_- \), that is, the center lies inside the region delimited by the homoclinic orbit. Since outside the interval between the two saddles the potential decreases unbounded, we conclude that the homoclinic orbit is the separatrix between a region filled of periodic orbits and a region characterized by unbounded trajectories.

4. Homoclinic chaos and Melnikov method

Now we include weak dissipation and periodic forcing. The equation of motion becomes
\[
\begin{cases}
\dot{\phi} = J \\
J = \phi^3 + \xi \phi^2 + \phi + \epsilon (A \cos \omega t - \gamma J),
\end{cases}
\]
(19)

where \( \epsilon \ll 1 \) takes into account the weakness of the perturbation, \( A \) and \( \omega \) are the amplitude and frequency of the forcing, respectively, and \( \gamma \) is the damping constant. Under the effect of the perturbation, the stable and unstable manifolds split, and may eventually intersect each other transversally, giving rise to transverse homoclinic orbits. The existence of such orbits implies, via the Smale–Birkhoff theorem, the presence of Smale horseshoes, and it is a signature of chaotic behavior [8].

These transversal intersections may be found by searching the simple zeros of the Melnikov function [8]. For the case under investigation the Melnikov function is given by
\[
M(t_0) = \int_{t_0}^{t_\infty} J(t) \left[ A \cos \omega (t + t_0) - \gamma J(t) \right] dt. \quad (20)
\]

We split this integrals into two parts. Using integration by parts
\[
\int_{t_0}^{t_\infty} J(t) \cos \omega (t + t_0) dt = \phi(t) \cos \omega (t + t_0) \bigg|_{t_0}^{t_\infty} + \omega \int_{t_0}^{t_\infty} \phi(t) \sin \omega (t + t_0) dt. \quad (21)
\]

It is easy to see that the first contribution is null, as \( \phi(t) \) goes to zero for \( t \to \pm \infty \). With the substitution \( x = (t - K)/2 \), and using the properties of hyperbolic functions, the second contribution becomes
\[
\omega \int_{-\infty}^{t_\infty} \phi(t) \sin \omega (t + t_0) dt =
6 \omega \int_{-\infty}^{t_\infty} \sin(2x + K + t_0) \frac{\sin(2x + K + t_0)}{\cosh x} dx. \quad (22)
\]

This integral can be solved by the method of residues, observing that we have regularly spaced simple poles at \( z = i(\pi/2 + k\pi) \) and \( z = \text{atanh}(3 \sqrt{2}/(2 \xi)) + i(\pi/2 + k\pi) \), and considering the integration path shown in figure 2.

![Figure 2: The path of integration for (22).](image)

By using Green theorem, the second part can be recast as a line integral, since it is not explicitly time dependent, obtaining
\[
\int_{t_0}^{t_\infty} J^2(t) dt = \int_{C(t_0)} J(\phi) d\phi = 2 \int_{\phi_0}^{\phi_1} \phi \sqrt{\frac{\phi^4}{2} + \frac{2 \phi}{3} + 1} d\phi, \quad (23)
\]
which can be solved by usual methods. The final results for \( \rho(0) \) is
\[
M(t_0) = -\gamma \rho + \frac{A}{\omega} \sin \left[ \omega \left( t_0 + K + \text{atanh} \left( \frac{3 \sqrt{2}}{2 \xi} \right) \right) \right] \quad (24)
\]
where
\[
\rho = \frac{4 - \xi^2}{9} + \frac{2 \xi(4 \xi^2 - 18)}{27 \sqrt{2}} \ln \frac{2 \xi + 3 \sqrt{2}}{\sqrt{4 \xi^2 - 18}} \quad (25)
\]
\[
\alpha = 2 \sqrt{2} \pi \omega \cosh (\omega \pi) \sin \left[ \omega \text{atanh} \left( \frac{3 \sqrt{2}}{2 \xi} \right) \right]. \quad (26)
\]

From (24), the Melnikov function has infinitely many zeros provided
\[
\sin \left[ \omega \left( t_0 + K + \text{atanh} \left( \frac{3 \sqrt{2}}{2 \xi} \right) \right) \right] = \frac{\rho \gamma}{\alpha A} \quad (27)
\]
These zeros are simple if \( \frac{dM(t_0)}{dt_0} \neq 0 \). For the derivative we have
\[
\frac{dM(t_0)}{dt_0} = \omega A \alpha \cos \left[ \omega \left( t_0 + K + \text{atanh} \left( \frac{3 \sqrt{2}}{2 \xi} \right) \right) \right]. \quad (28)
\]
A sufficient condition for \( \frac{dM(t_0)}{dt_0} \neq 0 \) when \( M(t_0) = 0 \) is
\[
-1 < \sin \left[ \omega \left( t_0 + K + \text{atanh} \left( \frac{3 \sqrt{2}}{2 \xi} \right) \right) \right] < 1 \quad (29)
\]
from which we finally derive the condition to have homoclinic tangencies
\[
\frac{A}{\gamma} > \frac{\rho}{\sigma}.
\] (30)

In figure 3 are shown the stable and unstable manifolds, obtained through numerical simulations, for \(\xi = 4, \epsilon \gamma = 0.1, \omega = 1\), and different values of \(\epsilon A\). The homoclinic loop is also shown (dashed line) for reference. For \(A\) less than the critical value, the manifolds are well apart. For \(A = 0.02126\), very close to the theoretical value \(A_c = 0.02\) obtained from (30), the first homoclinic tangency occurs. For higher values of \(A\) the manifolds intersect transversally.

Figure 3: Numerically obtained stable and unstable manifolds for different forcing amplitudes. Upper: \(A = 0.017\). Middle: \(A = 0.02126\). Lower: \(A = 0.025\).

Under the effect of the perturbations, we expect that the periodic trajectories inside the homoclinic loop undergo some kind of bifurcation. In particular, we expect the emergence of limit cycles from the center, with period multiple to that of the periodic forcing. These scenario is commonly known as subharmonic resonances [8], its analysis is left to a future work.

5. Conclusions

Recently, there has been a great amount of attention toward the possible application of QED effects in nano-electro–mechanical devices.

We have analyzed the dynamical behavior of one such apparatus, e.g. the Casimir nonlinear oscillator. Resorting to the Melnikov method, we have shown that, due to the nonlinear nature of Casimir force, when a periodic forcing is applied, the oscillator can exhibit transversal homoclinic intersections between stable and unstable manifolds. Via the Smale–Birkhoff theorem, this implies the existence of Smale’s horseshoes.

The importance of this result is twofold. On the one hand, it is relevant in view of possible applications of such oscillator. On the other hand, we have shown that chaotic behavior can arise in a practical device, due to a QED effect. To the best of our knowledge, this result is here reported for the first time.

Acknowledgments

The first author grateful thanks the Istituto Superiore Mario Boella and the regional government of Piedmont for financial support. This work was partially supported by the CRT Foundation.

References

Forced Chaos Generator with CMOS Variable Active Inductor Circuit

Yusuke Tsubaki†, Munehisa Sekikawa ‡, and Yoshihiko Horio†,‡

†Graduate School of Engineering, Tokyo Denki University, 2-2 Kanda-Nishiki-cho, Chiyoda-ku, Tokyo 101-8457, Japan
‡Aihara Innovative Mathematical Modelling Project, FIRST, JST,
IIS, the University of Tokyo, 4-6-1 Komaba, Meguro-ku, Tokyo 153-8505, Japan
E-mail: 09kme35@ms.dendai.ac.jp, sekikawa@sat.t.u-tokyo.ac.jp, horio@eee.dendai.ac.jp

Abstract—We propose a forced chaos generator with a CMOS variable active inductor circuit. The equivalent inductance of the variable active inductor in the proposed circuit can be controlled by an external voltage. The oscillation frequency of the circuit can be tuned by applying an external signal. We then realize the folding-and-stretching mechanism of chaotic motion by applying a periodic external signal. The chaotic dynamics are confirmed through SPICE simulations with TSMC 0.35 μm CMOS semiconductor process parameters. Moreover, we present bifurcation phenomena, which are generated when the amplitude and the period of the external signal are changed as bifurcation parameters.

1. Introduction

Chaotic phenomena observed in electronic circuits have been extensively investigated. It is well known that chaos, torus, and frequency entrainment occur in non-autonomous circuits for certain values of the amplitude and period of the external signal [1]. In addition, autonomous chaotic circuits such as the double scroll circuit [2],[3] and the hysteresis chaotic circuit [4] have also been proposed. Recently, researchers have attempted to apply chaos to real-world problems such as chaotic communication, chaotic encryption, and combinatorial optimization problems. For these applications, IC implementation is necessary because miniaturization, high-speed operation, and large-scale integration of the chaotic circuits are essential. Against this background, several chaotic circuits have been implemented as integrated circuits [5]-[7].

As one of these chaotic circuits, a three-dimensional autonomous chaotic circuit, based on a change in the oscillation frequency, has been proposed (Fig. 1) [8]. This circuit consists of a linear negative conductance, a capacitor C, two inductors L1 and L2, and a diode D. By replacing the diode D in the circuit with a current-controlled switch, the sub-circuit consisting of L1, L2, and D can be treated as a variable nonlinear inductor. The oscillation frequency of the circuit is changed by the switching operation of the diode. Therefore, this effect causes a stretching-and-folding mechanism, which is the basic mechanism of the chaos generation.

In this paper, we propose a forced chaos generator based on the circuit in Fig. 1 with a CMOS variable active inductor circuit. The equivalent inductance of the variable active inductor in the proposed circuit can be controlled by the external voltage. We change the oscillation frequency of the circuit by applying a periodic external signal to the variable active inductor. Based on the above principle, the above-mentioned folding-and-stretching mechanism is realized. We confirm the chaotic dynamics of the proposed circuit through SPICE simulations with TSMC 0.35 μm CMOS semiconductor process parameters. Moreover, we present bifurcation phenomena, which are generated when the amplitude and period of the external signal are changed as bifurcation parameters.

2. Forced Chaos Generator with CMOS Variable Active Inductor Circuit

Figure 2 shows the forced chaos generator with a CMOS variable active inductor circuit. First, we focus on the CMOS variable active inductor and omit C3 from the circuit in the figure.

We assume that C ≡ C1 = C2 and R ≡ R1 = R2.

Figure 1: Three-dimensional autonomous chaotic circuit proposed in Ref. [8].

Figure 2: Forced chaos generator with CMOS variable active inductor circuit.
We also assume that each of the two NMOSFETs and two PMOSFETs match perfectly, and operate on equal dc currents: that is, $g_{mn} \equiv g_{m1} = g_{m2}$, $g_{mp} \equiv g_{m3} = g_{m4}$, $g_{dsn} \equiv g_{ds1} = g_{ds2}$, and $g_{dsp} \equiv g_{ds3} = g_{ds4}$. The input admittance $Y_m$ of the CMOS variable active inductor is derived as

$$Y_m = G + jB,$$  \hspace{1cm} (1)

where

$$G = a_0 + \frac{C^2R_0\omega^2(\omega_1 + \omega_2\omega^2)}{b^2 + C^2\omega^2}$$  \hspace{1cm} (2)

and

$$B = \frac{a_3\omega + a_4\omega^3}{b^2 + C^2\omega^2}$$  \hspace{1cm} (3)

$a_0$, $a_1$, $a_2$, $a_3$, and $b$ are given in Table 1. When $G < 0$ and $B < 0$ in Eq. (1), the input admittance of the CMOS variable active inductor consists of negative conductance and inductive susceptance connected in parallel. In this case, the equivalent inductance $L_{eq}$ of the circuit is expressed as

$$L_{eq} = -\frac{1}{\omega B} = -\frac{b^2 + C^2\omega^2}{a_3\omega^2 + a_4\omega^3};$$  \hspace{1cm} (4)

Figure 3(a) shows the admittance chart of the CMOS variable active inductor circuit obtained from SPICE simulations with TSMC 0.35 $\mu$m CMOS process parameters. As shown in Fig. 3(a), the value of $L_{eq}$ can be controlled by the external voltage $V_{bias}$. Table 2 lists the circuit parameters used in the SPICE simulations.

In addition, a periodic solution can be generated by connecting $C_3$ to the CMOS variable active inductor circuit in parallel, because the CMOS variable active inductor circuit has negative conductance in a certain frequency range. The oscillation frequency $f_p$ of the periodic solution is almost equal to the resonance frequency $1/(2\pi\sqrt{L_{eq}C_3})$. Thus, we can change the oscillation frequency of the periodic solution by tuning $V_{bias}$. As obtained from the SPICE simulations, Fig. 3(b) shows the oscillation frequency of the periodic solution as a function of $V_{bias}$.

On the basis of the above-mentioned principle, we change the oscillation frequency of the circuit by applying a periodic external voltage, shown in Fig. 4 as $V_{bias}$. As a result, the forced chaos generator based on the circuit in Fig. 1 is realized.

### Table 1: Coefficients in Eqs. (2) and (3).

| \(a_0\) | \(g_{dsn} + g_{dsp}\) |
| \(a_1\) | \(C^2[1 + (a_0 + g_{mp})R][1 + R(a_0 + g_{mp} - g_{mn} + g_{mp})R]\) |
| \(a_2\) | \(C(a_0 + g_{mp})(a_0 + g_{mp} - g_{mn}g_{mp}R)\) |
| \(a_3\) | \(C^2[1 + 2(a_0 + g_{mp})R + (a_0 - g_{mn} + g_{mp})(a_0 + g_{mn} + g_{mp})R^2 - g_{mn}(a_0 + g_{mp})2g_{mn}g_{mp}R^2]\) |
| \(a_4\) | \(a_0 + g_{mp} + C^2R[1 + (a_0 + g_{mp})R]\omega^2\) |

### Table 2: Circuit parameters.

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$W/L$ of $M_1, M_2$</td>
<td>7.2/0.6 $\mu$m</td>
</tr>
<tr>
<td>$W/L$ of $M_3, M_4$</td>
<td>6.6/0.6 $\mu$m</td>
</tr>
<tr>
<td>$R_1, R_2$</td>
<td>5 k$\Omega$</td>
</tr>
<tr>
<td>$C_1, C_2$</td>
<td>3 pF</td>
</tr>
<tr>
<td>$C_3$</td>
<td>10 pF</td>
</tr>
<tr>
<td>$V_{DD}$</td>
<td>1.65 V</td>
</tr>
<tr>
<td>$V_{SS}$</td>
<td>-1.65 V</td>
</tr>
</tbody>
</table>

Figure 3: (a) Admittance chart of the CMOS variable active inductor circuit. (b) Oscillation frequency of periodic solutions as a function of $V_{bias}$.

Figure 4: External periodic signal for $V_{bias}$.
of the external periodic signal as shown in Fig. 6. From this figure, we can confirm that the attractor in Fig. 5 is chaotic, because a folding-and-stretching mechanism is observed.

Bifurcation phenomena are observed when the amplitude and period of the external signal are swept as bifurcation parameters. In the following, we define the Poincaré section at each rising edge of the external square waveform.

Figure 7 shows the observed attractors and their Poincaré sections. The amplitude parameters of the external signal are fixed at $V_{\text{bias}}^+ = -0.11 \, \text{V}$ and $V_{\text{bias}}^- = -0.72 \, \text{V}$. The period $T$ of the external signal is varied. Here, we define a number of points on the Poincaré section as the period. For example, we can see 5 points in Fig. 7(a.2); therefore, the period of the attractor in Fig. 7(a.1) is 5. Figure 8 shows a bifurcation diagram of the attractor on the Poincaré section when the period $T$ of the external signal is swept. As $T$ is increased, the period-5 orbit bifurcates and a period-10 orbit is generated, as shown in Fig 7(b). In Fig. 8, a period-doubling route to chaos can be seen for $72.5 < T < 75.5$. Unfortunately, the limited calculation precision of the SPICE simulations prevents solutions with periods greater than 20 from being observed.

Figure 9 shows the bifurcation phenomena when $V_{\text{bias}}^+$ of the external signal is changed (Fig. 4). In the figure, the horizontal axis and the vertical axis are $V_{\text{bias}}^+$ and $v$ of the attractors on the Poincaré sections, respectively. The period and lower voltage of the external signal are fixed at $T = 100 \, \text{ns}$ and $V_{\text{bias}}^- = -0.72 \, \text{V}$, respectively. As shown in Fig. 9, period-adding phenomena [9], [10] are observed.

To understand the properties of the complex attractors in the region between periodic solutions, we examine the attractors on the Poincaré sections at various phases of the external signal. Between $V_{\text{bias}}^+ = -0.8 \, \text{V}$ and $V_{\text{bias}}^- = -0.3 \, \text{V}$ in Fig. 9, all of the attractors on the Poincaré sections asymptotically converge to closed curves. This confirms that the observed complex attractors in the range of $-0.8 \, \text{V} < V_{\text{bias}}^+ < -0.3 \, \text{V}$ in Fig. 9 are quasi-periodic attractors. Moreover, we conclude that when $V_{\text{bias}}^+ > -0.2 \, \text{V}$, the observed complex attractors are chaotic because a folding-and-stretching mechanism is observed. Therefore, the bifurcation phenomenon in Fig. 9 is the chaos via torus breakdown [9]. The border between chaos and torus exists around $V_{\text{bias}}^+ = -0.3 \, \text{V}$.

Furthermore, we consider the case where the duty ratio of the periodic external signal $T_{\text{bias}}^+/T$ is varied. Figure 10 shows the bifurcation diagram in which $T_{\text{bias}}^+/T$ is swept as
the bifurcation parameter. The horizontal axis of the figure is the duty ratio of the external signal in percent, and the vertical axis is $v$ of the attractors on the Poincaré sections. Here, $V_{bias}^+ = -0.11 \text{ V}$, $V_{bias}^- = -0.72 \text{ V}$, and $T = 100 \text{ ns}$.

To understand the properties of the complex attractors in Fig. 10, we examine the attractors on the Poincaré sections at various phases of the external signal. As a result, we observed the folding-and-stretching mechanism; thus, the complex orbits in Fig. 10 are chaotic.

4. Conclusions

We have proposed a forced chaos generator with a CMOS variable active inductor circuit. Chaos, a quasi-periodic attractor, and a periodic attractor were observed in SPICE simulations. Poincaré section analyses were used to confirm the chaotic dynamics of the proposed circuit. Moreover, we presented the bifurcation phenomena when the amplitude and period of the external signal were swept as bifurcation parameters. As a future task, we plan to develop the proposed circuit for practical IC implementation.

This research was supported in part by JSPS through its Funding Program for World-Leading Innovative R&D on Science and Technology (FIRST Program). This work was also supported by Kakenhi (20300085), and VDEC, the University of Tokyo in collaboration with Cadence Design Systems, Inc.

References


Chaos in Chua’s Oscillator with Chua’s Diode and Memristor

Tuyen Tran Xuan, Nguyen Tien Dzung and Thang Manh Hoang

Faculty of Electronics and Telecommunications
Hanoi University of Science and Technology
1 Dai Co Viet, Hanoi, Vietnam
Email: thang@ieee.org

Abstract—This paper presents a chaotic oscillator obtained by modifying the canonical Chua’s oscillator. Chua’s diode and memristor are combined into a circuit and investigated. Simulation result shows a strange chaotic attractor confirmed by positive Lyapunov exponents.

1. Introduction

A two-terminal circuit element called – the memristor was first postulated by Leon O. Chua in September 1971 [1]. It is known as the forth basic circuit element after resistor (R), capacitor (C), and inductor (L). In April 2008, Stanley Williams and researchers in HP Information and Quantum Systems Laboratory announced the fabrication of a nano scale memristor [2]. From this milestone discovery, memristor has received sharply increasing attention in both research and industry. So far, many potential applications of memristor have been proposed, as in artificial biological systems, non-volatile RAM (NVRAM), application specific integrated circuits (ASICs) and field programmable gate arrays (FPGAs). For integrated circuit technology, a significant reduction in area with an unprecedented memory capacity and device density of memristors enables the maintaining of Moore’s law. Many researchers around the world have been focusing on memristor applications in various areas of circuit design, alternative materials, spintronic memristors and memristor modeling.

With the nonlinear characteristic, memristor exhibits rich behaviors in dynamical system, especially in chaotic circuits. In this paper, we study the phenomena when adding memristor into the canonical Chua’s oscillator. Simulation results and Lyapunov exponents calculation demonstrate that the modified Chua’s circuit can generate chaos attractor.

1.1 Monotone-increasing piecewise-linear memristor

The memristor shown in Figure 1 is characterized by a non-linear constitutive relation between the voltage \( v \) and current \( i \) across the element as

\[
v = M(q)i, \quad \text{or} \quad i = W(\phi)v,
\]

(1)

Where \( q \), \( \phi \), \( M(q) \) and \( W(\phi) \) are the charge, flux, memristance and memductance of the memristor, respectively. Two function \( M(q) \) and \( W(\phi) \) are defined below:

\[
M(q) = \frac{dq}{dq}, \quad \text{and} \quad W(\phi) = \frac{d\phi}{d\phi},
\]

(2)

The charge-controlled memristor [3] has the "monotone-increasing" and "piecewise-linear" nonlinearity shown in Figure 2, with the relation between charge and flux demonstrated by the function \( q(\phi) \).

\[
q(\phi) = b\phi + 0.5(a - b)(|\phi + 1| - |\phi - 1|),
\]

(4)

1.2. Canonical Chua’s oscillator

The canonical Chua’s oscillator depicted in Figure 3 consists of an inductor \( L \), two capacitors \( C1, C2 \), a Chua’s diode and a negative conductance \( -G \). The function \( F(v) \) defined below represent the \( i-v \) characteristic of the Chua’s diode shown in Figure 4.

\[
i = F(v) = G_dv^2 + 0.5(G_d - G_b)(v + B_p)|v - B_p|, \quad (5)
\]
2. Fourth-order chaotic oscillator

We now study the modified canonical Chua's oscillator shown in Figure 5.

![Figure 5. The modified canonical Chua's oscillator with memristor and Chua's diode.](image)

The Kirchhoff equations of this circuit are here below:

\[
\begin{align*}
    i_1 &= i_3 - i, \\
    v_3 &= v_2 - v_1, \\
    i_2 &= -i_3 + i_4,
\end{align*}
\]

(6)

Since

\[
\begin{align*}
    i_1 &= C_1 \frac{dv_1}{dt}, \\
    i &= W(\varphi)v, \\
    v_3 &= L \frac{di_3}{dt}, \\
    i_4 &= -F(v_4), \\
    v_4 &= v_2, \\
    v &= v_1.
\end{align*}
\]

We have

\[
\begin{align*}
    C_1 \frac{dv_1}{dt} &= i_3 - W(\varphi)v_1, \\
    \frac{di_3}{dt} &= v_2 - v_1, \\
    C_2 \frac{dv_2}{dt} &= -i_3 - F(v_2), \\
    \frac{d\varphi}{dt} &= v_1,
\end{align*}
\]

(7)

Let \( x = v_1, \ y = i_3, \ z = v_2, \ w = \varphi, \ \alpha = 1/C_1, \ \beta = 1/C_2, \ \gamma = 1/L. \)

Above equations become

\[
\begin{align*}
    \frac{dx}{dt} &= \alpha[y - W(w)x], \\
    \frac{dy}{dt} &= \gamma[z - x], \\
    \frac{dz}{dt} &= -\beta[y + F(z)], \\
    \frac{dw}{dt} &= x,
\end{align*}
\]

(8)

where

\[
\begin{align*}
    q(w) &= bw + 0.5(a - b)(|w + 1| - |w - 1|), \\
    W(w) &= \frac{dw}{d\varphi} = \begin{cases} 
        a, & w < 1, \\
        b, & w > 1,
    \end{cases}
\end{align*}
\]

(9)

and

\[
F(z) = G_0z + 0.5(G_a - G_b)(|z + B_p| - |z - B_p|)
\]

(10)

Simulation result for Eq.(8) with the parameter set as listed in Table I is depicted in Figure 6, 7 and 8. Figure 6 illustrates the result of Lyapunov exponents using Lyapunov Exponents Tools [5] (LET). It is clear that two of four Lyapunov exponents are positive at around \( \lambda_1 = 5.2 \times 10^{-3} \) and \( \lambda_2 = 2.2 \times 10^{-3} \); therefore, the system exhibits chaotic behavior. Figure 7 and 8 depict the strange attractors of the system.

**TABLE I. CIRCUIT PARAMETER SET FOR A CHAOTIC ATTRACTOR**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>a</th>
<th>b</th>
<th>( G_a )</th>
<th>( G_b )</th>
<th>( B_p )</th>
<th>( \alpha )</th>
<th>( \beta )</th>
<th>( \gamma )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>0.2</td>
<td>9</td>
<td>-0.6</td>
<td>-0.4</td>
<td>0.5</td>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

3. Conclusion

There are some memristor-based chaotic circuits proposed in recent papers, these circuits obtained by replacing the Chua’s diode in Chua’s circuit by a memristor. This paper considers the case when the modified canonical Chua’s oscillator contains both Chua’s diode and memristor. With the results obtained, we conclude that this circuit can entrance the chaotic circuit family, extending the knowledge of memristor behaviors and chaotic circuits.
(a). Lyapunov Exponents of the system

(b). Two positive Lyapunov Exponents zoomed from (a)

Figure 6. Lyapunov exponents calculation

(a). 3D chaotic attractor, $\phi$ vs. $i_3$ vs. $v_1$

(b). 3D chaotic attractor, $v_2$ vs. $i_3$ vs. $v_1$

Figure 7. Trajectories in state space
Figure 8. Chaotic attractors

Acknowledgments

This work has been supported by Ministry of Science and Technology of Vietnam with the project number DTDL2009G/44. The authors would like to thank Vietnam’s National Foundation for Science and Technology Development (NAFOSTED).

References

Chaos Synchronization in Coupled Delayed Two-stage Colpitts Circuits for UWB Communications

Hien Quang Vu, Nguyen Tien Dzung, Vu Van Yem and Thang Manh Hoang

Faculty of Electronics and Telecommunications, Hanoi University of Science and Technology, 1 Dai Co Viet, Hanoi, VIETNAM
Email: thang@ieee.org

Abstract—The paper presents chaos synchronization in coupled delayed two-stage Colpitts circuits for the first time. Numerical simulation demonstrates and verifies the effectiveness of the proposed model. Furthermore, the spectrum analysis shows the potential utilization for UWB communications.

1. Introduction

The Colpitts oscillator has been attractive for many years due to the application in telecommunications [1,2] as a source of oscillation. Chaotic Colpitts oscillators have received increasing attention because they can generate chaotic signal at the frequency range of kilohertz [3,4], UHF [5,6], VHF [5], and very recently of UWB communications with the range from 3.1GHz – 10.6GHz [7]. However, Colpitts circuit is not easy to generating chaotic signals at the frequency range of 3.1GHz to 10.6 Ghz. With inclusion of time delay, the system is much easier in generation of chaotic behavior in compare with conventional ones [8]. Moreover, due to the complexity in dynamics, delay systems have a potential application in secure communications.

So far, there are several types of synchronization in coupled systems. Synchronization represents correlations between the master and slave systems i.e. schemes of complete [9,10], generalized [11], anticipating [10,12], lag [10,13], and phase synchronizations [13,14]. In oscillation circuits, synchronization is to adjust frequencies of weakly interacting periodic oscillators at the side of slave [2,3].

In this paper, for the first time we investigate a delay two-stage Colpitts oscillator by means of incorporating delay to conventional one. Also, synchronization in coupled delay two-stage Colpitts circuits is studied and hardware design for the system is presented. Simulation result shows the evidence of chaotic behavior and synchronization in frequency range of UWB (3.1GHz – 10.6GHz) communications.

2. Circuit Design

Fig. 1 depicts the conventional two-stage Colpitts oscillator [6]. The circuit includes one conductor L, one resistor R, three capacitors C1, C2, C3 and two transistors, Q1, Q2. In comparison with the classical single-transistor Colpitts oscillator [15,16], the two-stage modification includes an extra transistor Q3, and a capacitor C3 to obtain oscillation at higher fundamental frequency, up to $f = 0.3f_{cut-off}$.

The fundamental frequency $f$ of a two-stage Colpitts oscillator can be estimated as

$$f = \frac{1}{2\pi} \sqrt{\frac{C_1 C_2 + C_1 C_3 + C_2 C_3 - \frac{R^2}{L^2}}{L C_1 C_2}}$$

The circuit schematic for a delay two-stage system is as shown in Fig. 2. It consists of three bipolar junction transistors (BFG 425W, Phillip Co. Ltd.), cut-off frequency of 25GHz which is biased in the active region by means of Vee, Re and Vcc. The Q1 and Q2 compose for the intrinsic two-stage Colpitts oscillator while Q3 is an emitter follower inserted to buffer the influence of the measuring devices. Other components require an inductor L with series resistance R, and three capacitors C1, C2 and C3. The bias emitter current I0 can be manually changed by varying the voltage source Vee. A delay line consists of 20 pairs of LCs, and the time delay can be estimated by $\tau_d = n\sqrt{LC}$ seconds. With the presence of delay-line, chaotic behavior is more easily generated. The delayed signal is feedbacked to the collector of Q1.
In Fig. 3, Master and Slave are coupled each other through the resistor Re for impedance matching. Master’s and Slave’s circuits are identical as depicted in Fig. 2. The output signal from Master is fed to the emitter of Q2 at the Slave side. Full synchronization is achieved with \( x_m(t) \approx x_s(t) \).

The block diagram for synchronization using two Colpitts oscillators is as in Fig. 3. One Colpitts oscillator plays a role of Master and the other does as Slave. We will denote subscript “m” and “s” for the Master and Slave, respectively.

3. Synchronization model

As depicted in Fig. 2, the time delay signal VC1 is feedbacked to the collector of Q1, so state equations for Master can be expressed as

\[
\begin{align*}
C_{L,m} \frac{dV_{C1,m}}{dt} &= I_{L,m} - I_{RQ1,m}(r,V_{C2,m},V_{C3,m}) + \frac{1}{R_{d,m}} V_{C1,m,r} \\
L_{m} \frac{dI_{L,m}}{dt} &= V_{C,m} - V_{C1,m} - V_{C2,m} - V_{C3,m} - R I_{L,m} \\
C_{3,m} \frac{dV_{C3,m}}{dt} &= I_{L,m} - I_{RQ2,m}(r,V_{C2,m}) \\
C_{2,m} \frac{dV_{C2,m}}{dt} &= I_{L,m} - I_{Q,m}.
\end{align*}
\]

It is clear to observe from Fig. 3 that the signal from Master is connected to the emitter of Q2 of Slave’s circuit, so we can describe state equations of Slave as

\[
\begin{align*}
C_{L,s} \frac{dV_{C1,s}}{dt} &= I_{L,s} - I_{RQ1,s}(r,V_{C2,s},V_{C3,s}) + \frac{1}{R_{d,s}} V_{C1,s} \vphantom{m} \\
L_{s} \frac{dI_{L,s}}{dt} &= V_{C,s} - V_{C1,s} - V_{C2,s} - V_{C3,s} - R I_{L,s} \\
C_{3,s} \frac{dV_{C3,s}}{dt} &= I_{L,s} - I_{RQ2,s}(r,V_{C2,s}) \\
C_{2,s} \frac{dV_{C2,s}}{dt} &= I_{L,s} - I_{Q,s} + \frac{1}{R_{C}} (V_{C2,m} - V_{C2,s}).
\end{align*}
\]

Assumed that the forward current gain of transistors (denoted as \( \alpha \)) is equal to 1 as given in Eqs. (2) and (3). In the forward active mode, the intrinsic resistance (denoted as \( r \)) is the differential one of the base-emitter junction and the break-point voltage (denoted as UT) is around 0.7 voltage. Above expressions can be converted to dimensionless by using following assumptions

\[
\begin{align*}
x &= \frac{V_{C1}}{U_T}, y = \frac{\rho_{L1}}{U_T}, z = \frac{V_{C2}}{U_T}, v = \frac{V_{C3}}{U_T} \\
t = \frac{x}{\sqrt{L_C}}, \tau = \frac{\tau_d}{\sqrt{L_C}}, \tau_d = \frac{n I_d E_d}{R_d}.
\end{align*}
\]

By applying the method of piece-wise linear approximation to the current voltage characteristics of the base-emitter junctions, we have

\[
F_1(a,z,v) = \begin{cases} 1 - a(z + v), & a(z + v) < 1 \\ 0, & a(z + v) \geq 1 \end{cases}
\]

\[
F_2(a,z) = \begin{cases} 1 - az, & az < 1 \\ 0, & az \geq 1 \end{cases}
\]
From Eqs. (4), (5) and (6), Eq. (2) for Master can be represented in the form of dimensionless as

\[
\begin{align*}
\frac{dx_m}{dt} &= y_m - F_{1,m}(a,z,v) + dx_m, \\
\frac{dy_m}{dt} &= -x_m - z_m - v_m - by_m, \\
\epsilon_{3,m} \frac{dv_m}{dt} &= y_m - F_{2,m}(a,z), \\
\epsilon_{2,m} \frac{dz_m}{dt} &= y_m - 1,
\end{align*}
\]

(7)

and Slave from Eqn. (3) becomes

\[
\begin{align*}
\frac{dx_s}{dt} &= y_s - F_{1,s}(a,z,v) + dx_s, \\
\frac{dy_s}{dt} &= -x_s - z_s - v_s - by_s, \\
\epsilon_{3,s} \frac{dv_s}{dt} &= y_s - F_{2,s}(a,z), \\
\epsilon_{2,s} \frac{dz_s}{dt} &= y_s - 1 + c(z_m - z_s),
\end{align*}
\]

(8)

4. Simulation result

The numerical simulation is carried out for Eqs. (7) and (8) with \(a = 20, b = 0.8, c = 0.02, d = 0.02, \epsilon_{2,3} = 1\). The result for \(y\) versus \(x+v+z\) shown in Fig. 4 presents the strange attractor or chaotic behavior is exhibited. Simulation result for the circuit in Figs. 2 and 3 using Advanced Design System 2008 Update 2 (ADS) is carried out with the value of parameters chosen as \(R_0 = 100\Omega, R_1 = 510\Omega, R_2 = 3k\Omega, R_3 = 5.1k\Omega, R_4 = 3k\Omega, R_5 = 200\Omega, R_6 = 1.5k\Omega, C_0 = 47nF, C_1 = 1pF, C_2 = 270pF\), (other parameters of the tank elements, namely \(R, L, \) and \(C_1\) depend on the chosen fundamental frequency \(f\)). In order to achieve the frequency band of UWB communication (3.1 GHz - 10.6 GHz), the microwave transistor BFG425W (Phillip Co. Ltd) is chosen. The value for the delay line consisting of 20 pairs of LC circuit is
adopted as $L = 1\text{nH}$, $C = 1\text{pF}$, and $R_d = 50\Omega$. The specific value of the supply voltages $V_{cc}$ and $V_{ee}$ is adjusted to achieve the desired chaotic performance of the oscillator. Based on the previous result of numerical integration for the dimensionless system as given above, we choose $R=10\Omega$, $L=1\text{nH}$, $C_1 = C_2 = C_3 = 5\text{pF}$, $V_{cc} = 12\text{V}$, $V_{ee} = 30\text{V}$, and $R_c=50\Omega$. Hence, the fundamental frequency is $f \approx 3.1\text{GHz}$.

It is observed from Figs. 5 (a) and (b) that random-like, non-periodic signals generated by Master and Slave. This is chaotic signal [17]. The power spectrum in Fig. 5 (c) and (d) is obtained by analyzing the signal $V_m$ and $V_s$. In the range of the frequency beyond $f \approx 3.1\text{GHz}$, the power average is less than -41 dB. In summary, the synchronization in time delay Colpitts circuits can be utilized in UWB communications.

## 5. Conclusion

In this paper, we have demonstrated the chaos synchronization in the coupled time delay two-stage Colpitts circuits. The analysis on power spectrum has shown that time delay Colpitts circuit generates chaotic signals in the frequency range from 3.1 Ghz to 10.6 Ghz and the power is less than -41 dB. In summary, the synchronization in time delay Colpitts circuits can be utilized in UWB communications.

### Acknowledgments

This work has been supported by Ministry of Science and Technology of Vietnam with the project number DTDL2009G/44 and by Vietnam’s National Foundation for Science and Technology Development (NAFOSTED).

### References


An Energy Based Investigation of Rössler Type Chaos on Chua’s Circuit

Mustafa Kösem† and N. Serap Şengör†

†Electronics and Communication Engineering Department, Faculty of Electrical and Electronic Engineering, Istanbul Technical University, Maslak, TR-34469 Istanbul, Turkey
Email: kosem@itu.edu.tr, sengorn@itu.edu.tr

Abstract—In order to deal with nonlinear systems, energy functions have been considered in many applications. In this paper, the aim is to set an approach based on energy functions to understand the mechanism behind the chaotic behavior of Chua’s Circuit. In order to explain the effect of nonlinear resistor, different nonlinearities are considered and the simulation results based on energy functions are given.

1. Introduction

Euler-Lagrangian and Hamiltonian formalisms of circuits, especially nonlinear circuits, have been subject of interest for decades [1–3]. One reason for this interest is their versatility in explaining the behavior of mechanical systems and providing a sound approach to the stability analysis due to the consideration of energy function along with differential equations defining the dynamics of the system. As no general method can be developed for the analysis of nonlinear circuits like the well defined, easily applicable methods of linear circuit analysis, energy based approaches would give insight to understanding the behavior of nonlinear circuits.

The energy storage elements, namely capacitors and inductors characterize the dynamics of an electric circuit. So understanding energy exchange between these energy storage elements and other elements in an electric circuit would be functional in explaining the complex behavior of nonlinear circuits. This is the main motivation of energy-based techniques in nonlinear circuit theory [1]. Even though Lagrangian and Hamiltonian formulations are being used effectively in classical mechanics, this is not so easy in electric circuits as in circuit theory it is not clear how to define potential energy and kinetic energy. There are some studies in the literature to overcome this difficulty [2, 3].

Also energy point of view is effective in studying the control [4] and synchronization [5, 6] of the chaotic systems. In this work we will use energy function to investigate chaos producing mechanism in Chua’s circuit. Chua’s circuit is preferred for its simplicity and rich dynamic behavior and these attributes of the circuit made it very popular in chaos theory [7, 8].

Our approach will be to follow the exchange in the Hamiltonian of the nonlinear circuit along the trajectories and investigate energy exchange in the Chua’s circuit. Nonlinear resistance $N_R$, is the only nonlinear element in Chua’s circuit. If we replace it with a passive resistance there is no interesting dynamic in the circuit as the solutions decay in time and approaches zero [9]. To give an idea of how to use Hamiltonian function, we will take this case as our starting point and will examine the effect of the nonlinear resistor on the energy exchange. In this work we will examine the Rössler type chaos of Chua circuit from energy point of view.

This paper contains the following sections: In the following section Hamiltonian equations and Hamiltonian of the Chua’s Circuit is derived. In Section III, the equations related to energy exchange of the circuit are given. Simulation results and related discussions for different Chua diodes are presented in Section IV. Finally, the conclusions are given in Section V.

2. Hamiltonian Equations of Chua’s Circuit

The equations related to Chua’s circuit, shown in Fig.1 are commonly given with the following state equations:

$$\begin{align*}
C_1 \dot{v}_1 &= G(v_2 - v_1) - g(v_1) \\
C_2 \dot{v}_2 &= i_3 - G(v_2 - v_1) \\
L \dot{i}_3 &= -v_2
\end{align*}$$

where $i_R = g(v_R) = G_b v_R + \frac{1}{2} (G_a - G_b) [v_R + E - |v_R - E|]$ (2) is the node equation of the nonlinear resistor $N_R$.

Figure 1: Chua’s Circuit.

To derive Lagrangian of the circuit the method proposed by Chua and McPhearson [2] has been used. Following this method generalized coordinates $(\phi)$ and generalized velocities $(\dot{\phi})$ are defined as follows:

$$\begin{align*}
\phi &= [\phi_1 \ \phi_2]^T \quad (3a) \\
\dot{\phi} &= [v_1 \ v_2]^T \quad (3b)
\end{align*}$$

Euler-Lagrangian and Hamiltonian formalisms are being used effectively in classical mechanics, this is not so easy in electric circuits as in circuit theory it is not clear how to define potential energy and kinetic energy. There are some studies in the literature to overcome this difficulty [2, 3].

Also energy point of view is effective in studying the control [4] and synchronization [5, 6] of the chaotic systems. In this work we will use energy function to investigate chaos producing mechanism in Chua’s circuit. Chua’s circuit is preferred for its simplicity and rich dynamic behavior and these attributes of the circuit made it very popular in chaos theory [7, 8].

Our approach will be to follow the exchange in the Hamiltonian of the nonlinear circuit along the trajectories and investigate energy exchange in the Chua’s circuit.
where $\phi_1$ and $\phi_2$ are fluxes of the capacitors. We will omit details in deriving Lagrangian and Euler-Lagrange equations for simplicity. As a result Lagrangian becomes,

$$\mathcal{L} = \frac{1}{2} C_1 \dot{v}_1^2 + \frac{1}{2} C_2 \dot{v}_2^2 - \frac{1}{2L} [\phi_2 - \phi_2(0) + \phi_3(0)]^2$$  \hspace{1cm} (4)

where

$$\phi_3(0) = L_i(0).$$  \hspace{1cm} (5)

One could derive Hamiltonian by using below equations:

$$\mathcal{H} = y^T \dot{x} - \mathcal{L}(x, \dot{x})$$  \hspace{1cm} (6)

$$y \equiv \frac{\partial \mathcal{L}}{\partial \dot{x}}$$  \hspace{1cm} (7)

Here $y$ denotes the generalized moments as given in [10].

Using equations (3b), (4), (6) and (7) $\mathcal{H}$ is derived as follows:

$$\mathcal{H} = \frac{1}{2} C_1 \dot{v}_1^2 + \frac{1}{2} C_2 \dot{v}_2^2 + \frac{1}{2L} [\phi_2 - \phi_2(0) + \phi_3(0)]^2.$$  \hspace{1cm} (8)

This is Hamiltonian of the Chua’s circuit. In order to see that it is equal to sum of energies of the dynamic elements, it could be rewritten by using $L-C$ loop equation in Chua’s circuit (Fig.1). With this manipulation we could get below equation which is in conventional form:

$$\mathcal{H} = \frac{1}{2} C_1 \dot{v}_1^2 + \frac{1}{2} C_2 \dot{v}_2^2 + \frac{1}{2L} Li^2.$$  \hspace{1cm} (9)

From this point to the end of the paper we will work with dimensionless Chua equations for simplicity. Dimensionless form of the Chua’s equations are given as follows:

$$dx/d\tau = \alpha(y - x - f(x))$$

$$dy/d\tau = x - y + z$$

$$dz/d\tau = -\beta y$$  \hspace{1cm} (10)

where

$$f(x) = bx + \frac{1}{2}(a-b)[|x+1| - |x-1|]$$  \hspace{1cm} (11)

is a piecewise linear function with 3-segments and is associated with Chua’s diode. In order to obtain these equations, rescaling relations given in [7] are used. Actually using quantities $m_0$ and $m_1$ instead of $a$ and $b$ with the relations $m_0 = 1 + a$ and $m_1 = 1 + b$ gives slightly simpler equations [8]. However for our purpose using $a$ and $b$ is beneficial.

Hamiltonian can be rewritten as

$$\mathcal{H} = \frac{1}{2} \left[ \frac{1}{\alpha} x^2 + y^2 + \frac{1}{\beta} z^2 \right]$$  \hspace{1cm} (12)

with a scale difference from (9) as given in [5].

3. Energy Surfaces for Double Scroll

Fig.2 gives an idea about how Hamiltonian is changing in the phase space. In the shaded regions where Hamiltonian is increasing; Chua’s diode is providing energy to the circuit.

To determine the separating planes in Fig.2, one need to solve the equation $d\mathcal{H}/d\tau = 0$. By taking derivative of (12) and then substituting (10) into the the result yields the following equation

$$\frac{d\mathcal{H}}{d\tau} = -(x - y)^2 - xf(x).$$  \hspace{1cm} (13)

where the term $f(x)$ in (13) has been given in (11).

In order to investigate the relation between the Chua’s diode’s characteristic and the energy exchange between the elements in the circuit, the characteristic of the diode will be considered further and each piecewise linear part will be considered step by step. In the upper side of Fig.3, (a) and (b), these cases are given. The equations for (a) and (b) are as follows, respectively:

$$f(x) = \begin{cases} ax, & \text{for } x \geq -1 \\ bx - a + b, & \text{for } x < -1 \end{cases}$$  \hspace{1cm} (14a)

$$f(x) = \begin{cases} ax, & \text{for } x \geq -1 \\ bx - a + b, & \text{for } x < -1 \end{cases}$$  \hspace{1cm} (14b)

By using (13) and (14), one could derive the solution of the equation $d\mathcal{H}/d\tau = 0$. Since (14b) covers (14a) as special case we will give the result only for the case, (14b).

$$y = \begin{cases} x \left(1 \pm \sqrt{-a}\right), & \text{for } x \geq -1 \\ x \pm \sqrt{(a-b)x - bx^2}, & \text{for } x < -1 \end{cases}$$  \hspace{1cm} (15)

These equations are separatrices for the regions where Hamiltonian is increasing. They are given in the lower side of Fig.3. In the next section each case will be investigated in detail.

4. The Effect of Nonlinear Resistor on Energy Function

Since (13) and as a result of this, (15) are independent of $z$, $x - y$ plane of generalized velocities is sufficient to follow how Hamiltonian is changing. We will consider those separatix planes and their effect on energy exchange.
4.1. $N_R$ as Active Resistance

When the characteristic of $N_R$ given in Fig.3(a) is used in Chua’s circuit the only dynamic behavior that could be observed is equilibrium point. Trajectory goes to either zero or infinity. These cases are shown in Fig.4. The value of bifurcation parameter $\alpha$ is equal to 9 and initial conditions are as $(x_0, y_0, z_0) = (0, 0.2, 0)$.

From this point to the end of the paper Chua’s circuit will be simulated under below conditions for different parameter sets:

- Temporary dynamics will be excluded in the figures.
- Value of the system parameter $\alpha$ is fixed and it is 9.
- The values for the bifurcation parameters $a$ and $b$ will be chosen such that the system has one real and a pair of complex conjugate roots for both parameters.

- Initial points will be chosen such that solutions won’t be affected by unstable eigenvector. This dynamic behavior can be followed from Fig.4(c) and (d).

4.2. $N_R$ as 2-segment Active Resistance

In the following different parameter values will be considered.

4.2.1. $a=-13/70$, $b=-5/7$

In Fig.5 the bifurcation dynamic of the circuit is given. For small values of $\beta$, system always gains energy, every trajectory diverges (see Fig.5(a)). While $\beta$ increasing an unstable limit cycle occurs. System loses energy in inner region and gains energy in outer region. Energy exchange of the system is given for two different values of $\beta$ in Fig.5 (b) and (c). Further, increasing the value of $\beta$ causes limit cycle to extinct. In this case system always loses energy, every trajectory converges to an equilibrium point (see Fig.5(d)).

4.2.2. $a=-5/7$, $b=-13/70$

In the previous case, we have observed that when inner region loses energy and outer region gains energy an unstable limit case occurs. Using this result, first insight may be swapping the values of the system parameters would be enough to create a stable limit cycle. Interestingly this is not enough. In Fig.6(a-c) the bifurcation dynamic of the circuit is given. To understand why system couldn’t produce stable limit cycle, it is needed to examine energy exchange. In Fig.6(d) energy exchange of the system is plotted, each time the trajectory crosses the line $y = x \{1 \pm \sqrt{a}\}$. When system gains energy red line is used and when system loses energy blue line is used.

4.2.3. $a=-3/7$, $b=-13/70$

In Fig.7 the bifurcation dynamic of the circuit is given. As a comparison with previous case for $\beta = 21$ system gains less energy than it loses in the outer region and a stable limit cycle formed.
4.2.4. \( a = -13/70, b = -8/7 \)

In Fig. 8 the bifurcation dynamic of the circuit is given. Starting with large \( \beta \) values, the system energy is always decreasing and as \( \beta \) values are decreased, three regions occur, where in the in-between region the energy increases. This case \( \beta = 19 \) corresponds to occurrence of stable limit cycle. As \( \beta \) is decreased more, interference occurs between regions where energy is decreasing and increasing in the outer regions. This phenomena keeps on as \( \beta \) is decreased more, interference occurs more; first period doubling then, with period doubling cascade or symmetry breaking chaotic behavior raises.

In the inner region, the energy could be either increasing or decreasing for different bifurcation parameter values, but this does not have an effect on dynamical behavior. If in the inner region energy is increasing while chaotic behavior occurs this chaotic behavior is robust.

5. Conclusion

In this paper, the period doubling mechanism in Chua’s circuit is investigated utilizing energy relations for Rössler type Chaos. Stepwise, first Hamiltonian of the circuit is obtained and then with different Chua diodes, the relation between the energy function and the dynamic behavior is observed. The expectation is to set a theoretical method to investigate chaos based on energy function.

References


Global Asymptotic Stability Analysis of Nonlinear Circuits for Solving the Maximum Flow Problem

Norikazu Takahashi

Department of Informatics, Kyushu University
744 Motooka, Nishi-ku, Fukuoka 819-0395 Japan
Email: norikazu@inf.kyushu-u.ac.jp

Abstract—Global asymptotic stability of a nonlinear circuit for solving the maximum flow problem, which was first proposed by Sato et al., is studied in this paper. The circuit consists of two independent DC voltage sources, capacitors and nonlinear resistors. It is proved rigorously that the circuit has a unique equilibrium point which is globally asymptotically stable. From the viewpoint of dynamical systems, the circuit is a cooperative system, and thus some fundamental results concerning the convergence property of cooperative systems play important roles.

1. Introduction

From the mid-eighties, there have been many attempts to solve optimization problems by using nonlinear circuits [1, 2, 3, 4]. One of the most important examples is the Hopfield neural network [1]. This continuous-time recurrent neural network model, which can be implemented by a nonlinear circuit, is a powerful tool for finding an approximate solution of the traveling salesman problem. Also, some authors proposed to use SPICE, the most widely used circuit simulator, for solving constrained optimization problems [3, 4].

Recently, Sato et al. [5] proposed a nonlinear circuit1 for solving the maximum flow problem [6, 7, 8]. The circuit consists of two independent DC voltage sources, capacitors and nonlinear resistors. They performed a number of computer simulations and observed that the circuit always converges to an equilibrium point that corresponds to the maximum flow. However, the convergence property of the circuit has not been completely understood so far.

In this paper, it is proved under certain mild conditions that the nonlinear circuit proposed by Sato et al. [5] has a unique equilibrium point which is globally asymptotically stable. From the viewpoint of dynamical systems, the circuit belongs to an important class of dynamical systems called the cooperative system [9, 10]. Thus the proof given in this paper is based on a fundamental result [11] concerning the global stability of cooperative systems. First, the boundedness of state trajectories is proved. Second, it is proved that every equilibrium point is locally asymptotically stable. Third, the uniqueness of the equilibrium point is proved by making use of the Brouwer degree [12, 13]. Finally, it is proved that the unique equilibrium point is globally asymptotically stable.

2. Maximum Flow Problem

First of all, we briefly review the maximum flow problem. Let \( G = (V,E) \) be a simple directed graph where \( V = \{v_0,v_1,\ldots,v_n\} \) is the set of vertices and \( E = \{e_1,e_2,\ldots,e_m\} \) is the set of edges. An edge \( e_k \in E \) directed from \( v_i \in V \) to \( v_j \in V \) is denoted by \( e_k = (v_i,v_j) \). The set \( V \) contains two distinguished vertices: the source \( v_0 \) and the sink \( v_{n+1} \). The source \( v_0 \) is the vertex such that \( E \) contains no edge directed to \( v_0 \). On the other hand, the sink \( v_{n+1} \) is the vertex such that \( E \) contains no edge directed from \( v_{n+1} \).

Throughout this paper, we assume:

Assumption 1 For each vertex \( v_i \in V \setminus \{v_0,v_{n+1}\} \), there is at least one directed path from \( v_0 \) to \( v_i \) and there is at least one directed path from \( v_i \) to \( v_{n+1} \).

Let \( c : E \rightarrow \mathbb{R}_+ \) be a capacity function where \( \mathbb{R}_+ \) is the set of positive numbers. The capacity of an edge \( (v_i,v_j) \) is denoted by \( c(v_i,v_j) \). A flow on the graph \( G \) is a function \( f : E \rightarrow \mathbb{R} \) satisfying the following conditions:

\[
0 \leq f(v_i,v_j) \leq c(v_i,v_j), \quad \forall (v_i,v_j) \in E
\]

\[
\sum_{j : (v_i,v_j) \in E} f(v_i,v_j) = \sum_{j : (v_j,v_i) \in E} f(v_j,v_i), \quad i = 1,2,\ldots,n
\]

The maximum flow problem is to find a flow \( f \) which maximizes

\[
|f| = \sum_{j : (v_0,v_j) \in E} f(v_0,v_j)
\]

An example of simple directed graphs is shown in Fig. 1 where \( v_0 \) and \( v_s \) are the source and the sink, respectively, and numbers beside edges represent the capacity.


Sato et al. [5] proposed a nonlinear circuit for solving the maximum flow problem. Their circuit consists of two independent DC voltage sources, \( n \) capacitors and \( m \) nonlinear resistors (see Fig. 2). Nodes of the circuit correspond

---

1 In their paper [5], the circuit is called the maximum-flow neural network.
Let $x(t) = (x_1(t), x_2(t), \ldots, x_n(t))^T$ be a solution of the differential equation (1). Then

$$\lim_{t \to \infty} A_{ij} \sigma(x_i(t) - x_j(t)), \quad i, j = 0, 1, \ldots, n$$

correspond to an approximate solution of the maximum flow problem. Therefore, the circuit must converge to an equilibrium point for any initial condition in order to work properly as a maximum flow problem solver.

4. Global Asymptotic Stability Analysis

4.1. Boundedness of Solutions

Lemma 1 Let $\Omega = [L, U]^n \subset \mathbb{R}^n$. Every solution $x(t) = (x_1(t), x_2(t), \ldots, x_n(t))^T$ of the system (1) belongs to $\Omega$ for all $t \geq 0$ if $x(0) \in \Omega$.

Proof. Let $x$ be any point in $\partial \Omega$ where $\partial \Omega$ represents the boundary of $\Omega$. If $x_i = L$ then $\sigma(U - x_i), \sigma(x_j - x_i) (j = 1, 2, \ldots, n), -\sigma(x_i - x_j) (j = 1, 2, \ldots, n)$, and $-\sigma(x_i - L)$ are all nonnegative because $\sigma$ is a monotone increasing function. Hence $F_i(x)$ is nonnegative. Similarly we can show that $F_i(x)$ is nonpositive if $x_i = U$. Therefore, any solution $x(t)$ such that $x(0) \in \Omega$ cannot leave $\Omega$. In other words, $\Omega$ is a positively invariant set for the system (1). \hfill \square

4.2. Local Asymptotic Stability of an Equilibrium Point

Lemma 2 If $x^* = (x_1^*, x_2^*, \ldots, x_n^*)^T \in \Omega = [L, U]^n$ is an equilibrium point of the system (1) then $x^* \in \text{int} \Omega$ and is locally asymptotically stable.

Proof. Suppose that the system (1) has an equilibrium point $x^* \in \partial \Omega$. Then there exists at least one $i$ such that $x_i^* = L$, or $x_i^* = U$. In the case where $x_i^* = L$, the value of $x_j^*$ must be $L$ for all $j$ such that $(v_j, v_i) \in E$ because otherwise $F_i(x^*)$ is positive which contradicts the assumption that $x^*$ is an equilibrium point. For the same reason, the value of $x_i^*$ must be $L$ for all $k$ such that $(v_k, v_i) \in E$ and $(v_i, v_j) \in E$ for some $j$. By repeating this discussion, we reach the conclusion that there exists an integer $l$ such that $(v_0, v_l) \in E$ and the value of $x_l^*$ must be $L$. However, this implies that $F_l(x^*)$ is positive which leads to a contradiction. In the case where $x_i^* = U$, we can show in the same way as above that there exists an integer $l$ such that $(v_l, v_{i+1}) \in E$ and the
value of $x_i$ must be $U$, which leads to a contradiction. This completes the proof of the first statement.

For the second statement, let us consider the Jacobian matrix $J \in \mathbb{R}^{n \times n}$ of the vector field $F(x) = (F_1(x), F_2(x), \ldots, F_n(x))^T$ at an equilibrium point $x^*$. The $(i, j)$ element of $J$ is given by

$$J_{ij} = \begin{cases} 
-A_{0j} \sigma'(U - x_i^*) + \sum_{k=1}^{n} [A_{kj} \sigma'(x_j^* - x_k^*) + A_{ij} \sigma'(x_i^* - x_j^*)]/C_i, & i = j \\
A_{ij} \sigma'(x_j^* - x_i^*) + A_{ij} \sigma'(x_i^* - x_j^*)]/C_i, & i \neq j
\end{cases}$$

(2)

Note that the nonnegative constant $A_{ij}$ is positive if and only if $(v_i, v_j) \in E$ and that $\sigma$ is a monotone increasing function. From these facts and Assumption 1, we see that every diagonal element of $J$ is negative, every off-diagonal element of $J$ is nonnegative, and $J$ is irreducible (for the definition of the irreducible matrix, see [14] for example). Also, it is easily seen from (2) that $J$ satisfies

$$|J_{ij}| \geq \sum_{j=1,j \neq i}^{n} |J_{ij}|, \quad i = 1, 2, \ldots, n.$$  

In particular,

$$|J_{ik}| > \sum_{j=1,j \neq k}^{n} |J_{jk}|$$

holds for all $k$ such that $(v_i, v_j) \in E$ or $(v_i, v_{n+1}) \in E$. Hence $J$ is irreducibly diagonally dominant [14]. It is well known that if a square matrix is irreducibly diagonally dominant then it is nonsingular and if, in addition, its diagonal elements are negative then every eigenvalue has negative real part [14, Theorem 4.9]. Therefore, $J$ is nonsingular and every eigenvalue has negative real part, which means that the equilibrium point $x^*$ is locally asymptotically stable. □

4.3. Uniqueness of Equilibrium Point

Lemma 3 The system (1) has a unique equilibrium point in $\Omega = [L, U]^n$.

Proof. We first consider the system of linear differential equations:

$$\frac{dx_i}{dt} = \frac{1}{C_i} \left[ A_{0j} (U - x_j) + \sum_{j=1}^{n} [A_{ij} (x_j - x_i) - A_{ij} (x_i - x_j)] \right] \pm G_i(x),$$

$$i = 1, 2, \ldots, n. \quad (3)$$

Let $C = \text{diag}(C_1, C_2, \ldots, C_n) \in \mathbb{R}^{n \times n}$. Let a constant matrix $K = (K_{ij}) \in \mathbb{R}^{n \times n}$ and a constant vector $b = (b_1, b_2, \ldots, b_n)^T \in \mathbb{R}^n$ be defined as

$$K_{ij} = \begin{cases} 
-A_{0j} - \sum_{k=1}^{n} (A_{kj} + A_{ik}) - A_{ij}, & \text{if } i = j \\
A_{ij} + A_{ji}, & \text{if } i \neq j
\end{cases}$$

and

$$b_i = A_{0i} U + A_{ij} L.$$ 

Then (3) can be rewritten in a matrix form as follows:

$$\frac{dx}{dt} = C^{-1} (K x + b) \pm G(x). \quad (4)$$

By applying the same argument as in the proof of the second part of Lemma 2 to $C^{-1} K$, we can show that $C^{-1} K$ is nonsingular and every eigenvalue of $C^{-1} K$ has negative real part. In particular, $K$ is nonsingular and every eigenvalue of $K$ is a negative real number because $K$ is symmetric. Hence the system (3) or (4) has a unique equilibrium point $\hat{x} = -K^{-1} b \in \mathbb{R}^n$ which is locally asymptotically stable. Moreover, we can show that $\hat{x} \in \text{int} \Omega$ as follows. Suppose that $\hat{x} \not\in \text{int} \Omega$. Let $i_1 = \arg\min_{1 \leq i \leq n} \hat{x}_i$ and $i_2 = \arg\max_{1 \leq i \leq n} \hat{x}_i$. Then, at least one of two inequalities: $\hat{x}_{i_1} \leq L$ and $\hat{x}_{i_1} \geq U$ holds. In the former case, by applying the same argument as in the proof of the first part of Lemma 2, we reach the conclusion that there must exist an integer $i$ such that $(v_{i_1}, v_j) \in E$ and $\hat{x}_i = L$. However, this leads to a contradiction. In the latter case, we reach the conclusion in the same way that there must exist an integer $i$ such that $(v_i, v_{i_1}) \in E$ and $\hat{x}_i = U$. However, this also leads to a contradiction. Therefore we can conclude that $\hat{x} \in \text{int} \Omega$.

The Brouwer degree [12, 13] of the vector field $G(x) = (G_1(x), G_2(x), \ldots, G_n(x))^T$ with respect to $\text{int} \Omega$ and value 0, which is denoted by $d(G, \text{int} \Omega, 0)^2$, satisfies

$$d(G, \text{int} \Omega, 0) = \text{sgn}(C^{-1} K) = \text{sgn}(K) = (-1)^n$$

where the last equality follows from the fact that $n$ eigenvalues of $K$ are real and negative. Let $H(x, s)$ be defined by

$$H(x, s) = sF(x) + (1 - s)G(x).$$

It is obvious that $H(x, s)$ is continuous on $\Omega \times [0, 1]$, $H(x, 0) = G(x)$ and $H(x, 1) = F(x)$. Moreover, we claim that $H(x, s) \neq 0$ for all $x \in \partial \Omega$ and all $l \in (0, 1)$. To see this, suppose that $H(\tilde{x}, \tilde{s}) = 0$ for some $\tilde{x} \in \partial \Omega$ and $\tilde{s} \in (0, 1)$. Then we have

$$F(\tilde{x}) = -\frac{1 - \tilde{s}}{\tilde{s}} G(\tilde{x}). \quad (5)$$

However, since $G_i(\tilde{x}) > 0$ for all $i$ such that $\tilde{x}_i = L$ and $G_i(\tilde{x}) < 0$ for all $i$ such that $\tilde{x}_i = U$, (5) implies that there exists at least one $i$ such that one of two conditions:

1) $\tilde{x}_i = L$ and $F_i(\tilde{x}) < 0$
2) $\tilde{x}_i = U$ and $F_i(\tilde{x}) > 0$

holds, which contradicts Lemma 1. Therefore, two vector fields $F$ and $G$ are homotopic. Since the Brouwer degree is homotopy invariant, we have

$$d(F, \text{int} \Omega, 0) = (-1)^n. \quad (6)$$

This paper employs the notation used in [12]. In Reference [13], the Brouwer degree is denoted by $\deg(G, 0, \text{int} \Omega)$. 

- 360 -
On the other hand, it follows from the definition of the Brouwer degree and Lemma 2 that
\[ d(F, \text{int } \Omega, 0) = m \times (-1)^n \]  
(7)
where \( m \) is the number of equilibrium points of the system (1) in \( \text{int } \Omega \). From (6) and (7), we can conclude that \( m \) must be one, that is, the system (1) has a unique equilibrium point in \( \Omega \). \( \square \)

4.4. Global Asymptotic Stability of the Unique Equilibrium Point

**Theorem 1** The system (1) has a unique equilibrium point in \( \Omega = [L, U]^n \) which is globally asymptotically stable.

**Proof.** The system (1) is a \( C^1 \) cooperative system \([9, 10, 11]\) on \( \Omega \) because
\[ \frac{\partial F_i(x)}{\partial x_j} \geq 0 \]
holds for all \( i \neq j \) and all \( x \in \Omega \). It is known that a \( C^1 \) cooperative system in a closed box \( X \subset \mathbb{R}^n \) has a globally asymptotically stable equilibrium point if and only if two conditions:

1) The system has a unique equilibrium point in \( X \).

2) Every forward semi-orbit has compact closure in \( X \).

hold \([11, \text{Theorem C}]\). Since we have already seen in Lemmas 1 through 4 that the system (1) satisfies these conditions with \( X = \Omega \), it has a unique equilibrium point which is globally asymptotically stable. \( \square \)

5. Conclusion

It was proved in this paper that the nonlinear circuit for solving the maximum flow problem has a unique equilibrium point which is globally asymptotically stable. The main result of this paper is restricted to nonlinear resistors such that the voltage-current characteristic is sigmoidal and cannot be applied to piecewise-linear resistors. Extension of the result to the nonlinear circuit with piecewise-linear resistors is a future problem. Another future problem is to clear the relation between the unique equilibrium point of the circuit and the maximum flow of the graph.

**Acknowledgments**

This work was supported in part by Grant-in-Aid for Scientific Research (C) 21560068 from the Japan Society for the Promotion of Science (JSPS).

**References**


Analysis and Design of Class-DE Amplifier with Nonlinear Shunt Capacitances at Any Duty Ratio

Hiroo Sekiya† and Marian K. Kazimierczuk‡

†Graduate School of Advanced Integration Science, Chiba University
1-33, Yayoi-cho, Inage-ku, Chiba, 263-8522 Japan
‡Department of Electrical Engineering, Wright State University
3640 Colonel Glenn Highway Dayton, OH 45435, USA
Email: sekiya@faculty.chiba-u.jp, marian.kazimierczuk@wright.edu

Abstract—This paper presents an analytical expressions for the class-DE amplifier designs at any switch-on duty ratio, taking into account the nonlinear MOSFET drain-source parasitic capacitances. A design example along with PSPICE-simulation and experimental results indicate the validity of our analysis.

1. Introduction

The class-DE power amplifier [1]–[4] is one of the optimized class-D amplifiers, which satisfy both zero-voltage switching (ZVS) and zero-derivative switching (ZDS) conditions. Because of class-E ZVS/ZDS operation, the class-DE amplifier can operate with high power conversion efficiency at high operating frequencies.

Since class-E ZVS/ZDS should be satisfied with two conditions simultaneously, it is difficult to determine the element values of class-DE amplifiers. Therefore, several analyses were carried out to design it [1]–[4]. It is important to consider the nonlinearity of the shunt capacitances, which are the MOSFET parasitic output capacitance, under high-frequency operations, in particular [2]–[4]. The switch-on duty ratio is also one of the important parameters to design the class-DE amplifier. It is well known that large switch-on duty ratio provides high output power. Conversely, the maximum frequency of the class-DE amplifier becomes low as switch-on duty ratio increases. Most analyses carried out until now assume 25% switch-on duty ratio [1], [3], [4]. It is useful to derive the analytical expressions of the class-DE amplifier at any duty ratio, especially under high-frequency operations.

The purpose of this paper is to obtain the analytical expressions for the class-DE amplifier designs at any switch-on duty ratio, taking into account the nonlinear MOSFET drain-source parasitic capacitances of the diode junction. A design example along with PSPICE-simulation and experimental results indicate the validity of our analysis.

2. Class-DE amplifier

Figure 1 shows the circuit topology of the class-DE amplifier [1]-[4]. The elements $L_f$ and $C_0$ realize a series-tuned circuit whose resonant frequency equals the operating frequency $f$. The element $L$ causes the phase-shift of the output current. Figure 2 shows example waveforms of the class-DE amplifier for the nominal operation. In the operation cycle of the class-DE amplifier, there are two dead-time intervals when both switches are OFF. During the dead-time intervals, the voltage across one of the switches decreases and reaches zero when the switch turns ON. Additionally, the slope of the voltage is also zero at turn-on instant, that is,

$$v_{S1}(\theta) = 0, \quad \frac{dv_{S1}(\theta)}{d\theta}_{\theta=2\pi} = 0, \quad (1)$$

$$v_{S2}(\theta) = 0, \quad \frac{dv_{S2}(\theta)}{d\theta}_{\theta=\pi} = 0. \quad (2)$$

Figure 1: Class-DE amplifier. (a) Circuit topology. (b) Equivalent circuit.

Figure 2: Nominal waveform of class-DE amplifier for $D = 0.25$. 

---

**Abstract**—This paper presents an analytical expressions for the class-DE amplifier designs at any switch-on duty ratio, taking into account the nonlinear MOSFET drain-source parasitic capacitances. A design example along with PSPICE-simulation and experimental results indicate the validity of our analysis.

1. Introduction

The class-DE power amplifier [1]–[4] is one of the optimized class-D amplifiers, which satisfy both zero-voltage switching (ZVS) and zero-derivative switching (ZDS) conditions. Because of class-E ZVS/ZDS operation, the class-DE amplifier can operate with high power conversion efficiency at high operating frequencies.

Since class-E ZVS/ZDS should be satisfied with two conditions simultaneously, it is difficult to determine the element values of class-DE amplifiers. Therefore, several analyses were carried out to design it [1]–[4]. It is important to consider the nonlinearity of the shunt capacitances, which are the MOSFET parasitic output capacitance, under high-frequency operations, in particular [2]–[4]. The switch-on duty ratio is also one of the important parameters to design the class-DE amplifier. It is well known that large switch-on duty ratio provides high output power. Conversely, the maximum frequency of the class-DE amplifier becomes low as switch-on duty ratio increases. Most analyses carried out until now assume 25% switch-on duty ratio [1], [3], [4]. It is useful to derive the analytical expressions of the class-DE amplifier at any duty ratio, especially under high-frequency operations.

The purpose of this paper is to obtain the analytical expressions for the class-DE amplifier designs at any switch-on duty ratio, taking into account the nonlinear MOSFET drain-source parasitic capacitances of the diode junction. A design example along with PSPICE-simulation and experimental results indicate the validity of our analysis.

2. Class-DE amplifier

Figure 1 shows the circuit topology of the class-DE amplifier [1]-[4]. The elements $L_f$ and $C_0$ realize a series-tuned circuit whose resonant frequency equals the operating frequency $f$. The element $L$ causes the phase-shift of the output current. Figure 2 shows example waveforms of the class-DE amplifier for the nominal operation. In the operation cycle of the class-DE amplifier, there are two dead-time intervals when both switches are OFF. During the dead-time intervals, the voltage across one of the switches decreases and reaches zero when the switch turns ON. Additionally, the slope of the voltage is also zero at turn-on instant, that is,

$$v_{S1}(\theta) = 0, \quad \frac{dv_{S1}(\theta)}{d\theta}_{\theta=2\pi} = 0, \quad (1)$$

$$v_{S2}(\theta) = 0, \quad \frac{dv_{S2}(\theta)}{d\theta}_{\theta=\pi} = 0. \quad (2)$$

---

**Abstract**—This paper presents an analytical expressions for the class-DE amplifier designs at any switch-on duty ratio, taking into account the nonlinear MOSFET drain-source parasitic capacitances. A design example along with PSPICE-simulation and experimental results indicate the validity of our analysis.

1. Introduction

The class-DE power amplifier [1]–[4] is one of the optimized class-D amplifiers, which satisfy both zero-voltage switching (ZVS) and zero-derivative switching (ZDS) conditions. Because of class-E ZVS/ZDS operation, the class-DE amplifier can operate with high power conversion efficiency at high operating frequencies.

Since class-E ZVS/ZDS should be satisfied with two conditions simultaneously, it is difficult to determine the element values of class-DE amplifiers. Therefore, several analyses were carried out to design it [1]–[4]. It is important to consider the nonlinearity of the shunt capacitances, which are the MOSFET parasitic output capacitance, under high-frequency operations, in particular [2]–[4]. The switch-on duty ratio is also one of the important parameters to design the class-DE amplifier. It is well known that large switch-on duty ratio provides high output power. Conversely, the maximum frequency of the class-DE amplifier becomes low as switch-on duty ratio increases. Most analyses carried out until now assume 25% switch-on duty ratio [1], [3], [4]. It is useful to derive the analytical expressions of the class-DE amplifier at any duty ratio, especially under high-frequency operations.

The purpose of this paper is to obtain the analytical expressions for the class-DE amplifier designs at any switch-on duty ratio, taking into account the nonlinear MOSFET drain-source parasitic capacitances of the diode junction. A design example along with PSPICE-simulation and experimental results indicate the validity of our analysis.

2. Class-DE amplifier

Figure 1 shows the circuit topology of the class-DE amplifier [1]-[4]. The elements $L_f$ and $C_0$ realize a series-tuned circuit whose resonant frequency equals the operating frequency $f$. The element $L$ causes the phase-shift of the output current. Figure 2 shows example waveforms of the class-DE amplifier for the nominal operation. In the operation cycle of the class-DE amplifier, there are two dead-time intervals when both switches are OFF. During the dead-time intervals, the voltage across one of the switches decreases and reaches zero when the switch turns ON. Additionally, the slope of the voltage is also zero at turn-on instant, that is,

$$v_{S1}(\theta) = 0, \quad \frac{dv_{S1}(\theta)}{d\theta}_{\theta=2\pi} = 0, \quad (1)$$

$$v_{S2}(\theta) = 0, \quad \frac{dv_{S2}(\theta)}{d\theta}_{\theta=\pi} = 0. \quad (2)$$
Table 1: Switching Pattern

<table>
<thead>
<tr>
<th>Intervals</th>
<th>( D_{r1} )</th>
<th>( D_{r2} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( 0 \leq \theta &lt; 2\pi )D</td>
<td>ON</td>
<td>OFF</td>
</tr>
<tr>
<td>( 2\pi )D ( \leq \theta &lt; \pi )</td>
<td>OFF</td>
<td>OFF</td>
</tr>
<tr>
<td>( \pi \leq \theta &lt; \pi + 2\pi )D</td>
<td>OFF</td>
<td>ON</td>
</tr>
<tr>
<td>( \pi + 2\pi )D ( \leq \theta &lt; 2\pi )</td>
<td>OFF</td>
<td>OFF</td>
</tr>
</tbody>
</table>

These conditions are called as the class-E zero-voltage switching (ZVS) and zero-derivative switching (ZDS) conditions. Because of the class-E ZVS/ZDS conditions, the switching power losses in the class-DE amplifier become zero. Therefore, the high power conversion efficiency can be achieved at high operating frequencies.

3. Waveform and Design Equations

3.1. Assumptions

The analysis in this paper is based on the following assumptions.

(a) The shunt capacitances consist of only the drain-source parasitic capacitances of the MOSFETs whose characteristic is expressed as

\[
C_{ds} = \frac{C_{j0}}{1 + \frac{v_S}{V_{bi}}},
\]

where \( v_S \) is the drain-to-source voltage, \( V_{bi} \) is the built-in potential, which typically ranges from 0.5 to 0.9 V, and \( C_{j0} \) is the capacitance at \( v_S = 0 \) [2], [3].

(b) Both MOSFETs are identical and are modeled as ideal switches and drain-source parasitic capacitances.

(c) All passive elements except shunt capacitances are linear elements and do not have parasitic resistances.

(d) The loaded quality factor of the resonant filter \( Q = \omega L_0/R \) is high enough to generate nearly pure sinusoidal output current. The current through the \( L_f \ C_0 \) circuit and the load resistance is sinusoidal at the operating frequency \( f \),

\[
i_o = I_m \sin(\varphi + \theta),
\]

where \( \varphi = \omega t = 2 \pi f t \) represents the angular time.

(e) The switching pattern is the same as that given in Table 1.

(f) At the end of the dead time, both the switch voltages \( v_{S1} \) and \( v_{S2} \) satisfy class-E ZVS/ZDS conditions.

Under the above assumptions, the equivalent model of the class-DE amplifier can be obtained as shown in Fig. 1(b).

3.2. Voltage Waveforms

The analysis for steady state is performed in the interval \( 0 \leq \theta < 2\pi \). The following relation between \( v_{S1} \) and \( v_{S2} \) is always valid:

\[
\frac{v_{S1}(\theta)}{V_{DD}} = \frac{1}{2} + \frac{1}{2} \left\{ \frac{2}{1 + \frac{V_{DD}}{V_{bi}}} \left[ \frac{1}{\sqrt{1 + \frac{V_{DD}}{V_{bi}}} - 1} \right] \right\}^{1/2} + \frac{1}{2} \left\{ \frac{2}{1 + \frac{V_{DD}}{V_{bi}}} \left[ \frac{1}{\sqrt{1 + \frac{V_{DD}}{V_{bi}}} - 1} \right] \right\}^{1/2}.
\]

The slope of the switch voltage \( v_{S1} \) is expressed as

\[
\frac{dv_{S1}(\theta)}{d\varphi} = \frac{I_m}{\omega(C_{ds1} + C_{ds2})} \sin(\varphi + \theta).
\]

From the class-E ZVS condition for \( v_{S1} \) in (1), we obtain \( \varphi = 0 \) and \( \varphi = \pi \). In this paper, we consider that the amplitude of the output current \( I_m \) is positive. Therefore, the phase difference is determined as

\[
\varphi = \pi.
\]

Similarly, (7) is valid for the class-E ZDS condition for \( v_{S2} \). For \( 0 \leq \theta < 2\pi \), the switch \( S_1 \) is on and the switch \( S_2 \) is off. Therefore, the switch voltages are constant,

\[
v_{S1}(\theta) = 0, \quad v_{S2}(\theta) = V_{DD}.
\]

From (3), (6), and \( v_{S1}(2\pi) = 0 \), we obtain

\[
2\omega C_{j0} V_{bi} \left[ \sqrt{1 + \frac{V_{DD}}{V_{bi}}} \left[ \frac{1}{\sqrt{1 + \frac{V_{DD}}{V_{bi}}} - 1} \right] \right] = I_m [\cos(2\pi) \cos \varphi].
\]

The switch voltages are \( v_{S1}(\theta) = V_{DD} \) and \( v_{S2}(\theta) = 0 \) because of the class-E ZVS condition in (2). By substituting \( \varphi = 0 \) and \( v_{S1} = 0 \), the amplitude \( I_m \) is

\[
I_m = \frac{4\omega C_{j0} V_{bi} \left[ \sqrt{1 + \frac{V_{DD}}{V_{bi}}} \left[ \frac{1}{\sqrt{1 + \frac{V_{DD}}{V_{bi}}} - 1} \right] \right]}{1 + \cos(2\pi)}.
\]

From (9) and (10), we have

\[
\frac{\sqrt{1 + \frac{V_{DD}}{V_{bi}} V_{S1} V_{DD}}}{V_{bi} V_{DD}} \left[ \sqrt{1 + \frac{V_{DD}}{V_{bi}} V_{S1} V_{DD}} \frac{V_{DD}}{V_{bi}} V_{DD} \right] + \frac{\sqrt{1 + \frac{V_{DD}}{V_{bi}} V_{S1} V_{DD}}}{V_{bi} V_{DD}} \left[ \sqrt{1 + \frac{V_{DD}}{V_{bi}}} \left[ \frac{1}{\sqrt{1 + \frac{V_{DD}}{V_{bi}}} - 1} \right] \right] = 0.
\]

We can solve (11) for \( v_{S1} \) analytically,
where the sign “±” of the second term on the right-hand side changes at the boundary of $v_{S1}/V_{DD} = 1/2$. Namely, the sign is “−” for $2D \leq \cos^{-1}((1 \cos(2D))/2)$ and “+” for $\cos^{-1}((1 \cos(2D))/2) \leq v_{S1} < 1$. For $\theta < -2D$, the switch voltages and currents have the following relationships because of the symmetry of the operation of the switches,

$$v_{S1}(\theta) = v_{S2}(\theta), \quad v_{S2}(\theta) = v_{S1}(\theta). \quad (13)$$

### 3.3. DC Supply Current and Output Voltage, Current, and Power

The dc supply current $I_D$ is given as the average of the supply current flowing from the dc voltage source $V_{DD}$,

$$I_D = \frac{1}{2} \int_0^{2\pi} [i_{S2}(\theta) + i_{CS2}(\theta)] d\theta$$

$$= 2\omega C_{c0} V_{bi} \left[ \frac{1 + V_{DD}}{V_{bi}} \frac{1}{[1 + \cos(2D)]} \right]. \quad (14)$$

Note that the average value of the current through the shunt capacitance $i_{CS2}$ means the amount of charging/discharging electric charge. Therefore, it is always zero. From (10) and (14), the normalized amplitude of the output current is expressed as

$$I_{on}/I_D = \sqrt{2} \cos(2D). \quad (15)$$

Using (15), we obtain the dc supply power as

$$P_{dc} \equiv V_{DD} I_D = I_{on} V_{DD} [1 - \cos(2D)]. \quad (16)$$

The output power $P_o$ is given by

$$P_o = \frac{I_{on}^2 V_m}{2} = \frac{R I_{on}^2}{2} = \frac{V_m^2}{2R}. \quad (17)$$

Ideally, the power conversion efficiency of the class-DE amplifier is 100% on the nominal operation,

$$P_{dc} = P_o. \quad (18)$$

From (16), (17), and (18), we can obtain the amplitude of the output voltage normalized with respect to the dc supply voltage as

$$V_m/V_{DD} = \frac{1}{2} \cos(2D). \quad (19)$$

Figure 3(a) shows the normalized amplitudes of the output current as a function of the switch-on duty ratio $D$.

From (17) and (19), we have

$$P_o = \frac{R I_{on}^2}{2} = \frac{V_m^2}{2R} = \frac{V_{DD}^2}{2R} \frac{1}{2} \frac{\cos(2D)}{2}. \quad (20)$$

Figure 3(b) shows the normalized output power as a function of the switch-on duty ratio. It is seen from Fig. 3(b) that the output power increases as the switch-on duty ratio increases.

![Normalized Amplitude of Output Current](image1)

![Normalized Output Power](image2)

#### 3.4. Voltage Across the Load Reactance

The fundamental component of the voltage $v_L(\theta)$ across the reactance $L$ is expressed as

$$v_L(\theta) = V_L(\cos \theta), \quad (21)$$

where $V_L = \omega L I_m$. (22)

From (22) and $V_m = RI_m$, we obtain

$$\frac{V_m}{V_m} = \frac{\omega L}{R}. \quad (23)$$

The normalized magnitude $V_L/V_{DD}$ is derived from the Fourier integral

$$\frac{V_L}{V_{DD}} = \frac{1}{2} \int_0^{2\pi} \frac{v_{S1}(\theta)}{V_{DD}} (\cos \theta) d\theta. \quad (24)$$

(24) can be solved numerically. Here, the function $H(m, V_{DD}/V_{bi}, D)$ is defined as

$$H \equiv \frac{V_L}{V_{DD}} = \frac{1}{2} \int_0^{2\pi} \frac{v_{S1}(\theta)}{V_{DD}} (\cos \theta) d\theta. \quad (25)$$

Figure 4 shows plots of the function $H$. In order to illustrate Fig. 4(a), the trapezoidal rule with $2/10000$ of time-step of $\theta$ is used to calculate the integration of (24).

#### 3.5. Design Equations

From (20), the load resistance $R$ is given as

$$R = \frac{V_{DD}^2}{2P_o} \frac{1}{2} \frac{\cos(2D)}{2}. \quad (26)$$

From the definition of the loaded-quality factor $Q$, the inductance $L_0$ is given as

$$L_0 = \frac{Q R}{\omega} = \frac{Q R}{2f}. \quad (27)$$

From (19), (23), and (25), the inductance $L$ is

$$L = \frac{Q}{2} \frac{R H}{\cos(2D)}. \quad (28)$$
From (27) and (28), the inductance \( L_f \) is obtained as

\[
L_f = L_0 \quad L = \frac{R}{2f} \left[ \frac{Q}{1} \frac{H}{\cos(2D)} \right].
\]

The identical resonant filter with the resonant frequency \( f = \omega/2 \) is realized by \( L_f \) and \( C_0 \). From \( f = 1/(2 \sqrt{L_f C_0}) \) and (29), the resonant capacitance \( C_0 \) is expressed analytically as

\[
C_0 = \frac{1}{2fR} \left[ \frac{Q}{1} \frac{H}{\cos(2D)} \right].
\]

In these design equations, we need numerical calculations for obtaining \( H \), which is used for the derivation of \( C_0 \) in (30).

4. Design Example and Experimental Verification

The design example with the discrete MOSFET devices is given. The design specifications are: operating frequency \( f = 1 \) MHz, dc supply voltage \( V_{DD} = 24 \) V, output resistance \( R = 50 \) , and loaded quality factor \( Q = 8 \).

It is considered that the IRF 530 MOSFETs are used as the switching devices. The values of grading coefficient \( m \), the built-in potential \( V_{bi} \), and \( C_{00} \) of IRF530 MOSFET are obtained as \( V_{bi} = 0.8 \) V and \( C_{00} = 1.03 \) nF from the PSPICE model. From (10) and (20), the duty ratio \( D \) is obtained as \( D = 0.35 \). Therefore, \( L_0 = 63.6 \) H is obtained from (27). From the numerical calculations of (25), \( H \) is 0.366. Therefore, the resonant capacitance \( C_0 \) is 427 pF from (30).

Figure 5 shows the PSPICE-simulated waveforms and the experimental waveforms. From Fig. 5, it can be confirmed that both the PSPICE-simulated waveforms and the experimental waveforms satisfy the class-E ZVS/ZDS conditions. Table 2 gives the analytical predictions and experimental measurements. From this table, it is seen that the experimental measurements agreed with the analytical predictions quantitatively. The results in Figure 5 and Table 2 indicate the validity of the analytical expressions in this paper. The laboratory measurements showed the 95.7% power conversion efficiency with 1.3 W output power.

Table 2: Analytical Predictions and Experimental Measurements

<table>
<thead>
<tr>
<th>( f )</th>
<th>Calculated</th>
<th>Measurement</th>
<th>Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f )</td>
<td>1 MHz</td>
<td>1.02 MHz</td>
<td>2.0%</td>
</tr>
<tr>
<td>( V_{DD} )</td>
<td>24 V</td>
<td>24 V</td>
<td>0.0%</td>
</tr>
<tr>
<td>( D )</td>
<td>0.35</td>
<td>0.35</td>
<td>0.0%</td>
</tr>
<tr>
<td>( C_0 )</td>
<td>427 pF</td>
<td>424 pF</td>
<td>0.7%</td>
</tr>
<tr>
<td>( L_0 )</td>
<td>63.6 H</td>
<td>62.0 H</td>
<td>2.5%</td>
</tr>
<tr>
<td>( R )</td>
<td>50</td>
<td>49.8</td>
<td>0.6%</td>
</tr>
<tr>
<td>( Q )</td>
<td>8</td>
<td>7.98</td>
<td>0.3%</td>
</tr>
<tr>
<td>( P_0 )</td>
<td>1.46 W</td>
<td>1.32 W</td>
<td>10.6%</td>
</tr>
</tbody>
</table>

References


Soft Fault Diagnosis of Nonlinear Analog Circuits Using the Continuation Approach

Michał Tadeusiewicz and Stanisław Halgas

Dept. of Electrical, Electronic, Computer and Control Eng. Technical University of Łódź, ul. Stefanowskiego 18/22 90-924, Łódź, Poland
Email: michal.tadeusiewicz@p.lodz.pl, stanislaw.halgas@p.lodz.pl

Abstract—This paper offers a method for multiple soft fault diagnosis of nonlinear analog circuits using the continuation (homotopy) approach. The considered soft faults include both slight and considerable deviations from the tolerance ranges. The method enables us to locate the faulty elements and evaluate the parameters both the faulty elements and the other elements considered as possible faulty. An important property of the method is its orientation on finding multiple solutions of the nonlinear diagnostic equation. As a result, it can find different sets of the parameters which satisfy the diagnostic test, rather than one specific set. The developed method is especially useful at the pre-production stage, where corrections of the technological process are possible and the diagnostic time is not crucial. To illustrate the proposed approach, a numerical example is given.

1. Introduction

Fault diagnosis of analog circuits is an important problem in the design and testing of electronic devices [1]-[7]. If circuit simulations take place after any testing, the diagnostic method is classified as the simulation-after-test (SAT) approach. A fault is called soft when the parameter deviates from its tolerance range, but does not produce a short circuit or an open circuit. Otherwise, when a fault leads to open nodes, shorts between nodes or other topological changes in a circuit, it is called catastrophic. Generally, fault diagnosis includes detecting faulty circuits, locating (identification) faulty elements and determining their parameters.

During the last several years, many methods devoted to soft fault diagnosis of analog circuits have been developed [6]-[13]. Appropriate tools for soft-fault diagnosis are SAT methods, based on measurements at some test points and analyses of the circuit under test. Most of the works address only the case when just one element is faulty. Several papers have been focused on the multiple fault diagnosis, e.g. [9]-[13].

Although many achievements in this field have been made, the problem is still open and no fully automatic method is available for analog circuits.

Many diagnostic methods employ linearized test equations. Consequently, they work correctly only if the parameters are slightly drifted from their tolerance ranges. The approach proposed in this paper does not utilize such simplification, enabling us to diagnose larger faults and evaluate the parameters of all elements in the set of elements considered as possible faulty. The number of faulty elements in this set does not influence significantly the time consumed by the method.

An important property of the method proposed in this paper is its orientation on finding multiple solutions of the nonlinear diagnostic equation. As a result it can find different sets of parameters which satisfy the test, rather than one specific set.

Location (identification) of the faulty elements and determination of their parameters play a very important role at the pre-production stage, where corrections of the technological process are possible. In this way some defects of the technological process can be eliminated, leading to its improvement. The approach proposed in this paper is especially useful at this stage of the production. In such a case, the time consumed by the diagnostic method is not crucial.

Let us consider a nonlinear resistive circuit driven by DC independent sources, e.g. a diode-transistor one. Suppose that n circuit elements, specified by parameters $x_1,\ldots,x_n$ such as resistances, factors of controlled sources, $\beta$ forward factors of bipolar transistors, are considered as possible faulty. We wish to find the actual values of all the parameters. For this purpose a diagnostic test must be performed leading to a systems of n nonlinear equations in n unknown variables $x_1,\ldots,x_n$.

The test equation can be presented in the form

$$f(\hat{x}) = u,$$

where $\hat{x} = [x_1 \ldots x_n]^T \in \mathbb{R}^n$ is a vector of the parameters, $u = [u_1 \ldots u_n]^T \in \mathbb{R}^n$ is a vector of the measured signals, $f(\hat{x}) = [f_1(\hat{x}) \ldots f_n(\hat{x})]^T$ is a nonlinear function mapping $\mathbb{R}^n$ into $\mathbb{R}^n$. Generally it is impossible to formulate the nonlinear functions $f_i(\hat{x})$, $i = 1,\ldots,n$, in explicit analytical form, but the values of $f_i(x_1,\ldots,x_n)$ and their derivatives with respect to $x_j$ ($i,j = 1,\ldots,n$) can be found numerically for given values of $x_1,\ldots,x_n$. For example, the test can be arranged as follows. Assume that the nonlinear circuit under test has one accessible input.
node and \( r \) accessible output nodes. We apply a DC voltage source to the input node (see Fig. 1) and for \( K \) different values of this source measure the corresponding output voltages \( v_1, \ldots, v_r \). As a result we obtain \( rK \) values of the output voltages: \( v_1^{(1)} = u_1, \ldots, v_1^{(K)} = u_r \), \( \ldots \), \( v_r^{(1)} = u_{(K-1)r+1}, \ldots, v_r^{(K)} = u_{Kr} \). This test enables us to perform the diagnosis of a circuit with \( n \leq rK \) elements considered as possible faulty. Some other appropriate variants of the test can be arranged.

In this paper we propose the continuation (homotopy) method [14]-[15] for finding multiple solutions of the diagnostic equation (1). The idea of the method is as follows. To solve an equation \( g(x) = 0 \), a parameter \( \lambda \) is embedded into this equation to obtain a new one

\[
h(x, \lambda) = 0 ,
\]

called a homotopy equation, such that for \( \lambda = 0 \), \( h(x, \lambda) = 0 \) can be easily solved and for \( \lambda = 1 \) it reduces to the original equation \( g(x) = 0 \). The augmented equation (2) is deformed as the parameter \( \lambda \) varies. At each stage of the deformation the corresponding solution is calculated, taking into account the previous solution. The Newton-Raphson method can be used for this purpose. As a result a solution (homotopy) path is traced and each intersection of the path with \( \lambda = 1 \) plane is a solution of the diagnostic equation.

2. Fault Diagnosis

Let us consider a circuit belonging to the class defined in Section 1 and assume that the parameters \( x_1, \ldots, x_n \) have their nominal values \( \bar{x} = x^{(0)} \). We analyse this circuit and find the output signals leading to \( u = u^{(0)} \). Thus, the equation

\[
f'(x^{(0)}) = u^{(0)}
\]

holds. For the diagnostic equation (1) repeated below

\[
f(\bar{x}) = u ,
\]

we form the Newton homotopy [14]

\[
h(\bar{x}, \lambda) = 0 ,
\]

where

\[
h(\bar{x}, \lambda) = f(\bar{x}) - u^{(0)} - \lambda(u - u^{(0)}) .
\]

Note that for \( \lambda = 0 \) the equation specified by (5) and (6) reduces to the equation \( f(\bar{x}) = u^{(0)} \) having the known solution \( x^{(0)} \), whereas for \( \lambda = 1 \) it becomes the original diagnostic equation. Denote \( x_{n+1} = \lambda \) and form an augmented vector \( x = [x_1 \cdots x_n x_{n+1}]^T \). Then the homotopy equation can be written as

\[
h(x) = 0 .
\]

Equation (7) represents a system of \( n \) nonlinear scalar equations in \( n + 1 \) variables. As \( x_{n+1} = \lambda \) varies, starting from \( x_{n+1} = 0 \), the solution of (7) traces a homotopy path. Each intersection of this path with the \( x_{n+1} = 1 \) plane is a solution of the diagnostic equation (1). If there are several intersection points, then the test equation has several different solutions. Let us parameterize the path with respect to arc length [15]-[16]. This is the standard treatment of representing a path in parametric coordinates \( x = x(s) \). If

\[
\sum_{i=1}^{n+1} \left( \frac{dx_i}{ds} \right)^2 = 1 ,
\]

then \( s \) is the arc length of the path. Using the parameterization we form the set of equations

\[
h(x) = 0 ,
\]

\[
\sum_{i=1}^{n+1} \left( \frac{dx_i}{ds} \right)^2 = 1 .
\]

As \( s = s_j \), then \( x = x^{(j)} \). The derivative \( \frac{dx_i}{ds} \) at \( s = s_{j+1} \) labeled \( \frac{dx_i}{ds}^{(j+1)} \), can be expressed in terms of \( x_i^{(j+1)} \) and \( x_i^{(j)} \) using the approximate formula

\[
\frac{dx_i}{ds}^{(j+1)} = \frac{1}{h} (x_i^{(j+1)} - x_i^{(j)}) , \quad i = 1, \ldots, n ,
\]

where \( h = s_{j+1} - s_j \). Using (10) we formulate the set of equations (8)-(9) at \( s = s_{j+1} \) as follows

\[
w(x^{(j+1)}) = 0 ,
\]

where

\[
\begin{bmatrix}
\hat{h}_1 [x_1^{(j+1)}, \ldots, x_n^{(j+1)}]
\vdots
\hat{h}_n [x_1^{(j+1)}, \ldots, x_n^{(j+1)}]
\sum_{i=1}^{n+1} (x_i^{(j+1)} - x_i^{(j)})^2 - h^2
\end{bmatrix}
\]

To simplify the notation we denote \( [x_1^{(j+1)}, \ldots, x_n^{(j+1)}]^T = z = [z_1, \ldots, z_n]^T \) and \( x_i^{(j)} = X_i \). As a result the equation (11) becomes

\[
-367 -
To solve the equation (13) we apply the Newton-Raphson method

\[
\mathbf{z}^{(m+1)} = \mathbf{z}^{(m)} - J^{-1}(\mathbf{z}^{(m)}) \mathbf{w}(\mathbf{z}^{(m)}),
\]

where \( m \) is the index of iteration. The Jacoby matrix is

\[
J(\mathbf{z}^{(m)}) = \begin{bmatrix}
\frac{\partial h_1}{\partial z_1} & \cdots & \frac{\partial h_1}{\partial z_n} & \frac{\partial h_1}{\partial z_{n+1}} \\
\vdots & \ddots & \vdots & \vdots \\
\frac{\partial h_n}{\partial z_1} & \cdots & \frac{\partial h_n}{\partial z_n} & \frac{\partial h_n}{\partial z_{n+1}} \\
2(z_1 - X_1) & \cdots & 2(z_n - X_n) & 2(z_{n+1} - X_{n+1})
\end{bmatrix}
\]

where \( \mathbf{z} = \mathbf{z}^{(m)} \). Using (6) and (7) we find

\[
\frac{\partial h_i}{\partial z_i} = u_i^{(0)} - u_i, \quad \frac{\partial h_i}{\partial z_j} = u_j^{(0)} - u_j.
\]

Moreover, for \( k, l = 1, \ldots, n \) it holds

\[
\frac{\partial h_k}{\partial z_l} = \frac{\partial f_k}{\partial z_l} |_{\mathbf{z}^{(m)}}.
\]

The derivatives \( df_k / dx_l \), \( k, l = 1, \ldots, n \) cannot be computed directly, because the function \( f \) is not given in explicit analytical form. To find the derivatives (17) we set \( x_i = z_i^{(m)} \), \( i = 1, \ldots, n \) and perform the DC and sensitivity analyses of the tested circuit at \( K \) values of the input voltages. In this way we find \( f(\mathbf{z}^{(m)}) \) and the sensitivities of the output voltages \( v_1, \ldots, v_r \) with respect to the parameters \( x_1, \ldots, x_n \)

\[
\frac{\partial v_d}{\partial x_l} |_{\mathbf{z}^{(m)}} = \frac{\partial f_k}{\partial z_l} |_{\mathbf{z}^{(m)}},
\]

where \( v_d \) is an appropriate output voltage. The vector \( \mathbf{w}(\mathbf{z}^{(m)}) \) that appears in (14) is given by

\[
\mathbf{w}(\mathbf{z}^{(m)}) = \begin{bmatrix}
f_1(z_1^{(m)}, \ldots, z_n^{(m)}) - u_1^{(0)} - z_1^{(m)}(u_1^{(0)} - u_1) \\
f_2(z_1^{(m)}, \ldots, z_n^{(m)}) - u_2^{(0)} - z_2^{(m)}(u_2^{(0)} - u_2) \\
\vdots \\
n_{n+1}^{(m)}(z_1^{(m)} - X_1)^2 - h_2^2
\end{bmatrix}
\]

To perform the computation process efficiently some strategy of changing the step \( h \) has been worked out and applied.

The proposed method was implemented on a computer using Delphi and tested on several circuits. The computations were executed using PC Pentium Core 2 Duo E6400.

3. Numerical Example

Let us consider the transistor circuit shown in Fig. 2. The nominal parameters, having the tolerances 5%, are indicated on the circuit diagram. To test the proposed method two series of diagnoses were carried out. In the first series all resistors \( R_1 \) - \( R_7 \) of the circuit were considered as possible faulty. Under the assumption that the nodes A and B are accessible for measurement we performed the diagnostic test, as described in Section 1, for different values of the input voltages: \( v_{in}^{(1)} = 15V \), \( v_{in}^{(2)} = 15V \), \( v_{in}^{(1)} = 5V \), \( v_{in}^{(2)} = 15V \); \( v_{in}^{(1)} = 15V \), \( v_{in}^{(2)} = 7V \); \( v_{in}^{(1)} = 7V \), \( v_{in}^{(2)} = 4V \). For these sets of input voltages we found (measured) 7 values of the output voltages. The measurement accuracy was assumed to be 1\muV.

![Fig. 2. A circuit for example](image)

We considered 20 different sets of the circuit parameters. In most of the cases, the method gave more than one set of the parameters which satisfied the test. Some of them can be discarded due to negative values of certain resistances. In each case, the correct set of parameters appeared in the provided results. One out of 20 considered cases is described below.

The parameters are as follows: \( R_1 = 108 \Omega \) (−60%), \( R_2 = 5 \Omega \) (−77%), \( R_3 = 21k \Omega \) (−51%), \( R_4 = 3k \Omega \) (−56%), \( R_5 = 1.89k \Omega \) (−30%), \( R_6 = 127 \Omega \) (−53%), \( R_7 = 125 \Omega \) (−54%).

Thus, all 7 elements of the set are faulty. The proposed method gives two sets of parameters which satisfy the test: \( R_1^{(1)} = 107.7 \Omega \), \( R_2^{(1)} = 4.98k \Omega \), \( R_3^{(1)} = 20.9k \Omega \); \( R_4^{(1)} = 2.99k \Omega \), \( R_5^{(1)} = 1.89k \Omega \), \( R_6^{(1)} = 127 \Omega \), \( R_7^{(1)} = 125 \Omega \) and \( R_1^{(2)} = 58.8k \Omega \), \( R_2^{(2)} = 789 \Omega \), \( R_3^{(2)} = 11.8k \Omega \), \( R_4^{(2)} = 495 \Omega \), \( R_5^{(2)} = 792 \Omega \), \( R_6^{(2)} = 51 \Omega \), \( R_7^{(2)} = 64 \Omega \).

Thus, the circuit is detected as faulty. The time consumed by the method is 14s. The homotopy path is a closed curve, its projection on \( \lambda - R_1 \) plane is shown in Fig. 3.
The second series of the diagnoses is discussed below. We consider all the resistors \( R_1 \) to \( R_7 \) and \( \beta \) forward factors of the transistors \((\beta_1 \) and \( \beta_2 \)) as possibly faulty. The tolerances of all the parameters are 5%. We extend the previous test by adding a set of the input voltages: 

\[
\begin{align*}
V_1^{(0)} &= 10 \text{ V}, \quad V_2^{(2)} &= 0 \text{ V}
\end{align*}
\]

and find (measure) 9 values of the output voltages. The measurement accuracy is assumed to be 1\( \mu \text{V} \). We considered 10 different sets of the parameters. In most of the cases the method gave one set of the parameters which satisfied the test. In each case the correct set appeared in the provided results. One of the cases is described below. The parameters are as follows: 

\[
\begin{align*}
 R_1 &= 340 \text{k}\Omega \quad (26\%) , \quad R_2 = 12 \text{k}\Omega \quad (\sim 45\%) , \quad R_3 = 44 \text{k}\Omega \quad (2\%) , \quad R_4 = 7.1 \text{k}\Omega \quad (4\%) , \quad R_5 = 2.5 \text{k}\Omega \quad (\sim 7\%) , \\
 R_6 &= 265 \Omega \quad (\sim 2\%) , \quad R_7 = 262 \Omega \quad (\sim 3\%) , \quad \beta_1 = 340 \quad (\sim 15\%) , \quad \beta_2 = 340 \quad (\sim 15\%)
\end{align*}
\]

Thus, 5 elements are faulty. The proposed method gives just one (correct) set of the parameters. The time consumed by the method is 27s. It is satisfactory at the pre-production stage. For illustration, the projection of the obtained homotopy path on \( \lambda - \beta_2 \) plane is shown in Fig. 4. There is one point of intersection of the curve with the vertical line \( \lambda = 1 \), corresponding to \( \beta_2 = 340 \).

4. Conclusion

The method enables us to locate multiple soft faults of nonlinear analog circuits and evaluate the parameters of all elements considered as possible faulty. The new aspect of the proposed approach is its orientation on finding different sets of parameters, which satisfy the diagnostic test, rather than one specific set. The soft faults accepted by the method may be caused both by slight and considerable deviations of the circuit parameters.

Acknowledgments

This work was supported by the Ministry of Education and Science under Grant N N515 066435.

References

Single-Tone Moments Based Adjoint Sensitivity Analysis of Nonlinear Intermodulation Distortion in RF Circuits

Dani Tannir and Roni Khazaka

Department of Electrical and Computer Engineering, McGill University
Montreal, QC, Canada, H3A 2A7
Email: dani.tannir@mail.mcgill.ca, roni.khazaka@mcgill.ca

Abstract—Intermodulation distortion analysis is one of the main computational bottlenecks in the simulation of Radio Frequency circuits. Recently, an efficient moments based approach for computing the third order intercept point (IP3) using moments analysis with only single-tone inputs was presented. This approach did not, however, provide any sensitivity information which is critical for design centering, optimization and yield analysis applications among others. In this paper, we propose an efficient method for computing the sensitivity of IP3 using single-tone adjoint moments analysis. The adjoint method finds the sensitivity of one output with respect to all variables with only minimal additional computation cost to the original algorithm, thereby making it very efficient.

1. Introduction

One of the main specifications that engineers must account for in the design of Radio Frequency (RF) front ends is the linearity of the core blocks such as low noise amplifiers (LNAs) and mixers. Measuring the amount of the third order nonlinear distortion is of particular importance since its effects will appear in the passband of the system and are thus very difficult to filter out. This distortion in turn could lead to many undesirable effects such as gain compression and adjacent channel interference [1]. The key figure of merit for quantifying the third order nonlinear distortion is the third order intercept point (IP3) [1]. Determining the value of IP3 through simulation has, however, been one of the main bottlenecks in the design automation process due to the multi-tone input requirement which considerably slows down steady-state simulators based on techniques such as the Harmonic Balance method [2].

In [3], [4], an efficient method for computing the value of IP3 was presented based on the computation of the Harmonic Balance moments [5] from the general Harmonic Balance equations with only a single-tone input and without the need to find the solution of the equations. This reduced the computation cost to that of solving a system of sparse, linear equations, and also resulted in a significantly smaller system than that with multi-tone inputs. This is especially the case with mixer circuits where only two tones (1 RF tone in addition to that of the Local Oscillator) are required instead of the typical three tones. However, this approach did not provide any insight into the sensitivity of IP3 with respect to various circuit parameters. For any sensitivity analysis to be performed, only brute-force perturbation could be employed, which is very inefficient. In [6], a new approach for computing the sensitivity of IP3, based on the adjoint sensitivity method [7], was presented. The method in [6], however, is limited in its application to the multi-tone moments method presented in [8].

In this paper, we extend the method in [6] to the computation of IP3 sensitivity using the single-tone moments method. The adjoint method has been a classical tool for the sensitivity analysis of both linear and nonlinear circuits, including those operating under large signal periodic and almost-periodic conditions as is the case with the Harmonic Balance method [9]. The approach proposed in this paper benefits from the same computational cost advantage as [3], [4], while also providing the sensitivity of IP3 with respect to all the circuit parameters. This would provide a key advantage for circuit optimization, design space exploration and design centering analysis. It is to be noted that similarly to the approach in [3], [4] this method is general and easily automated for any arbitrary circuit topology. Finally, since the adjoint moments are computed using the same set of linear equations that are used to determine the Harmonic Balance moments, the CPU cost of the operation is reduced to that of finding three additional moments over the CPU cost of the method in [3], which is very cheap.

2. System Formulation

In this section, we present the general formulation of the nonlinear Harmonic Balance equations followed by overviews of the moments computation algorithm and of the method for computing IP3 using single-tone moments analysis. This will provide the necessary background information for the new single-tone sensitivity analysis method presented in Section 3.

2.1. Harmonic Balance Formulation

The Harmonic Balance equations for a nonlinear system are of the form [10]

\[ AX + F(X) = B_{DC} + \alpha B_{RF} + \beta B_{LO}, \]  

where
• $X \in \mathbb{R}^{N_h}$ is a vector of unknown cosine and sine coefficients for each of the variables in $x(t)$.

• $B_{DC} \in \mathbb{R}^{N_h}$ contains the contribution of the DC independent sources while $B_{RF} \in \mathbb{R}^{N_h}$ and $B_{LO} \in \mathbb{R}^{N_h}$ show the location of the RF and LO input frequency tones, respectively.

• $\alpha$ and $\beta$ are the amplitudes of the RF and the LO voltages, respectively.

• $A \in \mathbb{R}^{N_h \times N_h}$ is a block matrix representing the contribution of the linear elements.

• $F(X) \in \mathbb{R}^{N_h}$ is the vector of nonlinear equations.

2.2. Harmonic Balance Moments

The Harmonic Balance moments are the coefficients of the Taylor series expansion of the Harmonic Balance vector of unknowns, $X$, with respect to the signal amplitude voltage $\alpha$, as given by

$$X = M_0 + M_1 \alpha + M_2 \alpha^2 + M_3 \alpha^3 + \ldots = \sum_{i=0}^{\infty} M_i \alpha^i$$  (2)

where $M_k$ is the $k^{th}$ moment vector. The zeroth moment $M_0$, is obtained by finding the solution of the system described by (1) with $\alpha = 0$. The remaining moment vectors $M_n$ can then be found by solving the system of equations given by [5]

$$\Phi M_1 = B_{RF}$$  (3)

$$\Phi M_n = -\frac{1}{n} \sum_{j=1}^{n-1} (n-j) T_j M_{n-j}, \quad n \geq 2$$  (4)

where $\Phi$ is the moments computation matrix given by

$$\Phi = A + \left. \frac{\partial F(X)}{\partial X} \right|_{\alpha=0},$$  (5)

and $T_j$ are the coefficients of the Taylor series expansion with respect to $\alpha$ of the nonlinear Jacobian given by

$$\frac{\partial F(X)}{\partial X} = \sum_{i=0}^{\infty} T_i \alpha^i.$$  (6)

It is important to note that the matrix $\Phi$ has the same structure as a Jacobian matrix but is evaluated with only the DC and LO tones present which makes it very sparse. As can be seen from (3) and (4), the computation of the moment vectors is a solution of a set of sparse linear algebraic equations where the left-hand-side matrix is the same throughout and is therefore very efficient.

![Figure 1: Location of distortion terms in moments for amplifier circuits](image)

![Figure 2: Location of distortion terms in moments for mixer circuits](image)

2.3. Computation of IP3 from the Moments

The value of the input referred IP3 (IIP3) for a circuit excited with a single frequency tone RF signal of $V_{in}(t) = \alpha \cos(\omega t)$, can be determined from the moments using the following relation [3],

$$IIP3 = \sqrt{\frac{m_{1,1}}{m_{1,3}}}$$  (7)

In this equation, $m_{1,1}$ represents the entry in the first moment vector at the fundamental frequency of $\omega$, while the term $m_{1,3}$ represents the entry in the third moment vector at the same fundamental frequency as illustrated in Fig. 1 for amplifiers and in Fig. 2 for mixers.

3. Moments Based Sensitivity

The relative sensitivity of IIP3 with respect to a general parameter $'y = \lambda_0 + \lambda$', is defined as follows

$$S_{IIP3}^{IIP3} = \lambda_0 \frac{\partial (IIP3)}{\partial y} = \lambda_0 \frac{\partial (IIP3)}{\partial \lambda}$$  (8)

where $\lambda_0$ is the nominal value of the parameter and $\lambda$ is the change in its value. Essentially, To find the sensitivity of
IP3 with respect to the parameter $\lambda$, from (7) we note that we need to obtain
\[
\frac{\partial}{\partial \lambda} (IIP_3) = \frac{1}{2} m_{1,1}^{-1} m_{1,3} \frac{\partial m_{1,1}}{\partial \lambda} - m_{1,3} \frac{\partial m_{1,3}}{\partial \lambda},
\]
From (9), we observe that determining the sensitivity of IP3 now essentially comes down to determining the value of $\frac{\partial m_{1,1}}{\partial \lambda}$ and $\frac{\partial m_{1,3}}{\partial \lambda}$, since the terms $m_{1,1}$ and $m_{1,3}$ are already available from the original computation of IP3.

### 3.1. Adjoint Moments Sensitivity Analysis

The computation of the sensitivity of IP3 has been reduced to finding the derivatives of $m_{1,1}$ and $m_{1,3}$ with respect to $\lambda$. In this section, we derive an efficient adjoint-based approach for computing these derivatives. As illustrated in Fig. 1 and Fig. 2, the terms $m_{1,1}$ and $m_{1,3}$ can be written as
\[
m_{1,1} = d^T M_1 \tag{10}
m_{1,3} = d^T M_3 \tag{11}
\]
where $d$ is a selection vector. Note that $m_{1,1}$ and $m_{1,3}$ appear in the Taylor expansion of $X_{out}$ defined as
\[
X_{out} = d^T X = m_{1,0} + m_{1,1} \alpha + m_{1,2} \alpha^2 + m_{1,3} \alpha^3 + \ldots \tag{12}
\]
The derivative of $X_{out}$ with respect to $\lambda$ can now be written as
\[
\frac{\partial X_{out}}{\partial \lambda} = \frac{\partial m_{1,0}}{\partial \lambda} + \frac{\partial m_{1,1}}{\partial \lambda} \alpha + \frac{\partial m_{1,2}}{\partial \lambda} \alpha^2 + \frac{\partial m_{1,3}}{\partial \lambda} \alpha^3 + \ldots \tag{13}
\]
From this equation, we can deduce that the terms $\frac{\partial m_{1,1}}{\partial \lambda}$ and $\frac{\partial m_{1,3}}{\partial \lambda}$, required in (9), are the first and third moments of the expansion of $\frac{\partial X_{out}}{\partial \lambda}$. In order to compute these moments, we start by using the general Harmonic Balance adjoint sensitivity expression to write [9]
\[
\frac{\partial X_{out}}{\partial \lambda} = X_{\alpha}^T \frac{\partial A}{\partial \lambda} X \tag{14}
\]
where $X_{\alpha}$ is the solution of the Adjoint equations
\[
J^T X_{\alpha} = -d \tag{15}
\]
From (14), it can be seen that the moments of $\frac{\partial X_{out}}{\partial \lambda}$ can be expressed in terms of the moments of $X$ and $X_{\alpha}$.

The adjoint moment vectors are defined as the Taylor series coefficients of the expansion of the adjoint solution vector $X_{\alpha}$, defined in (15), with respect to the signal amplitude voltage $\alpha$. The expansion of $X_{\alpha}$ can therefore be expressed as
\[
X_{\alpha} = M_{0\alpha} + M_{1\alpha} \alpha + M_{2\alpha} \alpha^2 + M_{3\alpha} \alpha^3 + \ldots = \sum_{i=0}^{\infty} M_{i\alpha} \alpha^i \tag{16}
\]
where $M_{ik}$ is the $k^{th}$ adjoint moment. By substituting (2), (13) and (16) in (14), we get the final expressions in terms of the moments. By equating the powers of $\alpha$ and $\alpha^3$ on both sides of the resulting expressions we can write
\[
\frac{\partial m_{1,1}}{\partial \lambda} = M_{0\lambda}^T \left( \frac{\partial A}{\partial \lambda} \right) M_1 + M_{1\lambda}^T \left( \frac{\partial A}{\partial \lambda} \right) M_0 \tag{17}
\]
\[
\frac{\partial m_{1,3}}{\partial \lambda} = M_{0\lambda}^T \left( \frac{\partial A}{\partial \lambda} \right) M_3 + M_{2\lambda}^T \left( \frac{\partial A}{\partial \lambda} \right) M_2 + M_{3\lambda}^T \left( \frac{\partial A}{\partial \lambda} \right) M_0 \tag{18}
\]
It is important to note that the matrix $\frac{\partial A}{\partial \lambda}$ contains only the harmonic balance ‘stamp’ of the derivative of the element that $\lambda$ is a parameter of. It is, therefore, an extremely sparse matrix with at most four non-zero block entries.

### 3.2. Computation of the Adjoint Moments

The computation of the adjoint moment vectors (defined in (16)) is a very CPU efficient algorithm. This is achieved by first substituting (6) and (16) into (15), which gives the following general relation
\[
\left( A + \sum_{i=0}^{\infty} T_i \alpha^i \right)^T \left( \sum_{i=0}^{\infty} M_{i\alpha} \alpha^i \right) = -d \tag{19}
\]
To determine the expressions for computing each individual adjoint moment vector, we equate powers of $\alpha$ on both sides of (19). This results in the following set of equations that can be solved sequentially:
\[
\Phi_i^T M_{0\alpha} = -d \tag{20}
\]
\[
\Phi_i^T M_{1\alpha} = -T_i^T M_{0\alpha} \tag{21}
\]
\[
\Phi_i^T M_{2\alpha} = -T_i^T M_{1\alpha} - T_i^T M_{0\alpha} \tag{22}
\]
\[
\Phi_i^T M_{3\alpha} = -T_i^T M_{2\alpha} - T_i^T M_{1\alpha} - T_i^T M_{0\alpha} \tag{23}
\]
Note that $\Phi$ is the same sparse moments computation matrix that was used to determine the Harmonic Balance moments in (3) and (4). This means that no additional LU decompositions are required to find the adjoint moments.

### 4. Numerical Examples

In this section, we compute the value of IP3 and its sensitivity for two example circuits using the single-tone moments method and then compare the results to those obtained using Harmonic Balance to demonstrate the accuracy and speed-up of the new approach. The two circuits considered are a differential Low Noise Amplifier (LNA) with an IP3 of $-7.24$dBm, in addition to a singly-balanced mixer circuit with an IP3 of $-3.4$dBm. Both circuits are implemented using Bipolar Junction Transistors and the sensitivity computed is with respect to a collector resistor $R_C$. The computation cost of the sensitivity with respect to additional parameters is negligible in both methods.

First, we use the Harmonic Balance method to determine the steady-state solution of both circuits. The Differential
Table 1: CPU Cost Comparison of Finding both IP3 and its Adjoint Sensitivity for the Differential Amplifier Circuit.

<table>
<thead>
<tr>
<th>Type of Computation</th>
<th>Harmonic Balance (s)</th>
<th>Proposed Method (s)</th>
<th>Speed-Up</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP3</td>
<td>44.67</td>
<td>0.34</td>
<td>129 x</td>
</tr>
<tr>
<td>Sensitivity</td>
<td>6.03</td>
<td>0.03</td>
<td>201 x</td>
</tr>
<tr>
<td>Total</td>
<td>50.7</td>
<td>0.37</td>
<td>137 x</td>
</tr>
</tbody>
</table>

Table 2: CPU Cost Comparison of Finding both IP3 and its Adjoint Sensitivity for the Mixer Circuit.

<table>
<thead>
<tr>
<th>Type of Computation</th>
<th>Harmonic Balance (s)</th>
<th>Proposed Method (s)</th>
<th>Speed-Up</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP3</td>
<td>118.43</td>
<td>0.43</td>
<td>275 x</td>
</tr>
<tr>
<td>Sensitivity</td>
<td>21.63</td>
<td>0.21</td>
<td>103 x</td>
</tr>
<tr>
<td>Total</td>
<td>140.06</td>
<td>0.64</td>
<td>218 x</td>
</tr>
</tbody>
</table>

LNA circuit was run with two input tones at $f_1 = 1000$ MHz and $f_2 = 1001$ MHz. The sensitivity of IP3 with respect to a change in $R_C$ was found to be $-2.354 \times 10^{-4}$ V. For the singly-balanced mixer, the two input RF tones were at $f_1 = 1000$ MHz and $f_2 = 100.1$ MHz, while the LO frequency was set to $f_{LO} = 1$ GHz. The sensitivity of IP3 with respect to $R_C$ was found to be $3.104 \times 10^{-2}$ V for this particular circuit.

Next, using the single-tone moments method, we first compute the adjoint moments using the relations given in (20)–(23) and only a single tone at $f = 1000$ MHz for the LNA, and at $f = 100$ MHz for the mixer. The sensitivity expressions are then determined using (17) and (18) with the matrix $\frac{\partial A}{\partial a}$ being the Harmonic Balance stamp of the resistor. The sensitivity of IP3 is then computed by evaluating (9). The IP3 sensitivity obtained showed a difference of 0.08% for the LNA, and a difference of 0.63% for the mixer when compared to perturbation. As can be seen, the results are very accurate.

Tables 1 and 2 show a comparison of the CPU times between traditional Harmonic Balance and the proposed method for determining IP3 and its adjoint sensitivity obtained using a prototype MATLAB simulator on a local workstation. As can be seen, the proposed method presents significant computational speedup in the CPU time needed to find both IP3 and its sensitivity.

5. Conclusion

In this paper, a method for the efficient sensitivity analysis of third order nonlinear distortion based on single-tone adjoint moments analysis was presented. This approach adds insight to the results of the single-tone moments based method for computing IP3 presented in [3], while still remaining significantly more efficient than traditional multi-tone simulation approaches based on Harmonic Balance.

References

SPICE-Oriented Algorithm for Assessment of Stability for Periodic Solutions

Hiroshige Kataoka, Yoshihiro Yamagami, Yoshifumi Nishio and Akio Ushida

Dept. Electrical and Electronic Eng, Tokushima University
Tokushima 770-8506, JAPAN
Email: hiroshige@ee.tokushima-u.ac.jp, yamagami@ee.tokushima-u.ac.jp, nishio@ee.tokushima-u.ac.jp

Abstract—The assessment of the stability for periodic solutions is very important for designing the circuit. There are many method for the assessment of the stability. In this article, we propose a SPICE-oriented method for the assessment of the stability, that is based on the Floquet theory. By using our method, we can assess the stability of the circuit easily. First, we obtain the periodic solutions of the circuit by using the SPICE. Next, we calculate the eigenvalues of a Jacobian matrix by solving variational circuits based on the Floquet theory. As an example, we assess the stability of the periodic solutions for one order resonance circuit including nonlinear capacitors.

1. Introduction

When we simulate the circuit, we often refer to the assessment of the stability. In this paper, we propose a SPICE-oriented algorithm to the assessment of the stability for periodic solutions which is based on the Floquet theory [1]. In the conventional method, we have to calculate the Jacobian matrix for the periodic solutions by solving the variational equations. In this study, we obtain the Jacobian matrix by the transient analysis of SPICE for variational circuits which is easily derived from the original circuit.

Section 2.1 shows how to use the sine-cosine circuits [2], which is based on the HB (harmonic balance) method. We use the sine-cosine circuit to obtain the value of the voltages which are required in order to solve variational circuits. Section 2.2 shows the solution curve-tracing circuit. It is based on the arc-length method [3][4]. Section 2.3 shows the Floquet theory. Section 3 shows an illustrative example and how to solve the variational circuits by using SPICE. Section 4 shows the results and confirms the effectiveness of the proposed method. Section 5 concludes this article.

2. Frequency analysis and assessment of stability

2.1. Sine-cosine circuit

We introduce the sine-cosine circuit corresponding to the determining equation of the HB method. If we set the voltage through a capacitor $C$

\[ v_C = V_{CS} \sin \omega t + V_{CC} \cos \omega t, \]  

(1)

the current $i_C$ is given by

\[ i_C = C \frac{dv_C}{dt} = -\omega V_{CC} \sin \omega t + \omega V_{CS} \cos \omega t. \]  

(2)

Then, the voltage $v_L$ is given by

\[ v_L = L \frac{di_L}{dt} = -\omega I_{LC} \sin \omega t + \omega I_{LS} \cos \omega t. \]  

(5)

Thus, the coefficients of $\sin \omega t$ and $\cos \omega t$ are described by

\[ I_{LS} = -\omega V_{LS}, \quad V_{LS} = -\omega V_{LC}. \]  

(6)

Namely, the inductor is replaced by coupled voltage-controlled current sources in the sine-cosine transformation of the HB method. In the same way, let the current through an inductor $L$ be

\[ i_L = I_{LS} \sin \omega t + I_{LC} \cos \omega t. \]  

(4)

2.2. Solution curve-tracing circuit

Even we use our sine-cosine circuits, we can not obtain unstable periodic solutions, because we set the frequency as time in SPICE. In this section, we explain the STC (solution curve trace circuit) realizing the arc-length method.

First, we can express the arc length in $(n+1)$ dimensional euclidean space as Eq. (7)

\[ ds = \sqrt{(dx_1)^2 + (dx_2)^2 + (dx_3)^2 + \ldots + (dx_{n+1})^2} \]  

(7)

\[ \frac{dx_1}{\sqrt{1 + (dx_1)^2}} = \frac{dx_2}{\sqrt{1 + (dx_2)^2}} = \ldots = \frac{dx_{n+1}}{\sqrt{1 + (dx_{n+1})^2}}. \]  

\[ \frac{dx_1}{\sqrt{1 + (dx_1)^2}} = \frac{dx_2}{\sqrt{1 + (dx_2)^2}} = \ldots = \frac{dx_{n+1}}{\sqrt{1 + (dx_{n+1})^2}}. \]  

\[ ds = \sqrt{(dx_1)^2 + (dx_2)^2 + (dx_3)^2 + \ldots + (dx_{n+1})^2} \]  

(7)
In order to trace the solutions curve by SPICE, we replace the differentiation with respect to the arc-length \( s \) by the time \( t \). We assume \( x_k \) as voltages in SPICE. From this, we can obtain Eq. (8).

\[
\sum_{i=1}^{p} \frac{(dv_i)}{dt}^2 + \frac{(dv_o)^2}{dt} = 1 \tag{8}
\]

In this paper, where \( v_i \) \((i = 1, 2, \ldots, p)\) are the coefficient of voltages in Eqs. (1) and (4) and \( v_o \) corresponds to \( \omega \). They are realized by using differentiators (simply realized by capacitors with \( 1(F) \) in SPICE). The circuit in Fig. 3 realizes to satisfy the arc-length method Eq. (8). In this circuit, the voltages corresponding to the coefficients are inputted after differentiated and squared via the voltage-controlled current source (VCCS). If we set the voltage of node \( a \) as \( \dot{v}_o \), \( I_o = \dot{v}_o \) can be obtained by multiplier and voltage-controlled current source VCCS (MVCCS). Then, the node voltage \( v_o \) is integrated to obtain \( \dot{v}_o \). Note that \( R \) in Fig. 3 is a very large resistance used only to avoid the \( L - J \) cut-set.

### 2.3. Stability of periodic solutions

We suppose that there is a circuit equation as

\[
f(x, x, y, \omega t) = 0 \tag{9}
\]

and make the variational equation for the regular period solution of \( \dot{x} \). First, we assume the small change quantity as \((\Delta x, \Delta y)\) as

\[
\begin{align*}
x &= \dot{x} + \Delta x \\
y &= \dot{y} + \Delta y
\end{align*} \tag{10}
\]

and substitute Eq. (10) to Eq. (9). We obtain the equation as

\[
f(\ddot{x}, \dot{x}, \dot{y}, \omega t) + \frac{\partial f}{\partial x} \frac{\partial f}{\partial y} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = 0 \tag{11}
\]

In Eq. (11), the first term is regular period solution and second term is variational equation. We change the second term as

\[
\Delta x = A(t)\Delta x. \tag{12}
\]

In Eq. (12), \( A(t) \) is the periodic function. We apply the Jacobian matrix of the periodic solution as \( \Phi(t) \). From this, the solution after one period from initial value of \( \Delta x(0) \) is given as follows;

\[
\Delta x(T) = \Phi(T)\Delta x(0). \tag{13}
\]

Hence, when the eigenvalues \((\lambda_1, \lambda_2, \ldots, \lambda_n)\) of \( \Phi(T) \) satisfy \(|\lambda_k| < 1 \ (k = 1, 2, \ldots, n)\), the regular periodic solution \( \ddot{x} \) is stable.

### 3. Illustrative example

Figure 4 shows a circuit for an illustrative example. The nonlinear characteristics can be solved by using the SPICE model in Fig. 5.

We can express the circuit equations as follows;

\[
\begin{align*}
e(t) &= Ri + \frac{L}{dt} + \alpha q + \beta q^3 \\
\frac{dq}{dt} &= i
\end{align*} \tag{14}
\]

If we write the variables as periodic solutions with small variations:

\[
\begin{align*}
i &= i_0 + \Delta i \\
q &= q_0 + \Delta q
\end{align*} \tag{15}
\]

we obtain the following variational equations.

\[
\begin{align*}
e(t) &= Ri + \frac{L}{dt} + (\alpha + \beta q_0^3)\Delta q \\
\frac{d\Delta q}{dt} &= \Delta i
\end{align*} \tag{16}
\]
where we neglect higher-order small terms.

From these equations, we can make the variational circuit for Fig. 5. In Fig. 6, \( q_0 \) is the steady solutions given as
\[
q_0 = Q_c \cos \omega t + Q_s \sin \omega t.
\]

We analyze this circuit and calculate the values of two variables after one period from two different initial conditions; \((\Delta i_0, \Delta q_0) = (1, 0)\) or \((\Delta i_0, \Delta q_0) = (0, 1)\). We obtain 4 variational values for \( \Phi(T) \).

\begin{equation}
\Phi = \begin{bmatrix}
\Delta i(\Delta i_0, \Delta q_0) = (1, 0) \\
\Delta q(\Delta i_0, \Delta q_0) = (1, 0)
\end{bmatrix}
\end{equation}

We calculate the eigenvalues of \( \Phi(T) \) by using MATLAB and assess the stability of the periodic solutions.

4. Simulation results

Figures 7 and 8 show the simulation results of the frequency response of \( i \) and \( q \), respectively, which are obtained by the sine-cosine circuits and the STC with SPICE.

We set the parameters as follows; \( E_m = 0.35[V] \), \( \alpha = 1.0 \), \( \beta = 0.8 \), \( R = 0.05[\Omega] \), \( L = 0.1[H] \). In this section, we compare and check the our results with the results which obtained by transient analysis of Fig. 4. We analyze the case \( \omega = 4.78[rad/s] \), \( \omega = 4.51[rad/s] \) and \( \omega = 4.26[rad/s] \).

For the validation, we simulate the original circuit. The simulation results of transient analysis (Fig. 9, Fig. 10 and Fig. 11), indicate that \( \omega = 4.78[rad/s] \) is stable, and that \( \omega = 4.51[rad/s] \) and \( \omega = 4.26[rad/s] \) are unstable.

Next, we show the results which obtained by our method. First, we show the solutions for the case of \( \omega = 4.78[rad/s] \).

\[
\Phi = \begin{bmatrix}
0.599 & 0.012 \\
-13.897 & 0.593
\end{bmatrix}
\]

Second, we show the solutions for the case of \( \omega = 4.51[rad/s] \).

\[
\Phi = \begin{bmatrix}
0.0447 & 0.149 \\
-2.831 & 1.707
\end{bmatrix}
\]

Lastly, we show the solutions for the case of \( \omega = 4.26[rad/s] \).

\[
\Phi = \begin{bmatrix}
0.4001 & 0.1209 \\
0.7348 & 1.412
\end{bmatrix}
\]

Table 1 shows the calculated eigenvalues of \( \Phi \) for the two \( \omega \).
We can see that $\omega = 4.78\,[\text{rad/s}]$ is stable, because all of eigenvalues satisfy $|\lambda| < 1$. However, for the other patterns, the solutions are unstable, because one of the two eigenvalues does not satisfy $|\lambda| < 1$.

Namely, we can say that our method gives the same results as the results obtained by transient analysis with simpler SPICE-oriented algorithm.

5. Conclusion

We proposed a SPICE-oriented algorithm to assess the stability of periodic solutions for nonlinear circuits. We obtained periodic solutions by using SPICE and we assessed the stability based on the Floquet theory. In detail, we analyzed the second order resonance circuit with nonlinear capacitors for three different conditions of $\omega$ which gives both stable and unstable solutions. Our results agree well with the previously obtained results. We would like to improve the proposed method more effectively for the analysis of larger scale circuit.

Acknowledgment

This work was partly supported by CASIO Science Promotion Foundation.

References


Table 1: Eigenvalues of $\Phi$

| $\omega\,[\text{rad/s}]$ | $|\lambda_1|$ | $|\lambda_2|$ |
|-------------------------|---------------|---------------|
| $\omega = 4.78$         | 0.5963+0.4093i | 0.5963-0.4093i |
| $\omega = 4.51$         | 0.358         | 1.3936         |
| $\omega = 4.26$         | 0.319         | 1.4928         |
A Combinatorial Optimization Method which Combines Ant Colony Optimization and Chaotic Dynamical

Hideki Igeta† and Mikio Hasegawa*

†Dept. of Electrical Engineering, Tokyo University of Science
1-14-6 Kudankita, Chiyoda-ku, Tokyo, 102-0073 Japan
Email: igeta@haselab.ee.kagu.tus.ac.jp

Abstract—There are various combinatorial optimization problems in various fields, such as science, engineering, economy and so on. For heuristic solution search in combinatorial optimization problems, the chaotic dynamics has been shown effective by many studies. Such chaotic search algorithms are based on an improvement of the solutions by some local search heuristics. In order to improve the performance of algorithms, the tabu search and gradient search have been combined with global search algorithms, such as the genetic algorithms, ant colony optimization and so on. In such hybrid approach, the ant colony optimization has the better performance than the genetic hybrid. In this paper, we propose a novel hybrid method that combines the chaotic neural tabu search, which is based on the neural search and the ant colony optimization. Our results show that the proposed hybrid algorithm has better performance than the original chaotic search and its hybrid with the genetic algorithm.

1. Introduction

There are various combinatorial optimization problems in various fields. In those problems, exact solution of the NP-hard problems cannot be solved in reasonable time. For such problems, it is important to develop effective heuristic methods for solving better solutions in reasonable time.

As one of the heuristic approaches, effectiveness of the chaotic dynamics has been shown by various studies [5,6,7,8]. There are three major methods in chaotic optimization approach. The first method is to add chaotic noise in the mutually connected neural network minimizing energy function [5]. It has been shown more effective than the stochastic noise, such as the white Gaussian noise [6]. The second method is to apply high dimensional chaotic neural dynamics to the mutually connected neural networks [7]. However, these two methods based on the mutually connected neural networks are applicable only to very small toy problems. To enable applications of such chaotic dynamics to much larger problems, the third method combines effective searching ability of the optimal solution by the chaotic dynamics with the heuristic methods that can be easily applied to large scale combinatorial optimization problems. This third chaotic method has been shown more effective than existing tabu searches, or stochastic searches, even for the large scale problems [8,9].

Such chaotic search algorithms searching solution search are based on chaotic updating of local effective. To improve the performance of local search based methods, such algorithms have been combined with global searches, such as the genetic algorithms (GA), ant colony optimization (ACO), and so on. The chaotic dynamical method has also been combined with a global search algorithm, the GA [13]. The global searching methods much improves the performance of the tabu search and the local search, and solved the best known solutions in many benchmark problems in the QAPLIB [18]. In Ref. [14,15], the ACO, the GH and other algorithms have been compared and it has been shown that the ACO has the best solving performance.

Therefore, in this paper, we propose a novel hybrid method that combines chaotic neural tabu search proposed in Ref. [9], which has the better performance than the tabu search or the stochastic search, and the ACO proposed in Ref. [15], which may be the best for the hybrid algorithms. The proposed method is applied to the QAPs and its performance is compared with the original chaotic neural tabu search, hybrid method that combines the GA and the chaotic neural tabu search.

2. Local search and global search algorithms

2.1. Local search algorithms

The local search algorithms are the methods to search neighboring solutions in a searching space. In this paper, such neighboring solution search is based on exchanges of the elements. The methods based on such neighboring solutions are effective for searching in detail in narrow space. However, it is difficult to jump to the states far from the current solution. Therefore, in this paper such local search algorithms are utilized for searching deeply in a limited area.

There are a lot of existing methods based on neighboring solution search, such as steepest descent method, tabu search [1], chaotic neural tabu search [8] and so on. Among such methods, the chaotic dynamics has been shown effectiveness in such approach [5-12]. As one of the more effective chaotic methods, chaotic neural tabu
search has been shown more effective than existing tabu searches and stochastic searches [9].

In this paper, the chaotic neural tabu search is realized by extending the Taillard’s tabu search [16] using the chaotic neural network [9]. It can be realized by the following chaotic neural network updating a local search heuristics,

$$\xi_j(t + 1) = \beta \Delta(t),$$  \hspace{1cm} (1)

$$\eta_j(t + 1) = -W \sum_{k = 11, \{k \neq i \}}^{n} x_{ki}(x) + W,$$  \hspace{1cm} (2)

$$\gamma_j(t + 1) = k_r \xi_{p(j)q(i)}(t) - \alpha \{x_{p(j)q(i)}(t) + z_{p(j)q(i)}(t)\} + R,$$  \hspace{1cm} (3)

$$\zeta_j(t + 1) = k_r \xi_j(t) - \alpha \{x_j(t) + z_j(t)\} + R,$$  \hspace{1cm} (4)

$$x_j(t + 1) = f\{\xi_j(t + 1) + \eta_j(t + 1) + \gamma_j(t + 1) + \zeta_j(t + 1)\},$$  \hspace{1cm} (5)

where, the output function is the sigmoidal function

$$f(y) = \frac{1}{1 + \exp(-\frac{y}{\xi})},$$

the refractory effects and gain inputs, $k_r$ is decay parameter of the tabu effects, $W$ is connection weights, $n$ is the size of the problem and $R$ is the positive bias, respectively.

When the output of the $(i, j)$ th neuron, $x_j(t + 1)$, becomes larger than 0.5, the $(i, j)$ neuron fires, and the element $i$ is assigned to the jth index as shown in Fig.1. At the same time the element in the $j$th index, $p(j)$, is assigned to the $q(i)$th index. Therefore, when the $(i, j)$ th neuron fires, not only the assignment of $(i, j)$ but also the $(p(j), q(i))$ must be tabu. $z_j(t)$ and $z_{p(j)q(i)}(t)$ are prepared to memories such assignments as $(p(j), q(i))$, which is updated by the following $z_j(t + 1)$ is reset to 0, when the $(i, j)$ th neuron is updated; $x_{p(j)q(i)}(t + 1)$ is added to $z_j(t + 1)$, when updating other neurons.

$$1, \ldots, j, \ldots, q(j), \ldots, n$$

$$P(1, \ldots, p(j), \ldots, i, \ldots, n)$$

$$1, \ldots, j, \ldots, q(i), \ldots, n$$

$$P(1, \ldots, i, \ldots, p(j), \ldots, p(n))$$

Figure 1: Update of the solution by firing of the $(i, j)$ th neuron.

The chaotic neural tabu search described above is a searching method that has not only features of the tabu search but also features of the chaotic dynamics that is known effective for combinatorial optimization. That is the reason why the chaotic neural tabu search is superior to the tabu search [9].

However, since the local search algorithms are based on the neighborhood search, they are not good at searching globally in a search space. To overcome such an issue, the local search algorithms have been combined with global search heuristic methods, such as the GA, ACO, and so on. Such Hybrid methods have been shown much better performance [14,17]. The chaotic method has also combined with the GA and improvement of the performance has been shown [13].

2.2. Hybrid algorithms

In the hybrid approaches, it has been shown that, the hybrid method combines the ant system and a local search has better performance than the GA and a local search [14,15]. Therefore in this paper, we propose the chaotic neural tabu search, which has good searching ability in local searches, combined with the ACO. The origin of the ant systems is to simulate the behavior of ants’ searching food. The ants find the sources of food in the following way: First, they explore the area surrounding their nest in a random manner. While they are moving, the ants left a pheromone (chemical trace) on the floor, in such a way that they can find their way back to the nest. When they find a source of food, the ants bring food back to the nest following the pheromone traces, leaving additional pheromone during the return trip. After a while, the paths between the nest and sources of food will be indicated by an amount of pheromone in relation with the length of the path. Indeed, short paths will be travelled at a higher rate than long ones and the amount of pheromone will grow faster on the short ways. Therefore, the ants are able to optimize their paths by this process. In this paper, we apply the ACO effective for global search to the QAPs.
3. Hybrid Method Combining Chaotic Search and Ant Colony Optimization

Our proposed algorithm is a combination of the chaotic neural tabu search as a local search proposed in Ref.[9] and the ACO as a global search proposed in Ref.[15]. The procedure of the proposed algorithm is shown in Fig.2. We prepare the pheromone matrix $T$, whose size is $n \times n$ to memorize previous better solutions. The $(i, j)$ th element of the matrix $T$, $\tau_{ij}$, is corresponding to the probability of the assignment of the element $i$ to the index $j$. $p(j) = i$. The pheromone matrix $T$ is used for generating the initial solutions for local search.

First, in the step 1 in Fig.2, all of the elements $\tau_{ij}$ in the pheromone matrix $T$ are set 1. In the step 2, the ACO generate initial solution for the chaotic neural tabu search according to the pheromone matrix $T$, by the following procedure.

1) $I = \phi, J = \phi$
2) While $|I| < n$ repeat:
   2a) Choose $j$, randomly, uniformly, $1 \leq j \leq n$, $j \notin J$.
   2b) Choose $i$, randomly, $1 \leq i \leq n$, $i \notin I$, with probability $\frac{\tau_{ij}}{\sum_{k=1}^{n} \tau_{ik}}$ and set $p(j) = i$.
   2c) $I = I \cup \{i\}, J = J \cup \{j\}$

Then, in the step 3, the chaotic neural tabu search described in the previous section is applied to the generated initial solution in step 2. During the solution search by the chaotic algorithm, the best solution found in the search in this step is memorized as $P^*$. If the $P^*$ is better than the best solution found so far in whole procedure, it is memorized as $P^**$. After the chaotic neural tabu search, in step 4, the pheromone matrix is updated, by the following procedure.

1) For $i = 1$ to $n$ do:
   1a) $\tau_{ip^*(i)} = \tau_{ip^*(i)} + r^*$
   1b) $\tau_{ip^{**}(i)} = \tau_{ip^{**}(i)} + r^{**}$,

Where, $r^*$ and $r^{**}$ are the parameters corresponding to reinforcement of the probability of the assignments in the solution obtained by the chaotic neural tabu search, and those in the best solution found so far, respectively. By repeating from the steps 2 to 4, better solutions become easy to be found by applying chaotic searches to more appropriate initial solutions generated by the pheromone matrix $T$.

4. Numerical Experiments

We apply the proposed hybrid algorithm to asymmetric QAPs, whose sizes are 20 to 100 from QAPLIB [18]. The performance is compared between the proposed method that combines the chaotic neural tabu search and the ACO with the original chaotic neural tabu search and the genetic hybrid combining the GA, and the chaotic neural tabu search.

Table 1: Results of the proposed hybrid algorithm combining chaotic neural tabu search and ACO, the original chaotic neural tabu search and the genetic hybrid combining the GA, and the chaotic neural tabu search combined with the GA, on QAPs. The results are shown by percentages of average gaps from the best known solutions.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Combining Chaotic Neural Tabu Search and Ant Colony Optimization</th>
<th>Chaotic Neural Tabu Search</th>
<th>Combining the GA and the Chaotic Neural Tabu Search</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tai20b</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Tai25b</td>
<td>0</td>
<td>0</td>
<td>0.0651</td>
</tr>
<tr>
<td>Tai30b</td>
<td>0.0588</td>
<td>0.147</td>
<td>0.0181</td>
</tr>
<tr>
<td>Tai35b</td>
<td>0.0102</td>
<td>0.430</td>
<td>0.349</td>
</tr>
<tr>
<td>Tai40b</td>
<td>0.0217</td>
<td>0.210</td>
<td>0.0153</td>
</tr>
<tr>
<td>Tai50b</td>
<td>0.0452</td>
<td>0.216</td>
<td>0.155</td>
</tr>
<tr>
<td>Tai60b</td>
<td>0.00887</td>
<td>0.172</td>
<td>0.123</td>
</tr>
<tr>
<td>Tai80b</td>
<td>0.0310</td>
<td>0.146</td>
<td>0.558</td>
</tr>
<tr>
<td>Tai100b</td>
<td>0.107</td>
<td>0.653</td>
<td>0.198</td>
</tr>
<tr>
<td>Average</td>
<td>0.0314</td>
<td>0.216</td>
<td>0.165</td>
</tr>
</tbody>
</table>

Table 1 shows the results of those three methods on Tai20b whose size is 20 to Tai100b whose size is 100. From the Table 1, that the proposed hybrid algorithm combining chaotic neural tabu search and ACO has better results than the original chaotic neural tabu search. This result confirms that the proposed hybrid algorithm improves the performance of the original chaotic search.
method. By comparing the proposed hybrid algorithm combining chaotic neural tabu search and ACO with the hybrid combining the chaotic neural tabu search and the GA, the proposed algorithm using ACO has better performance especially in the large scale problems. For hybrid approach using the chaotic neural tabu search, our results show that the ACO is the best and much improves the performance.

5. Conclusions

We propose a novel hybrid method that combines the chaotic neural tabu search and the ACO. By combining the chaotic search which improves the solution by efficiently moving it, with an efficient global search, the ACO, the performance can be much improved. Our simulation results show that the proposed hybrid algorithm has the better performance than the original chaotic search and its hybrid with GA. Our algorithm is effective especially for the large scale problems.

As future works, we would like to improve hybrid method by introducing other ACO hybrid methods. We also would like to apply this proposed approach to the large scale TSPs.

References


Slide-and-Insert Assignment Method with Chaotic Dynamics for Quadratic Assignment Problems

Yusuke Sakamoto and Yoshihiko Horio

†Graduate School of Engineering, Tokyo Denki University
2–2 Kanda-Nishiki-cho, Chiyoda-ku, Tokyo, 101–8457 Japan
Email: 10kme34@ms.dendai.ac.jp, horio@eee.dendai.ac.jp

Abstract—A quadratic assignment problem (QAP) is an NP-hard combinatorial optimization problem. Some heuristic algorithms for the QAPs is proposed in order to find the sub-optimum solutions efficiently. In this paper, we propose a novel heuristic algorithm for the QAPs based on the Or-opt algorithm used for traveling salesman problems. The proposed method introduces a slide-and-insert operation for the assignments of the elements in the QAP. Furthermore, we improve the proposed algorithm by combining it with the 2-opt algorithm. We also use chaotic dynamics instead of the random numbers. Numerical simulation results, which compare the solving performance of the proposed algorithms with the random numbers and the chaotic dynamics, are shown.

1. Introduction

A quadratic assignment problem (QAP)[1] is one of the combinatorial optimization problems, and belongs to the class of NP-hard. For large-size QAPs, it would take an impractical time to obtain the optimum solution. Therefore, heuristic algorithms have been proposed in order to find the sub-optimum solutions in reasonable time.

In this paper, we propose a slide-and-insert assignment method, which is based on the Or-opt algorithm for traveling salesman problems (TSP)[2], which is another NP-hard optimization problem. We also introduce chaotic dynamics into the proposed method. We compare the performance of the proposed method with the chaotic dynamics and that with the random numbers through numerical simulations.

1.1. Traveling Salesman Problem (TSP)

The TSP seeks the shortest route that visits each city only once, and returns to the starting point. The length of a tour can be expressed as eq. (1).

\[ \text{length} = \sum_{i=1}^{n-1} C(p(i), p(i+1)) + C(p(n), p(1)) \]

where \( p(i) \) (\( 1 \leq i \leq n \)) is the element of the permutation \( p \), which gives a feasible solution. \( C(a, b) \) is the distance between cities \( a \) and \( b \), and \( n \) is the number of cities. There are \( (n-1)!/2 \) possible routes for the size-\( n \) TSP, so that it is impossible to obtain the exact solution for reasonable time if the number of cities increases. Therefore, many heuristic algorithms were proposed to obtain the sub-optimum solutions in reasonable time. The Or-opt algorithm is one of these heuristic algorithms [3].

1.2. Or-opt Algorithm

In the Or-opt method [3], we select successive 1 to 3 cities (block). We then insert them into another path. Fig. 1 shows the schematic diagram of the Or-opt algorithm. In the example shown in the figure, the block starts from the city 5 with the block size of 3. The block is inserted between the city 4 and the city 1. First, we take out the city 5 to the city 7 from the tour. Next, we connect the path between the city 2 and the city 3. In addition, the path from the city 4 to the city 1 is removed in order to insert the block. Finally, the block is inserted to the empty path to make a tour.

1.3. Quadratic Assignment Problem (QAP)

The QAP can be described as follows, given two matrices \( A \) (distance matrix) and \( B \) (flow matrix). The objective of the QAP is to find a permutation \( p \) of the elements which minimizes the following object function \( F \) given by

Figure 1: A schematic diagram of the Or-opt algorithm. For example, we create a block consists of the city 5 to the city 7. Then we insert the block in the path between the city 4 and the city 1.
\[ F = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} b_p(i)p(j) \]  
\[ \text{index} : 1, 2, \cdots, n \]
\[ p : \{p(1), p(2), \cdots, p(n)\} \]

where \( a_{ij} \) and \( b_j \) are the \((i, j)\)th elements of the matrices \( A \) and \( B \), respectively, \( p(i) \) is the \( i \)th element of \( p \), and \( n \) is the problem size. There are \( n! \) total combinations for the QAP, therefore, it is hard to obtain the exact solution in the case of large-scale QAPs.

2. Slide-and-Insert Assignment Method with Chaotic Neurodynamics

2.1. Slide-and-Insert Assignment Method

The slide-and-insert assignment method shown in Fig. 2 is based on the Or-opt algorithm for the TSPs.

First, we arbitrarily choose a city. The element assigned to the city is the starting point of the block.

Next, we select the city that will be the insert point of the block. For example, if the insert-point is the city \( r \), the block is inserted between the element \( p(r) \) and the element \( p(r+1) \).

In addition, we choose the block size which is less than or equal to 3. We need to make a space between \( p(r) \) and \( p(r+1) \) to insert the block. As shown in Fig. 2(a), we temporarily take out the block from the permutation \( p \), so that the cities of the block are now empty. The elements assigned to the cities next to the end of the block are reassigned to the empty cities by sliding them to create an empty space for the insert of the block as shown in Fig. 2(b). This reassignment is referred to as the slide operation.

In the slide operation, the elements move in one way to the descending order of the city numbers. In addition, if the enough numbers of successive elements for the slide operation cannot prepared, we use the elements assigned at city I, II, III, and so on.

Finally, the elements of the block are inserted into the cities which became empty as a result of the slide operation as shown in Figs. 2(c) and 2(d). This operation corresponds to the insert operation.

It should be noted that the slide-and-insert assignment method sometimes alters the solution dramatically. As a result, it cannot spend enough time for the local search. Thus, we take advantage of this property. That is, we first use the 2-opt algorithm (Fig. 3) for the local search [4]. When the 2-opt algorithm is trapped in the local minimum, we then apply the slide-and-insert assignment method to escape from the local minimum.

Fig. 3 shows an example of the 2-opt algorithm. First, we select the element \( i \). Next, we select the city \( j \) to which we assign the element \( i \). At the same time, the element \( p(j) \) that was assigned to the city \( j \) is assigned to the city \( q(i) \). The objective function will be improved until the algorithm is trapped in the local minimum because the 2-opt algorithm is a local search method.

In contrast, the objective function is rarely improved by the slide-and-insert assignment method. Thus, we use random numbers in the slide-and-insert operations in order to explore a large solution space regardless of the current solution.

Moreover, we introduce the chaotic dynamics into the slide-and-insert assignment method instead of the random numbers to further control the searching space.

2.2. Chaotic Neural Network

In this paper, we use the chaotic neural network model [5] as described in eqs. (4) to (7) [6][7].

\[ \xi_{ij}(t+1) = \max_k \{ \beta \Delta_{ij}(t) \} \]  
\[ \zeta_{ij}(t+1) = k_r \xi_{ij}(t) - \alpha x_{ij}(t) + \theta(1-k_r) \]  
\[ y_{ij}(t+1) = \xi_{ij}(t+1) + \zeta_{ij}(t+1) \]  
\[ x_{ij}(t+1) = \frac{1}{1 + \exp \left( \frac{-y_{ij}(t+1)}{t} \right)} \]

where

\[ \xi_{ij}(t+1) = \max \{ \beta \Delta_{ij}(t) \} \]  
\[ \zeta_{ij}(t+1) = k_r \xi_{ij}(t) - \alpha x_{ij}(t) + \theta(1-k_r) \]  
\[ y_{ij}(t+1) = \xi_{ij}(t+1) + \zeta_{ij}(t+1) \]  
\[ x_{ij}(t+1) = \frac{1}{1 + \exp \left( \frac{-y_{ij}(t+1)}{t} \right)} \]

Figure 2: A schematic diagram of the slide-and-insert assignment method. In this example, the starting point of the block is the city II, and the block size is 3. (a) We temporarily take out the block from the permutation \( p \). In other words, the cities II, III, and IV are made empty. (b) The elements of the city V to the city VI are reassigned to the empty cities through the sliding operation. (c) Finally, the elements of the block are inserted into the insert point, the city IV.

Figure 3: An example of the 2-opt algorithm where the element \( i \) is assigned to the \( j \)th location, and element \( p(j) \) is assigned to the \( q(i) \)th location.
where $\xi_{ij}(t + 1)$ is the gain effect, $\zeta_{ij}(t + 1)$ is the refractory effect, $\beta$ is a scaling parameter of the gain effect, $\kappa_t$ is a decay parameter of the refractory effect, $\alpha$ is a scaling parameter of the refractory effect, $\theta$ is a bias, $y_{ij}(t + 1)$ is the internal state, $\epsilon$ is the steepness of the output function, $\Delta_{ijk} = F_0 - F_{ijk}$ is the gain of the objective function value, $F_0$ is the current value of the objective function $F$, and $F_{ijk}$ is the value of $F$ after the slide-and-insert assignment method.

In addition, we pay attention to the number of change in the assignments (changes-in-assignment) through the slide-and-insert assignment. In the example shown in Fig. 2, the elements of the city II to the city VI are reassigned. Therefore, the changes-in-assignments is 5. In the slide-and-insert assignment algorithm, $j \neq 1$ and 2. This is because when $j=1$, there is no change in $p$, and when $j=2$, the method is equivalent to the 2-opt algorithm.

Fig. 4 shows a schematic diagram for the slide-and-insert assignment method with the chaotic neural network. As shown in the figure, we prepare $n \times n$ in the chaotic neurons configuring a chaotic neural network for the size-$n$ QAPs. In the figure, $i$ represents the starting point of the block, and $j$ corresponds to the changes-in-assignment. As shown in Fig. 4, $2n$ neurons will not be updated because $j \neq 1$ and 2. As a result, we can reduce the computational time by $2n$ neurons exploiting the changes-in-assignments.

Fig. 5 shows a flow chart for the proposed method with the chaotic dynamics. First, we generate the initial permutation using the random numbers. As shown in the figure, we next select the neuron which has not been updated yet. Then, we determine the block size by evaluating the gains for the block sizes from 1 to 3. The block size will be the size that gives the largest gain. After the block size is determined, we update the internal state of the neuron. If $x_{ij}(t + 1) > 0.5$ (the neuron fires), then we apply the slide-and-insert assignment method.

For example, if the $(2,5)$-neuron fires, the element is assigned to the city II is the start point of the block, and assigned elements from the city II to the city VI are reassigned using the slide-and-insert assignment method. Because $p$ is changed largely, the searching space may also be changed dramatically. Thus, we should check whether the sub-optimum solution exists in the current solution space by the 2-opt algorithm. We continue to apply the 2-opt exchanges until the algorithm is trapped to the local minimum.

One iteration is completed when all the neurons in the network are updated.

### 3. Simulation Results

We show the numerical simulation results for the proposed method with the chaotic neurodynamics. The results with the random numbers instead of the chaos are also shown for comparison. The random number was generated by the rand function in C. For both methods, we execute 1000 iterations and 30 trials.

We evaluate the average error rate (AER) from the optimum solution defined by

$$\text{AER} = \frac{1}{30} \sum_{t=1}^{30} (F_{\text{min}}(t) - F_{\text{opt}}) \times 100 \% \quad (8)$$
where $F_{\text{min}}(t)$ is the minimum objective function obtained during the $t$-th trial, and $F_{\text{opt}}$ is the optimum objective function.

The network parameters for the chaotic neural network used in the simulation for each problem are shown in Table 1.

Table 2 shows the AER for each problem with the chaotic dynamics and random numbers.

As shown in Table 2, the proposed method with the chaotic dynamics is better than that with the random numbers. However, as shown in Table 1, the value of $\beta$ are different for each problem. This is because the gains obtained by the slide-and-insert assignment method are different depending on the problems. Therefore, the dynamics of the neuron in the neural network may change according to the characteristics of the problems.

### 4. Conclusions

In this paper, we have proposed the slide-and-insert assignment method for the QAPs based on the Or-opt algorithm for the TSPs. We have also introduced the chaotic dynamics to the proposed method.

Numerical simulation results have shown that the proposed method with the chaotic dynamics is more effective than that with the random numbers. However, the searching ability is affected by the value of $\beta$. Therefore, we need to investigate the dynamics of the proposed method in order to determine the optimal parameter set.

### Acknowledgment

This work was supported by Kakenhi (20300085).

### References


Heuristics Methods for Asymmetric Traveling Salesman Problem and their Applications to DNA Fragment Assembly

Tomohiro Kato† and Mikio Hasegawa†

†Dept. of Electrical Engineering, Tokyo University of Science
1-14-6 Kudankita, Chiyoda-ku, Tokyo, 102-0073 Japan

Abstract—There are various applications of the Traveling Salesman Problems (TSPs) in a real world. In this paper, a DNA fragment assembly problem is studied as one of the applications of the TSP. The DNA fragment assembly problem is to build a DNA sequence from several hundreds of fragments obtained by the genome sequencer, which can be formulated as the asymmetric TSP. We apply heuristic methods for the asymmetric TSP, such as the tabu search, the simulated annealing, and the genetic algorithm, to the DNA fragment assembly problem. Our simulation results show that the proposed algorithm using the tabu search on a block shift operation exhibits the best performance for the asymmetric TSPs and the DNA fragment assembly problem.

1. Introduction

In recent years, genetic technologies are progressing dramatically. This is caused by progress of the sequencer used for analyses of the DNA sequences. The analyses of the DNA sequences is important for the medical field, agriculture, environmental issues, and so on. Improvement in the speed of the DNA sequencing technique leads to improvement in the speed of the bioinformatics. One of the reasons of recent progresses the DNA sequencer is development of shotgun sequencing method.

The shotgun sequencing method is one of the methods which estimate a long DNA sequence from a lot of fragments. A single sequencer cannot read a quite long sequence with a huge number of bases into a computer directly in a short time. Therefore, in the shotgun sequencing method, first, the DNA clones are generated using a vector. Then, the DNA sequence is cut into small fragments using a restriction enzyme or a physical shearing force. The sequencer inputs such small fragments into the computer, in which the original sequence is reconstructed by an assembly algorithm. In the reading phase, the fragments are cut randomly, and they may have the overlaps. Therefore, the original sequence can be estimated by the assembly algorithms by connecting them according to the overlaps. Such an optimal fragment arrangement search becomes an optimization problem [1], which can be formulated as a modified version of the Asymmetric Travelling Salesman Problem (ATSP).

The Traveling Salesperson Problem (TSP) is to find a minimum length tour that visits each city exactly once when a list of cities and their traveling costs are given. Since the TSP and the ATSP belongs to a class of NP-hard, various heuristic algorithms to find near optimal solutions have been proposed for the TSPs and the ATSPs. In this paper, various heuristic methods are applied to the ATSPs and the DNA fragment assembly problems, to investigate effective algorithm. We evaluate the performance of the Hill Climbing Method, the Simulated Annealing[2], the Tabu Search[3], and the Genetic Algorithms[4], with three types of local search heuristics. Based on the results on the ATSP, those methods are applied to the DNA fragment assembly problem.

2. The DNA fragment assembly problem

The DNA sequencer used in this research generates a lot of small fragments with some overlaps as shown in Fig.1. According to the overlaps of the fragments, they can be connected like a puzzle when reconstructing the original sequence. This is an optimization problem to rearranges such mixed fragments into the right order. We apply optimization methods for the ATSP to this problem, which finds the right order of the cities. The cost in the DNA fragment assembly problem is calculated by comparing the arrangements around the ends of the fragments. It is called the overlap which is calculated by the semi global alignment method [5].
3. Application of Heuristic Methods to Asymmetric Travelling Salesman Problems

The cost function of the TSP is the total length of the tour, which is formulated as follows,

\[ f_{\text{per}} = \sum_{k=1}^{n} d_{r \mod (k+1) \mod n} \]  

where \( d_{i,j} \) is the cost when the salesman goes to the city \( j \) from the city \( i \), \( r(k) \) is the city visited at the \( k \)th order, \( n \) is the number of cities, respectively. In ATSP, it is not necessarily \( d_{i,j} = d_{j,i} \).

In this paper, to investigate effective methods for the ATSP, we use various heuristic methods based on three types of updating on two different definitions of neighboring solution. As the definition of the neighboring solutions, 2-exchange(2-ex) and two types of blockshift operations are introduced.

- **2-exchange (2-ex)**

  In the 2-exchange, the neighboring solutions are generated by a simple exchange of two cities. In this paper, we defined it as follows, when \( (i, j) \) is selected, the city \( j \) will be move to the position after the city \( i \). For example, when \( (1, 5) \) is chosen for \( (1, 8, 9, 6, 4, 7, 5, 2, 10) \), it is updated to \( (1, 5, 9, 6, 4, 7, 6, 2, 10) \).

- **BlockShift (BS)**

  In the blockshift operation, a selected block is moved to the other position, with keeping the order in the block. In our definition, when \( (i, j) \) is selected, the block starting from the city \( j \) is move just after the city \( i \). In this paper, we use two types of block sizes. The first one(BS1) fixes it at 3, and the second one(BS2) uses the best size at each iteration. For example, when \( (1, 5) \) is chosen for \( (1, 8, 9, 6, 4, 7, 5, 2, 10) \), it is updated to \( (1, 5, 2, 10, 9, 6, 4, 7, 6) \).

As the solution improvement method applied to the updating method to the neighboring solutions defined above, we introduce the following three algorithms, the hill climbing method(HC), the simulated annealing(SA), and the tabu search(TS).

- **Hill Climbing Method (HC)**

  When the better neighboring solution is found, the solution is immediately updated to the neighbor. However, this approach has a local minimum problem.

- **Simulated Annealing (SA)**

  The solution is updated when Eq. (2) is satisfied,

  \[ \text{rand} < \frac{1}{1 + e^{-\Delta_{i,j}/T}} \]  

  where \( \Delta_{i,j} \) is the improvement of the objective function by moving to the neighbor \( (i, j) \), \( T \) is the temperature, \( \text{rand} \) is the uniformly distributed random number between 0 and 1. Whenever iteration increases, temperature \( T \) is lowered, and the probability of selecting the better solution is increased.

- **Tabu Search (TS)**

  Once a move \( (i, j) \) is applied to updating the solution, the corresponding move is memorized in a tabu list. The moves in the tabu list are forbidden for a fixed term, called a tabu tenure. The TS updates the solution to the neighbor \( (i, j) \), which has the largest \( \Delta_{i,j} \) in non-tabu ones.

As a globally searching heuristic method, we introduce the genetic algorithm(GA), which has been applied to DNA fragment assembly problem in Ref. [4].

- **Genetic Algorithm (GA)**

  First, many solutions are created at random. Two solutions are selected from the set of the current solutions, and crossover operation is applied to create new solutions. Then, the worst ones in all of solutions in the current set are removed, that is called mutation. The crossover operation and the mutation procedures are repeated, and better solutions are found by such mixing of better solutions.

To evaluated performance of three algorithms on ATSPs, we generated benchmark problems whose sizes are 10 to 70, by combining eil76 and st70 in TSPLIB[10]. The results on the ATSPs are shown in Table 1. For each algorithm, the initial solution is generated by greedy algorithm connecting small cost links.

Table 1 shows that the GA exhibits the best only for the 10 city problem and the BS2 updated by the TS is the best for all other larger problems. The block shift is better updating method to the neighboring solution. The TS is the better SA for update decision method. From these results, we confirm that the BS2 and the TS is the best for ATSP in these algorithms.

4. Application of Heuristic Methods to the DNA Fragment Assembly Problem

In this research, the DNA fragment assembly problem is solved in a similar way to the ATSPs. A fragment is corresponding to a city, and an overlap to a cost. The cost function of the DNA fragment assembly problem is the total length of the sequence which is formulated as follows,
Table 1: The simulation result of various heuristic methods, based on the two exchange, the block shift, the hill climbing, the simulated annealing, the tabu search, and the genetic algorithm, for ATSPs whose sizes are 10 to 70.

<table>
<thead>
<tr>
<th>Method</th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
<th>60</th>
<th>70</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-ex with HC</td>
<td>161.1</td>
<td>307.9</td>
<td>432.5</td>
<td>503.1</td>
<td>563.2</td>
<td>644.4</td>
<td>738.6</td>
</tr>
<tr>
<td>2-ex with SA</td>
<td>147.2</td>
<td>286.3</td>
<td>414.1</td>
<td>475.2</td>
<td>510.7</td>
<td>620.6</td>
<td>721.9</td>
</tr>
<tr>
<td>2-ex with TS</td>
<td>143.2</td>
<td>274.2</td>
<td>402.6</td>
<td>460.3</td>
<td>503.8</td>
<td>609.8</td>
<td>700.4</td>
</tr>
<tr>
<td>BSI with HC</td>
<td>165.5</td>
<td>313.3</td>
<td>431.6</td>
<td>489.9</td>
<td>541</td>
<td>634.4</td>
<td>723.5</td>
</tr>
<tr>
<td>BSI with SA</td>
<td>146.5</td>
<td>286.7</td>
<td>415.7</td>
<td>462.4</td>
<td>498.8</td>
<td>617.2</td>
<td>681.7</td>
</tr>
<tr>
<td>BSI with TS</td>
<td>150.6</td>
<td>277.1</td>
<td>387.5</td>
<td>443.9</td>
<td>478.2</td>
<td>593.5</td>
<td>662.2</td>
</tr>
<tr>
<td>BS2 with HS</td>
<td>147.1</td>
<td>276.8</td>
<td>401.9</td>
<td>451.3</td>
<td>498.2</td>
<td>603.2</td>
<td>698.7</td>
</tr>
<tr>
<td>BS2 with SA</td>
<td>145.1</td>
<td>275.6</td>
<td>400.4</td>
<td>444.3</td>
<td>479.0</td>
<td>581.4</td>
<td>640.5</td>
</tr>
<tr>
<td>BS2 with TS</td>
<td>146.8</td>
<td>271.6</td>
<td>384.9</td>
<td>439.5</td>
<td>448.2</td>
<td>555.2</td>
<td>617.7</td>
</tr>
<tr>
<td>GA</td>
<td>140.5</td>
<td>283.6</td>
<td>451.6</td>
<td>522.5</td>
<td>689.5</td>
<td>765.0</td>
<td>904.0</td>
</tr>
</tbody>
</table>

\[ f_{DG}(j) = \sum_{k=1}^{n} l_{k} - \sum_{k=1}^{n-1} overlap_{r(k),r(k+1)}, \quad (3) \]

where \( overlap_{j,j} \) is the overlap for connecting the fragment \( j \) next to the fragment \( i \), \( r(k) \) is the fragment in the \( k \) th order, \( n \) is the number of fragments, respectively. The overlaps between two fragments are calculated by semi-global alignment method[5]. The target of the DNA fragment assembly problem is to search a shortest sequence by including longer overlaps. The shortest sequence may almost the same as the original sequence.

We have chosen three sequences from the NCBI[11]: a human MHC class II region DNA with fibronectin type II repeats HUMMCHB, with accession number X60189, which is 3835 bases long; a human apolipoprotein HUMAPOBF, with accession number M15421, which is 10,089 bases long; and the complete genome of bacteriophage lambda, with accession number J02459, which is 20,014 bases long. By amplifying each genome data and fragmenting them, we generated the DNA fragment assembly problems shown in Table 2. The rate of amplification of each genome data is called a coverage.

We apply ten heuristic methods used in the previous section to these DNA fragment assembly problems, and estimate the original sequence using the order of fragments obtained by the optimization algorithms. To evaluate the correctness of solution, the obtained sequences by each optimization algorithm are compared with the original sequence using the global alignment [6] and a similar rate is calculated, which is defined as follows,

\[ \text{Similar Rate} = \frac{AS_{\text{sim} \text{new}}}{AS_{\text{sim} \text{in}}} \times 100\%, \quad (4) \]

where \( AS_{\text{sim} \text{new}} \) is the alignment score which compares the input sequence with the output sequence, \( AS_{\text{sim} \text{in}} \) is the alignment score which compares the input sequence with itself. The shortest sequence found by minimizing the objective function in Eq.3 may have higher similarity to the original sequences.

The results of the heuristic algorithm applied to the ATSP are shown in Table 3 by the similar rate. Table 3 shows that the BS2 updated by the TS is the best for all problems. This algorithm was also the best for the ATSP in previous section. From these results, we confirm that the BS2 updated by the TS is the best for the DNA fragment assembly problem in the heuristic algorithms introduced in the paper.

Table 2: The DNA fragment assembly problems used in this paper.

<table>
<thead>
<tr>
<th>Coverage</th>
<th>5</th>
<th>6</th>
<th>5</th>
<th>7</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of fragments</td>
<td>60</td>
<td>72</td>
<td>125</td>
<td>175</td>
<td>350</td>
</tr>
</tbody>
</table>

5. Conclusions

In this research, we examined the effectiveness of the heuristics methods of the ATSP for the DNA fragmentation assembly problems. First, we applied various heuristic methods to the benchmark ATSPs. Our results shows that the blockshift operation updated by the TS is better than the GA and the SA for the ATSPs whose size are larger than 20. Next, we applied those heuristics methods to the DNA fragmentation assembly problems. Our results show that the same algorithm the blockshift with the TS exhibits better than other algorithms for all problems. From these results we confirm that the BS2 updated by the TS is effective for the DNA fragmentation assembly problem.

As future works, we will apply the chaotic algorithm [9] to DNA fragment assembly problem. Since the chaotic algorithms have been shown more effective than the TS, it may be even better algorithm for finding optimal se-
Table 3: The simulation result of various heuristic methods, based on the two exchange, the block shift, the hill climbing, the simulated annealing, the tabu search, and the genetic algorithm, for five DNA fragment assembly problems. The results are evaluated by the SimilarRate.

<table>
<thead>
<tr>
<th></th>
<th>X60189(5)</th>
<th>X60189(6)</th>
<th>M15421(5)</th>
<th>M15421(7)</th>
<th>J02459(7)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-ex with HC</td>
<td>87.318</td>
<td>86.897</td>
<td>85.517</td>
<td>81.781</td>
<td>72.451</td>
</tr>
<tr>
<td>2-ex with SA</td>
<td>89.533</td>
<td>89.039</td>
<td>87.053</td>
<td>85.437</td>
<td>76.224</td>
</tr>
<tr>
<td>2-ex with TS</td>
<td>89.402</td>
<td>89.414</td>
<td>87.269</td>
<td>86.443</td>
<td>76.394</td>
</tr>
<tr>
<td>BS1 with HC</td>
<td>88.226</td>
<td>87.237</td>
<td>85.699</td>
<td>81.973</td>
<td>74.968</td>
</tr>
<tr>
<td>BS1 with SA</td>
<td>90.372</td>
<td>89.131</td>
<td>87.089</td>
<td>85.737</td>
<td>76.249</td>
</tr>
<tr>
<td>BS1 with TS</td>
<td>90.024</td>
<td>88.960</td>
<td>87.237</td>
<td>86.895</td>
<td>78.512</td>
</tr>
<tr>
<td>BS2 with HS</td>
<td>97.850</td>
<td>97.414</td>
<td>89.068</td>
<td>86.278</td>
<td>80.391</td>
</tr>
<tr>
<td>BS2 with SA</td>
<td>97.159</td>
<td>97.350</td>
<td>88.977</td>
<td>86.767</td>
<td>83.682</td>
</tr>
<tr>
<td>BS2 with TS</td>
<td>98.909</td>
<td>97.842</td>
<td>90.029</td>
<td>88.728</td>
<td>85.293</td>
</tr>
</tbody>
</table>

References


A relay sensor node selection scheme in wireless sensor networks using a chaotic neural network

Kyohei Fujii†, Tomoyuki Sasaki†, Hidehiro Nakano†, Akihide Utani‡, Arata Miyauchi‡, and Hisao Yamamoto†

††Tokyo City University
1-28-1, Tamazutsumi, Setagaya-ku, Tokyo, 158-8557 Japan
Email: †{fujii, sasaki, nakano, miyauchi}@ic.cs.tcu.ac.jp, ‡{autani, yamahisa}@tcu.ac.jp

Abstract—Recently, Wireless Sensor Networks (WSNs) have been studied with a great amount of interests. In WSN, flooding is required for the dissemination of queries and event announcements. The original flooding causes the overlap problems. In the original flooding, each sensor node receiving a broadcast message forwards it to its neighbors, resulting in a lot of collisions and duplicate messages. In this study, we use a Chaotic Neural Networks (CNN) to selection of forwarding nodes for the dissemination of queries and event announcements. We evaluate the applicability of CNN by computer simulations and discuss its development potential.

1. Introduction

Recently, Wireless Sensor Networks (WSNs) have been studied with a great amount of interests [1]-[4]. In WSN, many sensor nodes are set up in an observation area. Sensing information of each sensor node is transmitted to a sink node by multi-hop wireless communications. Then, the observation in the large-scale area is possible. In general, each sensor node has only the limited function, and has a restriction in energy consumption. Therefore, it is necessary to control the communication load in order to prolong lifetime of WSN. In WSN, flooding is required for the dissemination of queries and event announcements. The original flooding causes the overlap problems. In the original flooding, each sensor node receiving a broadcast message forwards it to its neighbors, resulting in a lot of collisions and duplicate messages. For dense WSNs, the impact caused by the original flooding may be overwhelming. The original flooding may result in the reduced network lifetime. Therefore, the selection of forwarding nodes for the dissemination of queries and event announcements is needed to prolong the lifetime of WSNs. In many cases, all nodes do not have to broadcast, and it is possible to transmit information from a sink node to all nodes by appropriately selecting some Forwarding Nodes (FNs). This problem is referred to as Forwarding Node Selection Problem (FNSP). For the energy saving, the number of FNs should be minimized. However, if specific FNs are always selected, the FNs consume a lot of energy. Therefore, it is important to find plural solutions for FNSP and to switch them periodically. Then, the communication load of each sensor node is distributed. In our previous works, we have proposed a method for solving FNSP using a Chaotic Neural Network [5]. This method can search plural optimum selection patterns of FNs, efficiently [6]. This paper investigates performances of the method in detail in the viewpoints of long-term operation of WSNs. Some numerical simulation results are shown.

2. Forwarding Node Selection Problem (FNSP)

In this section, a problem named Forwarding Node Selection Problem (FNSP) in Wireless Sensor Networks (WSNs) is explained. In WSN, sensor nodes and a sink node are set up in an observation area. Each sensor node has a specific radio range, and transmits sensor information to the sink node by multi-hop wireless communications between sensor nodes which exist within the radio range to each other. In WSN, flooding is required for the dissemination of queries and event announcements. In the original flooding, each sensor node receiving a broadcast message forwards it to its neighbors, resulting in a lot of collisions and duplicate message. Therefore, the selection of Forwarding Nodes (FNs) is needed to prolong lifetime of WSN. As the selected FNs receive the broadcast messages, the FNs forward it to their neighbors. The other nodes only receive a broadcast message, and do not forward it. These nodes are referred to as Receiving Nodes (RNs). In order to prolong lifetime of WSN, the number of FNs should be minimized. However, the constrained conditions such that all the sensor nodes can receive broadcast messages should be satisfied. This problem is called FNSP. In FNSP, we assume that FNs are selected by a sink node. Therefore, the original flooding is assumed to be required only at first so that the sink node gathers location information of each sensor node. Fig. 1 illustrates an example of FNSP.
3. Conventional method

3.1. Chaotic Neural Network (CNN) for FNSP

In this section, a method for solving FNSP using a chaotic neural network (CNN) is explained [6]. First, a single sensor node is expressed by a single neuron. Each sensor node can directly communicate to the other sensor nodes within their radio range to each other. In this method, neurons corresponding to such sensor nodes are connected to each other. If a neuron fires, the neuron is regarded as FN. Otherwise, the neuron is regarded as RN. The model of CNN for FNSP is described by the following equations.

\[ i(t+1) = k_i i(t) + C_i W_i \]  
\[ \eta_i(t+1) = k_\eta \eta_i(t) - C_\eta \sum_{j \in \text{radio range}} x_j(t) \]  
\[ \zeta_i(t+1) = k_\zeta \zeta_i(t) - \alpha x_i(t) + a \]

where \( i \), \( \eta \), and \( \zeta \) are internal states of the \( i \) th neuron for external input, mutual connection and refractoriness, respectively. \( W_i \) denotes the number of sensor nodes within radio range of the \( i \) th sensor nodes. \( x_j \) denotes the outputs of other neurons corresponding to sensor nodes within radio range of the \( i \) th sensor node, and \( x_i \) denotes the output of the \( i \) th neuron. The output of the \( i \) th neuron is described by the following equations.

\[ x(t+1) = f\{y(t+1)\} \]  
\[ y_i(t+1) = \xi_i(t) + \eta_i(t) + \zeta_i(t) \]  
\[ f(x) = \frac{1}{1 + exp(-\frac{x}{\varepsilon})} \]

This model has 8 parameters: \( k_i, k_\eta, k_\zeta, C_i, C_\eta, C_\zeta, \alpha, a, \) and \( \varepsilon \).

3.2. Firing Decision Method (FDM)

After calculating all output values \( x_i \), FNs are decided. However, it is difficult to always obtain executable solutions that satisfy the constrained condition. Then, the following Firing Decision Method (FDM) is introduced. First, let the sink node be FN, and let sensor nodes within radio range of the FN be RNs. Next, as shown in Fig.2, let an RN existing outside an exception range \( E_r \) and having maximum output in the RNs be a new FN. Repeating in this manner for decided FNs, a selection pattern of FNs is obtained at every iteration. Then, selection patterns are found in all iteration. The additional parameter \( E_r \) is important for the solving performance of FNSP. Especially, we focus on this parameter in this paper.

4. Purpose of this study

In our previous works a method to solve FNSP using CNN has been proposed as explained in Sec.3 [6]. However, consideration for the settings of appropriate parameters to target environments has not been sufficient so far. When the parameter \( E_r \) is large, a sensor node distant from presently decided FNs becomes a new FN in performing FDM. As a result, it causes a decrease of the number of FNs. However, FNs such that all sensor nodes can receive broadcast message might not be decided because the number of candidates of FNs decreases in performing FDM. That is, searching solution by FDM might not be success. When \( E_r \) is small, a sensor node near presently decided FNs tends to be selected as a new FN. Then, the total number of FNs can increase. Moreover, in Ref.[6], only simulation results for uniform distribution WSNs have been presented. In actual use of WSNs, many sensor nodes are set up around a sink node because sensor nodes around the sink node consumes a lot of energy in relaying sensor information. It is necessary to realize the load-balancing of all the sensor nodes. This paper also considers solving performances in such a situation.
5. Experiment

The purpose of FNSP is to minimize the number of FNs, and to obtain plural selection patterns of FNs. It is important to achieve long-term operation of WSN. The proposed method is applied to 4 kinds of WSNs as shown in Table 1, and other fixed settings are shown in Table 2. The solving performances for the parameters \( E_r \) are investigated by varying \( E_r \) from 0 to 20 (the size of radio range). The parameters in CNN are fixed as shown in Table 3. Fig.3 shows examples of locations of 500 sensor nodes. Fig.4 shows relations between \( E_r \) and the number of obtained FN selection patterns with search success rate. As shown in Fig.4, it can be confirmed that as \( E_r \) increases the search success rate decreases simultaneously, and the number of selection patterns increases. Fig.5 shows transitions of the number of FNs for a single trial. As shown in Fig.5, the obtained minimum number of FNs is 4 in this experiment. However, it can be confirmed that FNs more than 4 have been decided in many iterations. So, the selection patterns consisting more than 4 FNs are focused on and non-overlapping patterns to each other are extracted. Fig.6 shows an example of obtained FN selection patterns without overlapping. Fig.7 shows overwriting of these selection patterns. It can be confirmed that all selected FNs do not overlap to each other. The best size of \( E_r \) such that the most number of non-overlapping patterns can be obtained is different depending on WSN environments as shown in Table 4. In the table, the ratio of \( E_r \) to radio range is also shown. It is thought that it is impossible to decide appropriate value of \( E_r \) from these results. However, it was able to be confirmed that the size of \( E_r \) increased, too, as the number of FNs increased. Therefore, the size of \( E_r \) in case of number 100 of sensor can say that 88.5% is effective the size of ExceptionRange in case.
of number 500 of sensors by 76.0%. For dense WSNs, the value of $E_r$ can be larger and a lot of selection patterns do not overlap to each other. However, too large values of $E_r$ decreases search success rate. For the problem, we should consider improvement of the model of CNN and the algorithm of FDM.

6. Conclusions

We have presented a method for solving a forwarding node selection problem in wireless sensor networks using a chaotic neural network. In the simulation, solving performances for a parameter in deciding forwarding nodes have been analyzed in detail. This method can obtain a lot of effective selection patterns of the forwarding nodes depending on the parameter. Future problems include improvement of the model of CNN and the algorithm in deciding forwarding nodes, and analysis of lifetime of wireless sensor networks in using this method.

References


Table 4: Size of $E_r$ in which the most number of non-overlapping patterns are obtained

<table>
<thead>
<tr>
<th>WSN</th>
<th>$E_r$ (%)</th>
<th>Average number of patterns</th>
</tr>
</thead>
<tbody>
<tr>
<td>WSN1</td>
<td>16.0 (80.0)</td>
<td>7.3</td>
</tr>
<tr>
<td>WSN2</td>
<td>14.5 (72.5)</td>
<td>4.6</td>
</tr>
<tr>
<td>WSN3</td>
<td>17.5 (87.5)</td>
<td>6.8</td>
</tr>
<tr>
<td>WSN4</td>
<td>16.0 (80.0)</td>
<td>5.8</td>
</tr>
</tbody>
</table>


Solving a sink node allocation problem in wireless sensor networks using a competitive particle swarm optimization

Yuta Kanamori†, Yu Taguchi†, Hidehiro Nakano†, Akihide Utani‡, Arata Miyauchi† and Hisao Yamamoto‡

†Tokyo City University
1-28-1, Tamazutsumi, Setagaya-ku, Tokyo, 158-8557 Japan
Email: †[kanamori,taguchi,nakano,miyauchi]@ic.cs.tcu.ac.jp, ‡[autani, yamahisa]@tcu.ac.jp

Abstract—Wireless Sensor Networks (WSNs) have attracted a significant amount of interest from many researchers for a wide range of applications, such as natural environmental monitoring and environmental control in residential spaces or factories. To realize long-term operation of WSNs, we discuss in this study a method of suppressing the communication load on sensor nodes by effectively placing a limited number of sink nodes in an observation area that integrate sensing data from nodes around them. As a technique of solving effective locations for sink nodes, this paper proposes a method using a simple competitive PSO for finding plural acceptable solutions. The simulation results show that obtained solutions can contribute to prolonging lifetime of WSNs.

1. Introduction

There is growing expectation for Wireless Sensor Networks (WSNs) as a means of realizing various applications, such as natural environmental monitoring and environmental control in residential spaces or factories[1]-[6]. In WSNs, hundreds or thousands of micro-sensor nodes are deployed in an observation area and sensor information of each node is gathered to sink nodes by inter-node wireless communication. To realize long-term operation of WSNs, it is necessary to gather sensor information efficiently by saving node power consumption. Ant-based routing algorithms[2]-[3], synchronization-based data gathering schemes[4]-[5] and clustering-based data gathering schemes[6], are under study as communication methods to prolong the lifetime of WSNs. We discuss in this study a method of suppressing the communication load (transmission-reception power) on sensor nodes by effectively placing a limited number of sink nodes in an observation area. As a technique of solving effective locations for sink nodes, we have proposed Suppression PSO (SPSO)[7] that is a fusion algorithm of Particle Swarm Optimization (PSO)[8] and Immune Algorithm[9]. SPSO can find plural allocation patterns of sink nodes. As the patterns are switched dynamically, long-term operation of WSNs can be realized. However, each solution performance of SPSO is often lower than that of original PSO, and SPSO provides different number of solutions for every trial depending on initial states. In addition, SPSO has many parameters, and the control of them is difficult. This paper proposes a method using a more effective method using a simple Competitive PSO (CPSO)[10]. In CPSO plural acceptable solutions can be found by parallel processing and the control is easy by adjusting a single parameter. Through numerical simulations, we show that the proposed method can find plural candidates for effective allocations of sink nodes.

2. Wireless Sensor Networks (WSNs)

In WSNs, sensor nodes monitor status information around them in an observation area, and transmit sensing data to sink nodes by multi-hop wireless communication[1]. In this paper we discuss a method of suppressing the communication load on sensor nodes by effectively placing a limited number of sink nodes in an observation area that integrate sensing data from sensor nodes around them. However, communication load is concentrated on sensor nodes around a sink node during the operation process of WSNs and causes them to break away from the network early. Therefore, as shown in Fig.1, it is needed to find plural allocations of sink nodes so that total hops in all sensor nodes are minimized, and to switch their allocations dynamically considering energy consumption of each sensor node. This problem is refereed to as a sink node allocation problem which is a kind of optimization problems. For solving this problem Suppression Particle
Swarm Optimization (SPSO) has been proposed[7]. In this paper we propose a new method for long-term operation of WSNs using a simple Competitive PSO (CPSO)[10].

3. Particle Swarm Optimization (PSO)

3.1. Original PSO

Generally, optimization problems in a real world require providing effective semi-optimal solutions in actual and reasonable computation time rather than providing a strict optimal solution in long computation time. One of them, there exists Particle Swarm Optimization (PSO) as the method to solve such problems[8]. PSO is a kind of metaheuristic algorithms emulating actions in swarms such as birds and fishes. These swarms decide actions to consider status information not only as each individual but also as whole of their swarms. In PSO each particle has a velocity vector and a position vector. The velocity vector of a particle \( v_{t+1} \) is given by the following equation.

\[
v_{t+1} = w v_t + c_1 \cdot \text{rand} \cdot (pbest_i - x_t) + c_2 \cdot \text{rand} \cdot (gbest - x_t) \tag{1}
\]

where \( pbest \) is a personal best solution which each particle has. \( gbest \) is a global best solution which all particles have. \( v_t \) is a current velocity vector. \( \text{rand} \) is the uniform random numbers for [0,1]. \( w \) is the inertia coefficient. \( c_1 \) and \( c_2 \) are the weight coefficients. The position vector \( x_{t+1} \) of a particle is given by the following equation.

\[
x_{t+1} = x_t + v_{t+1} \tag{2}
\]

PSO can fast solve various optimization problems in nonlinear continuous functions, although the algorithm uses only simple and fundamental arithmetic operations. However, a basic PSO can find only a single solution for a single trial.

3.2. Conventional Method: Suppression PSO

There has been a method of solving sink node allocation problems using Suppression PSO (SPSO) and the effectiveness has been presented[7]. SPSO has a simple self control mechanism and a memory mechanism like Immune Algorithm (IA)[9]. A simple self control mechanism suppresses searching plural similar solutions, and a memory mechanism saves plural different acceptable solutions. In Ref.[7], SPSO was compared with simple IA and PSO, and it was shown that SPSO was the most effective method for the sink node allocation problem. However, each solution performance of SPSO is often lower than that of original PSO, and SPSO provides different number of solutions for every trial depending on initial states. In addition, SPSO has many parameters, and the control of them is difficult.

3.3. Proposed Method: Competitive PSO

In this paper, we propose a method for solving sink node allocation problem using a simple Competitive PSO (CPSO) that can efficiently find plural different acceptable solutions by dividing particles into plural groups[10]. In the original PSO, it is difficult to find plural solutions because all the particles search a single solution by moving toward global best solution. So, in the CPSO, it is considered that particles are divided into arbitrary \( n \) groups. In addition, these groups have own local best solution (lbest) instead of global best solution as shown in Fig.2. As a result, plural solutions can be found because particles move toward each own lbest. Each group has a range in which the group search a solution preferentially. If a particle belonging to the \( i \)th group goes into the range of the other group, the particle is excepted from a candidate in updating the \( i \)th lbest. Each group has a range in which the group search a solution preferentially. If a particle belonging to the \( i \)th group goes into the range of the other group, the particle is excepted from a candidate in updating the \( i \)th lbest. Therefore, it is possible to search plural different solutions efficiently because each group does not go into the ranges of the other groups to each other. This range is referred to as priority search range. When a group can not search any solutions by always overlapping the ranges of the other groups, the group can obtain no solution because its lbest is reset at random every time.

CPSO can effectively find desired plural acceptable solutions and can easily control them by adjusting a single parameter for the range. Also, a group with the best priority can have almost the same solution performance to the original PSO. As relative works to our method. There has been PSO with Tabu Search[11]. This method can find effective solutions using the history of personal best solutions. However, this approach is different from searching plural acceptable solutions. The proposed PSO is not sequential search method like general tabu search but a parallel search method moving priority search regions dynamically. Therefore, the proposed CPSO can fast find plural solutions without repeating many trials.

4. Experiment

In order to confirm effectiveness of the proposed method, three methods, PSO, SPSO, and CPSO, are applied to a
sink node allocation problem described below, and compare the solving performances.

The problem to allocate five sink nodes in an observation area and to obtain plural allocation patterns is considered. Sink nodes can be allocated at the arbitrary positions in an observation area. Each particle has 10 dimensional position (and velocity) vector consisting of 2 dimensional locations of 5 sink nodes. This is a problem to search effective plural allocation patterns of sink nodes in order to suppress the communication load of sensor nodes.

The evaluation value (fitness) of each particle is defined by the following average hop counts.

$$fitness = \frac{\sum_{i=1}^{S} hop_{count_i}}{S} \quad (3)$$

where $S$ is the number of sensor nodes. $hop_{count_i}$ is the number of hops from the $i$-th sensor node to the nearest sink node. This fitness is used for all methods: PSO, SPSO, and CPSO.

The conditions in WSNs are shown in Table 1 and the parameters of each method are shown in Table 2. They were decided in preliminary experiments. In CPSO, the number of groups, $n$, is changed from 2 to 7.

For the sink node locations provided with each method, lifetime of sensor nodes is calculated. Each sensor node periodically transmits sensor information to the nearest sink node. Then, the sensor node and relative relay nodes consume energy[6]. If battery shutoff in a relay node, the node cannot relay sensor information. Each transmitting sensor node is assumed to recognize residual energy of its neighbor sensor nodes, and to select a receiving sensor node having higher residual energy. In such a situation, we evaluate Average Delivery Ratio (ADR) for the WSN.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area Size</td>
<td>500x500</td>
</tr>
<tr>
<td>Number of sensor node</td>
<td>1000</td>
</tr>
<tr>
<td>Number of sink nodes</td>
<td>5</td>
</tr>
<tr>
<td>Radio range</td>
<td>25</td>
</tr>
<tr>
<td>Total number of iterations</td>
<td>200</td>
</tr>
</tbody>
</table>

### Table 2: Parameters in each method

<table>
<thead>
<tr>
<th>Parameter</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inertia coefficient $w$</td>
<td>0.9</td>
</tr>
<tr>
<td>Weight coefficient $c_1$</td>
<td>1.0</td>
</tr>
<tr>
<td>Weight coefficient $c_2$</td>
<td>1.0</td>
</tr>
<tr>
<td>No. of group</td>
<td>2~7</td>
</tr>
<tr>
<td>Number of particles</td>
<td>$n \times 100$</td>
</tr>
<tr>
<td>Priority search range</td>
<td>50</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
<th>4th</th>
<th>5th</th>
<th>6th</th>
<th>7th</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSO</td>
<td>5.20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPSO(n=2)</td>
<td>5.17</td>
<td>5.26</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPSO(n=3)</td>
<td>5.16</td>
<td>5.22</td>
<td>5.43</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPSO(n=4)</td>
<td>5.16</td>
<td>5.22</td>
<td>5.42</td>
<td>5.59</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPSO(n=5)</td>
<td>5.16</td>
<td>5.22</td>
<td>5.40</td>
<td>5.57</td>
<td>5.59</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPSO(n=6)</td>
<td>5.16</td>
<td>5.21</td>
<td>5.39</td>
<td>5.55</td>
<td>5.76</td>
<td>6.18</td>
<td></td>
</tr>
<tr>
<td>CPSO(n=7)</td>
<td>5.16</td>
<td>5.22</td>
<td>5.39</td>
<td>5.57</td>
<td>5.80</td>
<td>6.36</td>
<td>7.27</td>
</tr>
<tr>
<td>SPSO</td>
<td>5.54</td>
<td>5.80</td>
<td>5.92</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3 shows the average fitness in each method. These are the average values for 100 trials. In the table, sorted fitness values are shown. Comparing the original PSO and CPSO, values of the best fitness (1st) are almost the same. In CPSO parallel search is possible in the solution space by groups of particles. Therefore, two or more solutions can be obtained. Meanwhile, in PSO the fitness converges to a single solution and it is not possible to search other solutions. In CPSO in $n = 7$, the 7th fitness is much worse than the other fitness. If a group has lower evaluation value than the other groups, the priority of the group for searching solution becomes lower. Depending on the solution space, groups having lower priority might not be able to find any acceptable solutions. We have confirmed that the 7th fitness does not converge.

In SPSO it is possible to search widely in the solution space by the self control mechanism and fitness does not converge monotonously. However, as comparing qualities of solutions, SPSO is worse than PSO and CPSO. In SPSO, self-control is applied if particles converge to the same position. Then, plural solutions can be obtained. This scheme causes insufficient search to each solution. SPSO has many parameters, and the control of them is difficult. On the other hand, qualities of solutions in CPSO can be controlled easily by adjusting the parameter of the priority search region, and can be better than those in SPSO. In SPSO the number of solutions saved on memory can be different for every trial. However, in CPSO the desired number of solutions can be easily obtained by changing the number of groups. Therefore, CPSO can effectively find desired plural acceptable solutions and can easily control them. They are advantages of CPSO.

Next, allocations of sink nodes finally obtained by CPSO in $n = 3$ are shown in Fig.3. In the figure, the circles represent the radio range. It should be noted that in allocations of all the sink nodes do not overlap to each other. This is very important in the viewpoints of suppressing communication load in each sensor node.

Finally, Fig.4 shows ADR for four methods: “Regular” is the method that sink nodes are allocated regularly in the area. “PSO” is the method that single allocation of sink nodes obtained by PSO is always selected. “CPSO”
is the method that allocations of sink nodes obtained by CPSO are switched in every 900/n iteration. “SPSO” is the method that three allocations of sink nodes obtained by SPSO are switched in every 300 iteration. It is found that CPSO (n = 6) shows the best performance in all the methods. “CPSO” and “SPSO” can keep higher ADR than “PSO” and “Regular”. Because, communication load in each sensor node is distributed by switching allocations of sink nodes. “CPSO” can keep higher ADR than “SPSO”. Because, the energy consumption of all the sensor nodes can be balanced by switching allocations and suppressing total hops of all the sensor nodes. Therefore, it is shown that CPSO is more effective for the long-term operation of WSN.

5. Conclusions

In this study, we have discussed a method of placing sink nodes effectively in an observation area to operate Wireless Sensor Networks (WSNs) for a long time. For the effective search of sink node locations, this paper has proposed a method using a simple competitive PSO for finding plural acceptable solutions. For prolonging lifetime of WSNs, it is important to provide several candidate locations for sink nodes by using a method capable of searching several acceptable solutions. In the simulation experiment, the effectiveness of the proposed method has been verified by comparison with Particle Swarm Optimization and Suppression Particle Swarm Optimization.

References

A Brief Overview of Some Recent Advances in Pinning Control of Complex Networks

Jinhu Lü†*, Xinghuo Yu†, and David Hill‡

† School of Electrical and Computer Engineering, RMIT University, Melbourne, VIC 3001, Australia
* LSC, Institute of Systems Science, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, P. R. China
‡ Research School of Information Sciences and Engineering, Australian National University, Canberra, ACT 0200, Australia
Email: jhlu@iss.ac.cn, x.yu@rmit.edu.au, David.Hill@anu.edu.au

Abstract—Complex networks are everywhere. Over the last ten years, various approaches have been proposed for controlling complex networks. Pinning control, as an effective method for controlling complex networks, has received increasing attention in recent years. This paper will briefly review some main advances in pinning control of complex networks, with emphasis on the potential applications in power electronic grid.

1. Introduction

Complex networks are everywhere today [1, 2]. Typical examples include the Internet, World Wide Web, power grid networks, communication networks, scientific citation networks, social networks, cellular neural networks, genetic regulatory networks, and so on [3, 4]. As we know now, the representative characteristics of complex networks are a large number of interconnected nodes and complex topological structure [5, 6, 7].

It is well known that the immune, vascular, endocrine, and nerve systems of our bodies regulate chemical reactions to keep equilibria in the face of ongoing attacks from disease and diet [4]. The above process is called homeostasis in biology. Here, the similar regulation processes of complex networks are called control [8, 9, 10].

Over the last ten years, numerous approaches have been proposed to control or intervene the dynamical behaviors of various real-world complex networks [11, 12, 13]. Since the real-world complex networks often have a large number of nodes, it is very difficult or even impossible to control all nodes to realize a given control goal [14, 15]. Therefore, we hope to control a portion of nodes to achieve the same control goal. In fact, the above idea of control of a portion of nodes is very effective in many real-world complex networks. Thus the above control technique of a portion of nodes is called pinning control. In 1997, Grigoriev, Cross, and Schuster introduced the pinning control of spatiotemporal chaos [5]. In 2004, Li, Wang, and Chen presented the pinning control of a complex dynamical network to its equilibrium [6]. In 2008, Zhou, Lu, and Lü studied the pinning adaptive synchronization of a general complex dynamical network [13]. In 2009, Wu, Zhou, and Chen further investigated the cluster synchronization of linearly coupled complex networks under pinning control [8]. Moreover, there are numerous results reported over the last few years. The intended purpose of this paper is to briefly review some recent advances in pinning control of complex networks. We hope to reflect the current state of the pinning control of complex networks.

This paper is then organized as follows. Section 2 introduces the basic idea of pinning control and its challenging questions. The adaptive pinning synchronization of complex dynamical networks is presented in Section 3. In Section 4, the pinning synchronization of undirected and directed complex dynamical networks is then discussed. Moreover, the global pinning controllability of complex networks is further investigated in Section 5. Finally, some potential applications are explored in Section 6.

2. Preliminary

This section will briefly review the basic idea of pinning control and its challenging questions.

As we know now, complex networks often have a large number of network nodes. The pinning control is proposed based on the following two main motivations: i) It is usually impossible to achieve a given control goal by controlling every node; ii) It is likely possible to reduce the number of controllers under the condition of the same control goal. Therefore, the basic idea of pinning control is to realize the same or even better control goal by employing a portion of network nodes. In general, there are two interesting basic questions in pinning control of complex networks: (i) How many nodes should a network with a given topological structure and coupling strength be pinned to realize the desired control goal? (ii) How much coupling strength should a network with a given topological structure and pinning nodes be applied to achieve the desired control goal? In 2008, Zhou, Lu, and Lü gave a positive answer to the above two fundamental questions for a special case [13]. In pinning control, the selection of network nodes is also an interesting question. It is well known that there are
two basic selective schemes: random scheme and specific scheme. The random scheme is to pin a portion of randomly selected network nodes. And the specific scheme is to pin a portion of network nodes by following a given rule, such as the degrees of network nodes and the betweenness of network nodes. A natural question is: "which kind of pinning schemes is much better?".

Fig. 1 shows the control effectiveness of the random scheme and specific scheme for a scale-free network [14]. Here, the scale-free network has 60 network nodes with the coupling strength $c = 8.246$ and $l = 15$. In specific scheme, one selects the top 15 largest-degree nodes and control gain is 29.7603. In random scheme, one randomly select 15 nodes and the control gain is 513.3709. It means that the specific scheme is much more effective than the random scheme for a scale-free network. However, it is not always true for all cases. Sometimes, the random scheme is much more effective than the specific scheme. It depends on the detailed network structure and node dynamics.

Moreover, the pinning control technique can be combined with some traditional or modern control methods, such as switching control, adaptive control, and robust control.

3. Adaptive Pinning Synchronization of Complex Dynamical Networks

To answer the above two fundamental questions in Section 2, this Section provides a simply approximate formula for estimating the detailed number of pinning nodes and the magnitude of the coupling strength for a given general complex dynamical network [13]. In this Section, all notations are described in [13].

Consider a general complex dynamical network consisting of $N$ identical nodes with linearly diffusive couplings [13], which is given by

$$\dot{x}_i = g(x_i, t) + \sum_{j=1}^{N} c_{ij} A x_j + v_i(x_1, \cdots, x_N), \quad (1)$$

where $1 \leq i \leq N$, $x_i = (x_{i1}, x_{i2}, \cdots, x_{in})^T \in \mathbb{R}^n$ is the state vector of the $i$th node, $g : \Omega \times \mathbb{R}^n \rightarrow \mathbb{R}^n$ is a nonlinear smooth vector field, $\dot{x} = g(x, t)$ is the node dynamics, $v_i \in \mathbb{R}^n$ are the control inputs satisfying $v_i(x_1, \cdots, x_N) = 0$. And $A \in \mathbb{R}^{n \times n}$ is the inner-coupling matrix and $C = (c_{ij})_{N \times N} \in \mathbb{R}^{N \times N}$ is the coupling configuration matrix. If there exists a link from node $i$ to node $j (j \neq i)$, then $c_{ij} > 0$ and $c_{ji}$ is the coupling strength; otherwise, $c_{ij} = 0$. Suppose that $C$ is an irreducible digraph matrix satisfying

$$\sum_{j=1}^{N} c_{ij} = 0.$$

Let $x = s(t; t_0, x_0) \in \mathbb{R}^n$ with $x_0 \in \mathbb{R}^n$, denoted as $s(t)$, be a solution of the node system $\dot{x} = g(x, t)$. Thus $S(t) = (s_{1}(t), s_{2}(t), \cdots, s_{n}(t))^T \in \mathbb{R}^{n \times N}$ is a synchronous solution of the general complex dynamical network (1). Here, $s(t)$ can be an equilibrium point, a periodic orbit, an aperiodic orbit, even a chaotic orbit in the phase space [3, 13].

**Proposition 1** [13] (P1) Assume that $||Dg(s)||_2$ is bounded, where $Dg(s)$ is the Jacobian of $g$ evaluated at $x = s$. That is, there exists a nonnegative constant $\alpha$ satisfying $||Dg(s)||_2 \leq \alpha$.

**Theorem 1** [13] Assume that P1 holds. If there exists a natural number $1 \leq l < N$ satisfying $\lambda_{l+1} < -\frac{\alpha}{4}$, then the synchronous solution $S(t)$ of the general complex network (1) is locally asymptotically stable under the pinning adaptive controllers

$$\left\{ \begin{array}{l}
    v_i = -p_i e_i, \quad p_i = q_i ||e_i||^2, \quad 1 \leq i \leq l \\
    v_i = 0, \quad (l+1) \leq i \leq N,
\end{array} \right. \quad (2)$$

where $q_i$ are positive constants for $1 \leq i \leq l$.

Rewrite the general complex network (1) as follows:

$$\dot{x}_i = G x_i + h(x_i, t) + \sum_{j=1}^{N} c_{ij} A x_j + v_i(x_1, \cdots, x_N), \quad (3)$$

where $1 \leq i \leq N$. 

- 399 -
Proposition 2 [13] (P2) Assume that \( h(x, t) \) is Lipschitz continuous. That is, there exists a Lipschitz constant \( \mu \) satisfying \( \|h(x, t) - h(s, t)\| \leq \mu \|e\| \) for \( 1 \leq i \leq N \).

Theorem 2 [13] Assume that P2 holds. If there exists a natural number \( 1 \leq l < N \) satisfying \( \lambda_{l+1} < \frac{\mu}{\delta + \tilde{\sigma}} \), then the synchronous solution \( S(t) \) of the general complex network (3) is globally asymptotically stable under the pinning adaptive controllers

\[
\begin{aligned}
\dot{x}_i(t) &= f(x_i(t), t) - \sigma_B N \sum_{j=1}^N l_{ij} x_j(t) + u_i(t), \\
\dot{v}_i(t) &= q_i \|e_i\|^2, & 1 \leq i \leq l, \\
\dot{v}_i(t) &= 0, & (l+1) \leq i \leq N,
\end{aligned}
\]

where \( q_i \) are positive constants for \( 1 \leq i \leq l \).

4. Pinning Synchronization of Complex Dynamical Networks

In this Section, the pinning synchronization of undirected and directed complex dynamical networks will be further investigated, where all notations are given in [7].

A general pinning controlled network is described by [7]

\[
\begin{aligned}
\dot{x}_i(t) &= f(x_i(t), t) + c \sum_{j=1}^N G_{ij} \Gamma x_j(t) + u_i, & i = 1, 2, \ldots, l, \\
\dot{x}_i(t) &= f(x_i(t), t) + e \sum_{j=1}^N G_{ij} \Gamma x_j(t), & i = l + 1, 2, \ldots, N.
\end{aligned}
\]

where

\[
u_i = -cd_i \Gamma (x_i - s(t)) \in R^p, i = 1, 2, \ldots, l,
\]

are \( n \)-dimensional linear feedback controllers with all the control gains \( d_i > 0 \).

Proposition 3 [7] P3 There exists a constant matrix \( K \) satisfying

\[
(x - y)^T f(x(t) - f(y(t)) \leq (x - y)^T K \Gamma (x - y),
\]

where \( \forall x, y \in R^n \).

Theorem 3 [7] Assume that P3 holds. The controlled undirected network (5) \( \gamma \) is globally synchronized if the following condition is satisfied:

\[
I_N \otimes (K \Gamma) + c(G - D) \otimes \Gamma < 0,
\]

where \( \otimes \) is the Kronecker product,

\[
D = \text{diag}(d_1, \ldots, d_l, 0, \ldots, 0),
\]

and \( I_N \) is the \( N \)-dimensional identity matrix.

Theorem 4 [7] Assume that the condition (7) holds and \( \Gamma \) is a positive definite matrix. Then, the adaptively controlled undirected network (9) is globally synchronized for a small constant \( \alpha > 0 \).

5. Global Pinning Controllability of Complex Networks

In this Section, the global pinning controllability of complex dynamical networks will be further explored, where all notations are given in [11].

Proposition 4 [11] (P4) If the feedback gain matrix \( K \), the inner linking matrix \( B \), and the coupling strength \( \sigma \) are chosen such that for every \( t \geq t_0 \) and for every \( y_1, \ldots, y_N \in R^n \)

\[
\lambda_i(y, t) < -\mu, \quad i = 1, \ldots, nN,
\]

where \( y = [y_1^T, \ldots, y_N^T]^T, \mu > 0, |\lambda_i(y, t)| \leq 0 \) are the eigenvalues of the matrix \( H(y, t) \) defined by

\[
H(y, t) = \mathcal{D}(y, t) - 2(\sigma L \otimes \text{sym} Q N + P \otimes \text{sym} Q K)
\]

with \( Q \) positive definite symmetric matrix in \( R^{n \times n} \), and

\[
\mathcal{D}(y, t) = 2 \text{diag} [\text{sym} F_{\alpha(t)}(y_1), \ldots, y_N] - \text{sym} F_{\alpha(t)}(y_1)
\]

Then, the dynamical system

\[
\dot{e}(t) = \mathcal{F}(e(t), t)e(t) - (\sigma L \otimes B + P \otimes K)e(t),
\]

where

\[
\mathcal{F}(e(t), t) = \text{diag} \{F_{\alpha(t)}(y_1), \ldots, F_{\alpha(t)}(y_N) - e(t)\},
\]

is globally exponentially stable about the origin, implying that the network

\[
\dot{x}_i(t) = f(x_i(t)) - \sigma B \sum_{j=1}^N l_{ij} x_j(t) + u_i(t)
\]

is globally pinning-controllable.

Corollary 1 [11] If, for some \( Q \) positive definite symmetric matrix in \( R^{n \times n} \), condition sym \( Q K = \kappa \) \( \text{sym} QB \) is satisfied, \( \text{sym} QB \) is a positive definite matrix and

\[
\lambda_{\min}(\sigma L + \kappa P) |\lambda_{\min}(\text{sym} QB)| > \alpha |Q|,
\]

where the positive constant \( \alpha \) satisfies \( |F_{\alpha(j)}| \leq \alpha \), then

\[
\dot{x}_i(t) = f(x_i(t)) - \sigma B \sum_{j=1}^N l_{ij} x_j(t) + u_i(t)
\]

is globally pinning-controllable.

Corollary 2 [11] If for some \( Q \) positive definite symmetric matrix in \( R^{n \times n} \), condition sym \( Q K = \kappa \) \( \text{sym} QB \) is satisfied and \( \text{sym} QB \) is a positive definite matrix, and the feedback gain \( \kappa \) satisfies

\[
\frac{\sigma \alpha}{\sigma(\alpha + \kappa)} > \frac{1}{\lambda_{\min}(\text{sym} QB)}
\]

then

\[
\dot{x}_i(t) = f(x_i(t)) - \sigma B \sum_{j=1}^N l_{ij} x_j(t) + u_i(t)
\]

is globally pinning-controllable.
\[ \dot{x}_i(t) = f(x_i(t), t) + c(t) \sum_{j=1}^{N} G_{ij} \Gamma x_j(t) - c(t) d_i \Gamma (x_i(t) - s(t)), i = 1, 2, \ldots, l, \]
\[ \dot{x}_i(t) = f(x_i(t), t) + c(t) \sum_{j=1}^{N} G_{ij} \Gamma x_j(t), i = l + 1, 2, \ldots, N, \]
\[ \dot{c}(t) = \alpha \sum_{j=1}^{N} (x_j(t) - s(t))^T \Gamma(x_j(t) - s(t)). \] (9)

6. Concluding remarks

This paper has briefly reviewed some recent advances in the pinning control of complex dynamical networks. It is certain that the pinning control will have a good prospect of application. For example, the operator of an electric power grid hopes to find an effective network model with pinning control design that will help form predictions of supply and demand to keep the stability of the whole power network [4], which requires the combined expertise of statisticians, economists, and power engineers. Moreover, the cost and effects of pinning control for network performance should be further investigated in the near future.

Acknowledgments

This work was supported by the ARC Future Fellowships under Grant FT0992226, the National Natural Science Foundation of China under Grants 60821091 and 60772158, the National Basic Research (973) Program of China under Grant 2007CB310805, the Important Direction Project of Knowledge Innovation Program of Chinese Academy of Sciences under Grant KJCX3-SYW-S01, and the Support of K. C. Wong Education Foundation of Hong Kong.

References


A Complex Network Perspective to Volatility in Stock Markets

Xiao Fan Liu and Chi K. Tse

Department of Electronic and Information Engineering, Hong Kong Polytechnic University, Hong Kong
URL: http://chaos.eie.polyu.edu.hk

Abstract—In this paper we examine the interaction of stock markets of different countries by constructing networks that connect 32 selected stock market indices from different countries. In the network being constructed, the nodes are the stock market indices and the edges are connections between the indices. Each edge has an edge weight equal to the cross-correlation between the pair of connecting indices over a window of \( w \) days. We consider the period from 7 March 2005 to 23 April 2009, i.e., 1078 days with \( w < 1078 \). In this period, networks are constructed for all \( w \)-day windows at 1-day intervals. By examining the variation of the network parameters as time elapses, we show that the dynamics of network connectivity is related to the fluctuation of the stock markets. Specifically, a form of network synchronization is found to be correlated with the volatility of the stock markets. Our study thus reveals that the stock markets in different countries generally behave in a synchronous manner when the markets experience fluctuation.

Keywords—Complex network, stock market, network dynamics, market volatility.

1. Introduction

The total domestic market capitalization of world equity markets has exceeded US$ 60 trillion in 2007 but dropped 46.5% in 2008 [1]. The phenomenon seemed to be global as the 2008 financial crisis swept almost every country [2]. It is thus clear that stock markets in different countries do not operate independently, and their interactions have a significant role to play in shaping the overall world stock market performance. The co-movement of world exchange indices has been studied since the 1970s [3]. Prior work uses variations of ARCH (autoregressive conditional heteroskedasticity) models [4, 5, 6] to study the correlations between stock market indices. Directed acyclic graphs are used to represent the structure of interdependence in international stock markets [7]. It has been found that a relationship exists between the structure of international stock markets and the market volatility [4, 8]. However, in much of the previous research, only a small number of stock markets from developed countries have been studied, resulting in somewhat biased conclusions on relationships between individual markets’ volatility and their correlations to the peer markets. Also, the previous approaches have over-simplified the structure of international stock markets.

In this paper, we construct a network of stock markets of 32 member countries of the World Federation of Exchanges.¹ The network nodes are the representative indices of the 32 stock markets.² Our study considers the daily closing value of each index during the 1078 working day period from 7 March 2005 to 23 April 2009. In case a stock market is closed on a working day, the day’s closing value inherits from the last available working day. In this 1078-day period, the stock markets network is constructed for all \( w \)-day windows at 1-day intervals. In the network, each pair of nodes are connected by an edge, with weight equal to the Pearson’s correlation coefficient between the two adjacent indices over a window of \( w \) days. It is obvious that the weight of edges evolve chronologically as the window slides in forward time. In this paper we will examine the network dynamics based on the variation of edge weights as time elapses. For each \( w \)-day window, we also calculate the properties of each stock market index, including its return, mean value and volatility in the window period. Our study focuses on investigating the relationships between network dynamics and financial properties of the indices under different choices of window size.

We begin with the network construction procedure in Section 2. Then, we examine the network properties and introduce the definition of network dynamics in Section 3. We will examine the dynamics of the stock market indices in Section 4 and show the relationship between network dynamics and stock markets’ financial dynamics in Section 5. Finally the discussion of our results will be presented in Section 6.

2. Network construction

In the 1078 working days from 7 March 2005 to 23 April 2009, the network is constructed for each of the \( w \)-day windows at 1-day intervals. Hence, the entire period is divided into \( M \) windows: \( W_1, W_2, \ldots, W_M \), where \( M = 1079 - w \). Let \( P_i(m) \) be the series of closing values of stock index \( i \) in the \( m \)th window. We consider the node of stock index \( i \) in the \( m \)th window. In the network construction procedure, each pair of network nodes are connected by an edge, with the edge weight equals to the Pearson’s correlation between the pair of adjacent indices [9]. Specifically, the edge weight \( \rho_{i,j}(m) \) between node \( P_i(m) \) and \( P_j(m) \) in the

¹They include Brazil, Mexico, Argentina, the USA and Canada from the Americas; the Netherlands, Austria, Belgium, France, Germany, the UK, Ireland, Spain, Denmark, Sweden, Portugal, Italy, Switzerland and Norway from Europe; Australia, India (both National Stock Exchange of India and Bombay Stock Exchange), Hong Kong, Indonesia, Malaysia, New Zealand, Japan, South Korea, China, Singapore from Asia/Pacific Region; Egypt and Israel from Africa and Middle East.
²They are Bovespa (Brazil), IPC (Mexico), MerVal (Argentina), S&P 500 (USA), S&P/TSX Composite (Canada), AEX (Netherlands), ATX (Austria), BEL-20 (Belgium), CAC 40 (France), DAX (Germany), FTSE 100 (United Kingdom), ISEQ20 (Ireland), Madrid General (Spain), OMX Copenhagen 20 (Denmark), OMX Stockholm 30 (Sweden), PSI 20 (Portugal), S&P Mib (Italy), Swiss Market (Switzerland), Total Share (Norway), All Ordinaries (Australia), BSE 30 (India), Hang Seng (Hong Kong), Jakarta Composite (Indonesia), KLSE Composite (Malaysia), NZSE 50 (New Zealand), Nikkei 225 (Japan), S&P CNX NIFTY (India), Seoul Composite (South Korea), Shanghai Composite (China), Strait Times (Singapore), CASE 30 (Egypt) and TA-100 (Israel). All data are retrieved from Yahoo! Finance.
Figure 1: A network for world stock market indices constructed from a 20-day window from 12 July 2008 to 2 August 2008. Edges with weights below 0.95 are excluded (unconnected). Yellow nodes represent stock market indices from the Asia Pacific region; red for the Americas; blue for Europe; and black for the Africa/Middle East region.

The edge weight distribution of the network constructed for the 20-day window of 17 Feb 2009 to 16 Mar 2009. \( p(\rho) \) is the probability of an edge weight falling in a 0.1 interval. This edge weight distribution resembles a normal distribution with mean 0.69 and standard deviation 0.21.

3. Network properties and dynamics

With all edges included, the number of nodes and edges in each of the constructed network are 32 and 496, respectively. We may now describe the connectivity and structure of the networks in terms of the distributions of the edge weights. Fig. 2 shows the edge weight distribution of the network for a particular window. The following two properties are particularly useful in characterizing the dynamics of the networks.

Definition 1: The node strength \( s_i(m) \) of node \( i \) in the \( m \)-th window is the average of the weights of all the edges connected to node \( i \), i.e.,

\[
s_i(m) = \frac{1}{31} \sum_{j=1,j \neq i}^{32} \rho_{ij}(m).
\]

4. Stock market properties and dynamics

By constructing the network of stock market indices for all the \( w \)-day windows, we can study how the individual stock markets interact. Here, we are interested to know whether the network
properties are related to any financial phenomena in the same windows. In particular, we calculate the return, average value and volatility of individual indices.

Let \( p_i(t) \) be the closing value of index \( i \) on day \( t \). The window return \( r_i(m) \) of index \( i \) in the \( m \)th \( w \)-day window \( W_M \), starting from \( t_m \) to \( t_{m+w-1} \), is given by

\[
r_i(m) = \frac{p_i(t_{m+w-1}) - p_i(t_m)}{p_i(t_m)} \times 100%.
\]

Likewise, the average closing value \( \mu_i(m) \) of index \( i \) in the \( m \)th \( w \)-day window \( W_M \) is given by

\[
\mu_i(m) = \frac{\sum_{t=t_m}^{t_{m+w-1}} p_i(t)}{w}.
\]

In stock markets, the stock prices and market indices change whenever stocks are traded. Moreover, there are many factors which could influence stock prices and hence make the fluctuation irregular. Such fluctuation, usually known as volatility, can be measured by calculating the standard deviation of the stock price or market index over a period of time. Specifically, the volatility \( \sigma_i(m) \) of index \( i \) in the \( m \)th \( w \)-day window \( W_M \) is given by

\[
\sigma_i(m) = \sqrt{\frac{\sum_{t=t_m}^{t_{m+w-1}} (p_i(t) - \mu_i(m))^2}{w-1}}.
\]

Here, we take the MSCI AC World Index (World Index) as the stock market index of the world financial system. Dynamics of the window return, average value and volatility of this World Index is an indication of how the world financial system behaves. Fig. 5 shows the return, average value and volatility of Hang Seng Index and World Index versus time, based on 20-day windows.

5. Application of networks: Connecting networks and stock markets

The key challenge for applying the study of complex networks in real-life applications is whether the properties found in networks have any corresponding physical meanings that would shed light on how the actual system behaves [10]. In this section we will explore the connection between the network dynamics (complex networks) and stock market dynamics (real systems).

First, we compare the network synchronization with dynamics of the World Index, which is used to characterize the world financial system. Specifically, given a \( w \)-day window, the correlations among series of network synchronization, index window return, average and volatility of length \( M = 1079 - w \) are calculated. Again we adopt the Pearson’s correlation. For example, the correlation coefficient \( \rho_{s,r} \) between network synchronization \( s \) and World Index return \( r \) is given by

\[
\rho_{s,r} = \frac{\langle sr \rangle - \langle s \rangle \langle r \rangle}{\sqrt{\langle s^2 \rangle - \langle s \rangle^2} \sqrt{\langle r^2 \rangle - \langle r \rangle^2}}
\]

where \( \langle \cdots \rangle \) denotes the expected value. We use different window sizes, i.e., \( w = 10, 20, 40, 60 \) and 120 days, to examine the correlation between the network and stock market dynamics in different time scales. The results of the calculation are shown in Table 1. We see from Table 1 that regardless of the choice of window size, the network synchronization and World Index volatility is strongly correlated with correlation coefficients around 0.6,
Table 1: Pearson’s correlation coefficients between each pair of dynamics of network synchronization $s$, World Index window return $r$, average $\mu$ and volatility $\sigma$.

<table>
<thead>
<tr>
<th>$w$</th>
<th>$\rho_{s,r}$</th>
<th>$\rho_{s,\mu}$</th>
<th>$\rho_{s,\sigma}$</th>
<th>$\rho_{r,\mu}$</th>
<th>$\rho_{r,\sigma}$</th>
<th>$\rho_{\mu,\sigma}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>-0.18</td>
<td>-0.10</td>
<td>0.65</td>
<td>0.07</td>
<td>-0.49</td>
<td>-0.21</td>
</tr>
<tr>
<td>20</td>
<td>-0.28</td>
<td>-0.13</td>
<td>0.65</td>
<td>0.12</td>
<td>-0.63</td>
<td>-0.21</td>
</tr>
<tr>
<td>40</td>
<td>-0.33</td>
<td>-0.10</td>
<td>0.65</td>
<td>0.24</td>
<td>-0.75</td>
<td>-0.20</td>
</tr>
<tr>
<td>60</td>
<td>-0.34</td>
<td>-0.05</td>
<td>0.61</td>
<td>0.31</td>
<td>-0.83</td>
<td>-0.19</td>
</tr>
<tr>
<td>120</td>
<td>-0.29</td>
<td>-0.38</td>
<td>0.55</td>
<td>0.38</td>
<td>-0.88</td>
<td>-0.33</td>
</tr>
</tbody>
</table>

Network Synchronization $s$

Window Volatility $\sigma$

Window Average $\mu$

Window Return $r$

-0.65 -0.13 -0.28 -0.21 -0.63 0.12

Figure 6: Graphical representation of relationships among network synchronization $s$, World Index window return $r$, average $\mu$ and volatility $\sigma$, with window size $w = 20$ days.

Table 2: Pearson’s correlation coefficients between each pair of node strength $s$, index window return $r$, average $\mu$ and volatility $\sigma$. The table is sorted by Pearson’s correlation coefficient $\rho_{s,\sigma}$ between node strength $s$ and index window volatility $\sigma$.

<table>
<thead>
<tr>
<th>Index</th>
<th>Country</th>
<th>$\rho_{s, r}$</th>
<th>$\rho_{s, \mu}$</th>
<th>$\rho_{s, \sigma}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALX</td>
<td>Brazil</td>
<td>-0.24</td>
<td>-0.10</td>
<td>0.06</td>
</tr>
<tr>
<td>All Ordinaries</td>
<td>Australia</td>
<td>-0.16</td>
<td>0.09</td>
<td>0.59</td>
</tr>
<tr>
<td>FTSE 100</td>
<td>UK</td>
<td>-0.25</td>
<td>-0.14</td>
<td>0.57</td>
</tr>
<tr>
<td>CAC 40</td>
<td>France</td>
<td>0.26</td>
<td>-0.16</td>
<td>0.57</td>
</tr>
<tr>
<td>DAX</td>
<td>Germany</td>
<td>0.24</td>
<td>0.00</td>
<td>0.56</td>
</tr>
<tr>
<td>S&amp;P Mib</td>
<td>Italy</td>
<td>-0.14</td>
<td>-0.25</td>
<td>0.55</td>
</tr>
<tr>
<td>TA-100</td>
<td>Israel</td>
<td>0.03</td>
<td>0.17</td>
<td>0.55</td>
</tr>
<tr>
<td>AEX</td>
<td>Netherlands</td>
<td>-0.24</td>
<td>-0.18</td>
<td>0.54</td>
</tr>
<tr>
<td>OMX Stockholm 30</td>
<td>Sweden</td>
<td>0.22</td>
<td>-0.11</td>
<td>0.53</td>
</tr>
<tr>
<td>Swiss Market</td>
<td>Switzerland</td>
<td>0.11</td>
<td>-0.08</td>
<td>0.53</td>
</tr>
<tr>
<td>Hang Seng</td>
<td>Hong Kong</td>
<td>0.13</td>
<td>0.10</td>
<td>0.52</td>
</tr>
<tr>
<td>OMX Copenhagen 20</td>
<td>Denmark</td>
<td>0.26</td>
<td>0.00</td>
<td>0.52</td>
</tr>
<tr>
<td>S&amp;P 500</td>
<td>USA</td>
<td>0.16</td>
<td>-0.08</td>
<td>0.52</td>
</tr>
<tr>
<td>BEL 20</td>
<td>Belgium</td>
<td>0.18</td>
<td>-0.16</td>
<td>0.51</td>
</tr>
<tr>
<td>Nikkei 225</td>
<td>Japan</td>
<td>0.11</td>
<td>-0.26</td>
<td>0.49</td>
</tr>
<tr>
<td>Total Share</td>
<td>Norway</td>
<td>-0.25</td>
<td>-0.06</td>
<td>0.49</td>
</tr>
<tr>
<td>NZSE 50</td>
<td>New Zealand</td>
<td>0.11</td>
<td>-0.04</td>
<td>0.48</td>
</tr>
<tr>
<td>Bovespa</td>
<td>Brazil</td>
<td>0.01</td>
<td>0.17</td>
<td>0.48</td>
</tr>
<tr>
<td>Straits Times</td>
<td>Singapore</td>
<td>0.08</td>
<td>-0.02</td>
<td>0.46</td>
</tr>
<tr>
<td>Madrid General</td>
<td>Spain</td>
<td>0.13</td>
<td>-0.24</td>
<td>0.45</td>
</tr>
<tr>
<td>IPC</td>
<td>Mexico</td>
<td>-0.02</td>
<td>-0.08</td>
<td>0.45</td>
</tr>
<tr>
<td>MerVal</td>
<td>Argentina</td>
<td>0.04</td>
<td>-0.06</td>
<td>0.44</td>
</tr>
<tr>
<td>KLSE Composite</td>
<td>Malaysia</td>
<td>-0.05</td>
<td>0.16</td>
<td>0.44</td>
</tr>
<tr>
<td>BSE 30</td>
<td>India (BSE)</td>
<td>-0.22</td>
<td>-0.14</td>
<td>0.43</td>
</tr>
<tr>
<td>S&amp;P TSX Composite</td>
<td>Canada</td>
<td>-0.11</td>
<td>-0.16</td>
<td>0.42</td>
</tr>
<tr>
<td>Seoul Composite</td>
<td>South Korea</td>
<td>-0.05</td>
<td>0.03</td>
<td>0.41</td>
</tr>
<tr>
<td>S&amp;P CNX NIFTY</td>
<td>India (NSE)</td>
<td>-0.20</td>
<td>-0.14</td>
<td>0.40</td>
</tr>
<tr>
<td>PSI 20</td>
<td>Portugal</td>
<td>-0.15</td>
<td>-0.07</td>
<td>0.40</td>
</tr>
<tr>
<td>Jakarta Composite</td>
<td>Indonesia</td>
<td>0.00</td>
<td>0.04</td>
<td>0.34</td>
</tr>
<tr>
<td>ISEQ20</td>
<td>Ireland</td>
<td>-0.06</td>
<td>-0.22</td>
<td>0.28</td>
</tr>
<tr>
<td>EGX30/CA.</td>
<td>Egypt</td>
<td>0.06</td>
<td>-0.06</td>
<td>0.13</td>
</tr>
<tr>
<td>Shanghai Composite</td>
<td>China</td>
<td>0.02</td>
<td>0.03</td>
<td>0.04</td>
</tr>
</tbody>
</table>

strongly correlated to the volatility of stock markets, with the exclusion of stock markets from developing countries. Thus, we may conclude that individual markets generally react in a synchronous fashion when the markets experience fluctuation.

6. Conclusion

Networks have been constructed for 32 important stock markets based on connecting each pair of stock markets according to the correlation between their representative indices. We have studied the dynamics of the networks during the period from 7 March 2005 to 23 April 2009. In order to make the study useful for application, we have established the relationship between network dynamics and stock market dynamics. By comparing the network synchronization and node strength to the indices’ window return, average and volatility, we discover that the network synchronization and most of the node strength dynamics are strongly correlated to the volatility of stock markets, with the exception of stock markets from developing countries. Thus, we may conclude that individual markets generally react in a synchronous fashion when the markets experience fluctuation.

References

**Property of the chaotic propagating pulse wave in a ring of coupled bistable oscillators**

Kyohei Kamiyama‡, Tetsuro Endo‡, Kuniyasu Shimizu‡ and Hiroyuki Kamata‡

‡Dept. of Electronics and Bioinformatics, Meiji University
1-1-1 Higashi-mita, Tama-ku, Kawasaki, Kanagawa 214-8571 Japan
‡Dept. of Electrical, Electronics and Computer Engineering, Chiba Institute of Technology
2-17-1 Tsudanuma, Narashino, Chiba 275-0016 Japan

Email: ce91023@isc.meiji.ac.jp, endoh@isc.meiji.ac.jp, kuniyasu.shimizu@it-chiba.ac.jp, kamata@isc.meiji.ac.jp

**Abstract**—In this paper, some properties of the chaotic propagating pulse wave in a ring of six coupled bistable oscillators are investigated. When coupling factor \(\alpha\) becomes large beyond a certain critical point, the standing pulse wave converts to a propagating pulse wave. Further, as \(\alpha\) is increased, the propagating pulse wave behaves chaotically. We find some interesting properties of chaotic propagating pulse wave such as random change of propagation direction, stepwise change of pulse position wrt time, and probability density of the time-length and distance, etc.

1. Introduction

The pulse wave propagation phenomena in coupled oscillator systems are very popular in recent years [1]. We have investigated the pulse wave in a ring of coupled bistable oscillator systems in [2], [3]. Generally speaking, for small coupling factor, there is a standing pulse wave which stays in one place. When the coupling factor becomes large beyond a certain critical value, the standing pulse wave converts to the propagating pulse wave. Further, for larger coupling factor, the propagating pulse wave becomes chaotic. It changes its propagation direction at random. In this paper, we investigate the properties of the chaotic propagating pulse wave such as probability density of the time-length and distance, etc.

2. Chaotic propagating pulse wave in a ring of six coupled bistable oscillators

In our previous paper, we investigate transition mechanism from a standing pulse wave to a propagating pulse wave in terms of coupling factor \(\alpha\) in a ring of six coupled bistable oscillators [2][3].

The equation we investigate is as follows:

\[
\begin{align*}
    x_i &= y_i \\
    \dot{y}_i &= -\varepsilon(1 - \beta x_i^2 + x_i^4)y_i \\
    &\quad - (1 - \alpha)x_i + \alpha(x_{i-1} + 2x_i + x_{i+1}) \\
    &\quad , \quad i = 1, 2, \cdots, N, \quad x_0 = x_N, x_N = x_{N+1}
\end{align*}
\]

where \(N\) is number of oscillators. The \(x_i\) denotes the normalized output voltage of the \(i\)-th oscillator, \(y_i\) denotes its derivative. The parameter \(\varepsilon (> 0)\) shows the degree of nonlinearity. The parameter \(\alpha (0 \leq \alpha < 1)\) is a coupling factor; namely \(\alpha = 1\) means maximum coupling, and \(\alpha = 0\) means no coupling. The parameter \(\beta\) controls amplitude of oscillation. Each isolated oscillator has two steady-states, namely, no oscillation and periodic oscillation depending on the initial condition. In this paper, parameters \(\beta\) and \(\varepsilon\) are fixed as \(\beta = 3.18\) and \(\varepsilon = 0.36\).

It has been already clarified that the transition from the standing pulse wave to the propagating pulse wave is a bifurcation from the periodic solution to the almost periodic solution, and that the bifurcation originates in a complex combination of the pitchfork and the heteroclinic bifurcations [3]. When coupling factor \(\alpha\) is increased, it is noted that the propagation speed increases and beyond a certain critical value of \(\alpha\), the propagating wave become chaotic. The variation of Lyapunov exponents is presented in term of \(\alpha\) in Figure 1.

![Fig. 1 Transition of 12 Lyapunov exponents of a ring of six coupled bistable oscillators in terms of \(\alpha\) for \(\beta = 3.18\) and \(\varepsilon = 0.36\). red: LE1, blue: LE3. In region B, LE1 and LE2 overlap.](image)

Namely, in region A(0.08 \(\leq\) \(\alpha\) \(<\) 0.0905), where the standing pulse wave exists, \(\text{LE1} = 0\) and \(\text{LE2}\sim\text{LE12} < 0\). Therefore, this is a periodic solution. In region B (0.0905 \(\leq\) \(\alpha\) \(<\) 0.1118) where the non-chaotic propagating pulse wave exists, \(\text{LE1} = \text{LE2} = 0\) and \(\text{LE3}\sim\text{LE12} < 0\). Therefore, this is an almost periodic solution. In region C (0.1118 \(\leq\) \(\alpha\) \(<\) 0.1162) where the chaotically propagating pulse wave exists, \(\text{LE1} > 0, \text{LE2} = 0\) and \(\text{LE3}\sim\text{LE12} < 0\). Therefore, this is a chaotic attractor. In region D (0.1162 \(\leq\) \(\alpha\)) there is no oscillation.
Figures 2 demonstrates 3D representation of typical (a) standing pulse wave, (b) non-chaotic propagating pulse wave, and (c) chaotic propagating pulse wave. Note that the standing pulse wave stays in one position, propagating pulse wave propagates in one direction; namely propagating direction is unchanged, once it is determined. In contrast, the chaotic propagating pulse wave changes its direction occasionally in random manner.

Fig. 2 Three typical waves: (a) standing pulse wave for $\alpha = 0.08$, (b) non-chaotic propagating pulse wave for $\alpha = 0.10$ and (c) chaotic propagating pulse wave for $\alpha = 0.115$. The fixed parameters are $\epsilon = 0.36$ and $\beta = 3.18$. The absolute magnitude of $(i : \text{number of oscillators})$ is shown in colors.

Figures 3 (a) and (b) present the propagating distance measured by oscillator number in terms of time for (a) $\alpha$ chosen in non-chaotic regime, and for (b) three values of $\alpha$ all chosen in chaotic regime. It is recognized that the propagating direction do not change in non-chaotic regime, but it suddenly changes in random manner in chaotic regime. It seems that the absolute value of propagation speed ($= \text{magnitude of the slope}$) in chaotic regime is constant for fixed values of $\alpha$.

Figure 4 shows absolute value of propagation speed in terms of $\alpha$. It is recognized that the propagation speed in chaotic regime is a smooth extension of the non-chaotic regime. That is, the (absolute value of) propagation speed increases with the increase of $\alpha$. 
This probability density is calculated by using the kernel density estimation method with bandwidth equals to 1 and with Gaussian kernel [4]. Here, one-section time length denotes the time in which a chaotic pulse propagates to one direction. The probability of occurrence of small time length and that of large time length are both large, while, that of medium time length is small. Moreover, the maximum time length for small $\alpha$ (the time corresponding to peak a), is larger than that for large $\alpha$ (the time corresponding to peak a’ and a”). This means that the probability of propagating direction change is small for smaller values of $\alpha$ compared to larger values of $\alpha$. One of the characteristic features of the probability density in Fig.5 is its tooth-like structure. The time $\Delta t$ in Fig.3 (a) is equal to the time between two peaks in Fig.5. This is the time for a pulse to move one oscillator unit. Namely, a pulse stays in one oscillator for a long time and quickly moves to the next one.

Figures 6 (a), (b) and (c) denote the probability density of one-section length. Same as Fig.5 the probability density is large in both sides and it is small in the middle. In particular, comparing three peaks b, b’ and b”, it is recognized that for smaller $\alpha$, the distance of one-section is longer.

4. Conclusions

We investigate properties of chaotic propagating pulse wave in a ring of six coupled oscillator system. Namely, we calculate propagating distance in term of time, propagating speed in terms of coupling strength, and probability density of the pulse direction change phenomenon. In the future, we investigate the same characteristics for larger number of oscillator cases.

References

Fig. 5 Probability density of one-section time length of a chaotic propagating pulse in different values of $\alpha$: (a) $\alpha = 0.113$, (b) $\alpha = 0.114$, (c) $\alpha = 0.116$. The total time for (a), (b) and (c) is 800000 seconds. The fixed parameters are $\varepsilon = 0.36$ and $\beta = 3.18$.

Fig. 6 Probability density of the distance of one section of a chaotic propagating pulse in different values of $\alpha$: (a) $\alpha = 0.113$, (b) $\alpha = 0.114$, (c) $\alpha = 0.116$. The total time for (a), (b) and (c) is 800000 seconds. The fixed parameters are $\varepsilon = 0.36$ and $\beta = 3.18$. 
Self-organized behaviors in an adaptive network of movable oscillators

Takaaki Aoki† and Toshio Aoyagi‡

†Graduate School of Informatics, Kyoto University
Yoshida-honcho, Sakyo-ku, Kyoto, 606-8501 Japan
‡JST CREST
Yoshida-honcho, Sakyo-ku, Kyoto, 606-8501 Japan
Email: aoki@acs.i.kyoto-u.ac.jp

Abstract—We investigate self-organization of both connections and phases in a system of coupled movable phase oscillators, in which the self-movable units on two-dimensional plane have their internal states which is described by phase oscillator model. In this system, both the connections linking the oscillators and the phases of the oscillators evolve simultaneously. The oscillators are coupled with its neighbors and the movement of the units is governed by the collective dynamics of the oscillators. We found that this co-evolving dynamical system exhibits four types of distinct collective behaviors of oscillators, which can be characterized by the topological structure of the network: an aggregate, a repulsive, a reticulate, and a islands-structure states. Furthermore, we demonstrate that these topological structure in a population of the coupled oscillators can be adaptively reorganized in response to the change in the environment. This behavior is consistent with the experimental observation of slime mold. We believe that this simple model of co-evolving dynamical system allows us to understand and clarify the fundamental mechanism of self-organization of adaptive network in biological and ecological systems.

1. Introduction

Many real-world complex networks can be regarded as adaptive network, where the structure of network is changing dynamically depending on the activity on the network [1, 2]. In other words, the structure of network is self-organized based on their dynamics of the system. Such self-organization is thought to be closely related to the emergence of the functional topological structure of complex networks and to provide a clue for designing intelligent self-assembled and decentralized autonomous systems [3]. We here consider a class of adaptive network, that is a network of self-movable units on a two-dimensional plane [4, 5]. Each movable unit has an internal state and interacts with its neighboring units. The unit is moving on the plane depending on the interaction with the neighboring units, while the connections linking the neighbors is also changing by the movement of the units. This type of activity-dependent network can be seen in biological and ecological systems. Recently, Tanaka has derived a general description of chemotactic system of sustained oscillators, in which movable particles with oscillatory internal state are driven by the spatial gradient of the chemical concentration [6].

In this paper, we study an adaptive group formation in a population of the coupled movable phase oscillators. We consider a certain-sized unit that causes the excluded volume effect and investigate the group formation of the units which is represented by a two-dimensional network linking the neighboring units. We investigate the asymptotic states of the proposed dynamical systems and show that it exhibits four distinct types of collective behaviors, which can be characterized by the topological structure of the network. Furthermore, we introduce a density-dependent oscillatory dynamics of the units and demonstrate that the topological structure in the population of oscillators can be adaptively reorganized in response to the change of the environment.

2. Methods

First, let us consider a coupled phase oscillators with time-dependent coupling weight \(k_{ij}(t)[7, 8, 9]\) as

\[
\frac{d\phi_i}{dt} = \omega_i + \sum_j k_{ij}(t) \sin(\phi_i - \phi_j + \alpha).
\]  

The oscillatory internal state of the movable unit is characterized by a phase \(\phi_i\). \(\omega_i\) is its natural frequency. In the coupling term of the above equation, the parameter \(\alpha\) represents the transmission delay between units. The movable unit have a position \(r_j(t)\) on a two-dimensional plane and a excluded area over a radius \(r_0 = 0.5\). Then, the coupling weight \(k_{ij}(t)\) is determined by the distance between the movable units (\(\equiv |r_{ij}|\)) as,

\[
k_{ij}(t) = W(|r_{ij}(t)| - 2r_0), \quad W(r) = [1 - r/\Delta],
\]  

where \([x] = x\) if \(x \geq 0\) and \([x] = 0\) elsewhere. \(\Delta\) is the maximum distance of the interactive neighbors at which \(k_{ij} = 0\). When a unit touches another unit at the closest distance (\(= 2r_0\)), then \(k_{ij} = 1\).

Next, we introduce the dynamics of the position \(r_j(t)\) of the movable unit determined by the activity of the oscillators. We consider a phase-dependent force \(F(\phi_i - \phi_j)\)
among units. The function $F(\phi)$ is generally $2\pi$-periodic, and assumed that $F(\phi_i - \phi_j) = \sin(\phi_i - \phi_j + \beta)$, taking only the first Fourier mode into account. Then, in a high-viscosity situation, the dynamics of the position is described by,

$$
\frac{dr_{ij}(t)}{dt} = \epsilon \sum_j \hat{r}_{ij}(t) \left[ k_{ij}(t) \sin(\phi_i - \phi_j + \beta) + f^H(|r_{ij}(t)|) \right],
$$

(3)

where $\hat{r}_{ij}(t)$ is a unit vector directed from a unit $j$ to $i$. $\epsilon$ determines the time-scale of the dynamics of $r_i(t)$, which is assumed to be small. Thus, the movement of units is much slower than the activity of oscillators. The last term represents the excluded volume effect of the units, which is given by

$$
f^H(|r_{ij}(t)|) = \begin{cases} 
1 & (|r_{ij}(t)| < 2r_0) \\
0 & (|r_{ij}(t)| \geq 2r_0) 
\end{cases}.
$$

(4)

The framework of the equations (1) and (3) can be derived as the reduced equations of a general chemotactic system with oscillatory internal dynamics[6]. In this sense, the presented model is based on a general description of movable oscillators driven by a diffusive substance on the plane.

In this paper, we consider a simple situation which the units interact with the only near neighbors ($\Delta \sim 1$) and its movement is slow ($\epsilon \ll 1$). Thus, two parameters $\alpha, \beta$ determine the behaviors of coupled movable oscillators. In particular, $\beta$ controls the characteristics of phase-dependent force between the movable units, which significantly affects the collective behaviors of this system. If $\beta \sim -\frac{\pi}{2}$, then phase-dependent force $F(\phi_i - \phi_j) \sim -\cos(\phi_i - \phi_j)$, so that the force is attractive between the units with similar phases and repulsive between ones with the different phases. In contrast, when $\beta \sim \frac{\pi}{2}$, the phase-dependent force has an opposite characteristic, which is repulsive between the synchronized oscillators. Consequently, the parameter $\beta$ will change the collective behaviors of the movable oscillators. Thus, we investigate dependency of the asymptotic behaviors on the parameters.

3. Result

3.1. Emergence of the four type of collective behaviors

In this section, we show that the proposed model exhibits four types of distinct collective behaviors depending on the parameters $\alpha$ and $\beta$: an aggregate, a repulsive, a reticulate and an islands-structure states (Fig. 1). We investigated asymptotic behaviors of oscillators changing $\alpha$ and $\beta$ started from an initial state. At the initial state, the oscillators are regularly located on a two-dimensional plane, keeping a distance to the nearest ones within the maximum interactive range $\Delta$.

First, we explain an aggregate state, which looks like a colony of movable oscillators. The oscillators exhibit a

![Aggregate state](image1)

![Repulsive state](image2)

![Reticulate state](image3)

![Islands-structure state](image4)

Figure 1: Typical collective behavior of movable oscillators emerged from the co-evolving dynamics of the oscillators and the connections, depending on the parameters ($\alpha, \beta$): an aggregate, a repulsive, a reticulate, and an islands-structure states. Left graphs show a population of movable oscillators on the two-dimensional plane with colored circles, in which the color gradient depicts internal phase $\phi_i$. Right graphs show schematic illustrations of the organized network among the oscillators.
target pattern of phases. Toward the center of the target pattern, the movable oscillators gather in an aggregation. Then, the coupling connections among oscillators have a lattice-like topology. This state is mostly observed in a region $\beta \in (-\pi/2, 0)$ and $\alpha \in [0, -\pi/4)$, where the phase-dependent force is attractive between in-phase oscillators (Fig. 2).

In another parameter region as $\beta \in (0, \pi)$, the phase-dependent force is repulsive between in-phase oscillators. Then, the movable oscillators exhibit another type of collective behaviors, a repulsive state. The phases of oscillators show the target pattern as same as the aggregate state, but the individual oscillators move apart from the others due to the repulsive force. Consequently, the distance to the nearest oscillators nearly equals to the maximum interactive range, which means that the oscillators are almost uncoupled.

Next, a reticulate state is observed in a region $\beta \in (-\pi, -\pi/2)$. In this state, oscillators also show the target pattern and they moves along this phase wave toward the opposite direction of the target center. Then, the oscillators spread over the two-dimensional plane as in the repulsive state. In this state, however, the oscillators does not uncoupled and keep contacting with the nearest oscillators. It is caused by the attractive phase-dependent force between in-phase oscillators. Owing to these two requirements, a spreading and a contacting, the group of movable oscillators organize a reticulate network on the two-dimensional plane.

As the parameter $\alpha$ gets larger, an islands-structure state is observed. The phase pattern of oscillators has a number of spirals and then, the oscillators are separately moved along the phase gradient of the individual spirals. For this reason, the oscillators are divided into a number of small clusters, which look like islands on the plane.

### 3.2. Adaptive reorganization of collective behaviors in response to the change in the environment

Four types of collective behaviors are emerged from the co-evolving dynamics of movable oscillators in a self-organizing manner. In this section, we study the response of the collective behaviors to an external stimulus and demonstrate that the organized collective behaviors can be reorganized adapted to the external stimulus.

Now, we consider an effect of the environment on the oscillators with the aim of introducing an external stimulus to the model. Our model is based on chemotactic system, such as a slime mold, where the unit is sensitive to some external chemical concentration and changes their internal dynamics depending on the chemical concentration. For example, it is known that slime mold modifies its oscillators frequency depending on cAMP of the environment, which is emitted from the neighboring units [10]. This fact means that the oscillation frequency is density-dependent. Thus, we redefine the natural frequency of the movable oscillators by

$$
\omega_i = G \left( \frac{1}{6} \sum_j k_j(t) + A(r_i) \right),
$$

where $A(r_i)$ is the environmental chemical concentration at the position of the $i$-th oscillator.

To investigate the response to the external chemical concentration, we consider the following situation (Fig. 3(a)). A block is located on the two-dimensional plane and the environmental chemical concentration is greater or less around the block. The parameter $\alpha, \beta$ is set to be the aggregate state. When the chemical concentration is greater, that is, the frequency of the oscillator becomes higher around the block, then the group of movable oscillators is attracted to the block, and eventually the block is encircled by them. In contrast, if the frequency of the oscillator becomes lower around the block, then the group of the movable oscillators escapes from the block. This result indicates that such a simple mechanism of the frequency modulation by the external chemical concentration can produce the attractive and repellent behavior of the group of movable units.

Next, we consider another situation where the chemical concentration is globally modified on the two-dimensional plane. This global attractive or repellent field will affect on the organization of the movable oscillators. Figure 3(b) shows topological reformation of the group of movable oscillators in response to the global change in the environmental chemical concentration. The oscillators are initially in the aggregate state and this state is also stable.
even on the attractive field. When the global chemical concentration is lowered, the aggregation of the movable oscillators is destroyed and reorganized to be reticulate structure for spreading over the plane. After that, when the global chemical concentration is recovered to be attractive, then reticulate structure is reorganized to be aggregation of several colonies. Such an adaptive reformation of collective behavior is similar to the experimental observation of slime mold. It is noted that this simple dynamical system can produce the adaptive organization of the collective behavior like biological chemotactic systems, based on the co-evolution of the phases of oscillators and the connections between the movable units.

4. Conclusion

In this paper, we have studied the group formation in a population of the coupled movable phase oscillators, in which both the connections linking the oscillators and the phases of the oscillators evolve simultaneously. We found that this co-evolving dynamical system exhibits four types of distinct collective behaviors: the aggregate, the repulsive, the reticulate, and the islands-structure states. Furthermore, we demonstrate that the aggregate and reticulate states can be adaptively reorganized in response to the frequency modulation caused by the change of the environmental chemical concentration.

Acknowledgments

We especially thank D. Tanaka for fruitful discussions. This work was supported by KAKENHI(19GS0208, 2112002) from MEXT and Grant-in-Aid for JSPS Fellows.

References

On Auto-Correlation Values of de Bruijn Sequences

Hiroshi Fujisaki and Yuhki Nabeshima

Graduate School of Natural Science and Technology
Kanazawa University
Kakuma-machi, Kanazawa, Ishikawa, 920-1192 Japan
Email: fujisaki@t.kanazawa-u.ac.jp

Abstract—We give a novel lower bound of the minimum values of the normalized auto-correlation functions for de Bruijn sequences of length \( N = 2^n (n \geq 3) \). The lower bound is tight in the sense that the equality holds for \( n = 3 \) and \( n = 4 \). For \( 3 \leq n \leq 6 \), we experimentally characterize the worst and the second-worst sequences in all de Bruijn sequences in terms of the normalized auto-correlation function.

1. Introduction

Correlational properties of pseudo-random sequences play important roles in the systems in which the sequences are used, such as cryptography and digital communication systems. Pseudo-random sequences appropriate for application in such systems are supposed to fulfill the property of full-length or maximal-period from the view point of complexity.

To generate full-length sequences, a LFSR (linear feedback shift register) is commonly used. On the other hand, in view of randomness in chaotic dynamics of one-dimensional ergodic transformations, sequences based on discretized Bernoulli transformations were proposed in [1] and [2]. The latter sequences have a great advantage in terms of their family size. For instance, for binary sequences of length \( 2^n \), while the total number of the former sequences is much less than \( 2^n/n \), the total number of the celebrated de Bruijn sequences is known to be \( 2^{2^n-1-n} \).

In [3], we generally defined discretized Markov transformations and found an algorithm to give the total number of full-length sequences based on discretized Markov transformations. The discretized Markov transformations, which can be regarded as examples of ultradiscrete dynamical systems [4], are permutations of subintervals in Markov partitions determined from the transformations. From this viewpoint, de Bruijn sequences are merely special examples of full-length sequences in the discretized Markov transformations. In fact, they are full-length sequences based on a subclass of the discretized dyadic transformations.

In previous research [5], we defined the piecewise-monotone-increasing Markov transformations and gave the bounded monotone truth-table algorithm for generating all full-length sequences which are based on the discretized piecewise-monotone-increasing Markov transformations. The algorithm proposed in [5] is applicable to generation of all de Bruijn sequences. We stress here that only a few algorithms are known for generating all de Bruijn sequences, while a number of results have contributed to generations of a single sequence or a small fraction of the sequences [6]–[7].

In light of the previous results [5], we can freely construct all full-length sequences, including all de Bruijn sequences, which are based on the discretized piecewise-monotone-increasing Markov transformations. Unfortunately, however, we know little of the statistical properties of full-length sequences which are based on the discretized transformations.

With the help of linearity, the algebraic structure of LFSR enables us to evaluate the correlational properties of full-length sequences based on the LFSR. On the other hand, because of the nature of nonlinearity, it is intractable to characterize the correlational properties of full-length sequences based on the discretized piecewise-monotone-increasing Markov transformations. Even for de Bruijn sequences, only bounds of the maximum values of the normalized auto-correlation functions are known [8]. For modified de Bruijn sequences of length \( 2^n - n (4 \leq n \leq 6) \), the auto-correlation values are experimentally examined in [9].

In this report, we study statistical properties of full-length sequences which are based on the discretized piecewise-monotone-increasing Markov transformations. The problem of finding a family of good sequences in terms of the correlational properties is not only mathematically challenging but also practically important as pointed out in the beginning of the Introduction. As the first step, we focus on a fundamental example of such full-length sequences, namely the de Bruijn sequences, and we study the bounds of the normalized auto-correlation values of the de Bruijn sequences.

This report is composed of five sections. In Sect. 2, we briefly review the previous results on the values of the normalized auto-correlation functions for de Bruijn sequences. In Sect. 3, we give a novel lower bound of the minimum values of the normalized auto-correlation functions for de Bruijn sequences of length \( N = 2^n (n \geq 3) \). In Sect. 4, we show that the lower bound is tight in the sense that the equality holds for \( n = 3 \) and \( n = 4 \). The report concludes with the summary in Sect. 5.
2. Preliminaries

The correlation functions for sequences are measures of the similarity, or relatedness, between two sequences. Mathematically they are defined as follows.

**Definition 1** The cross-correlation function of time delay $\ell$ for the sequences $X = (X_i)_{i=0}^{N-1}$ and $Y = (Y_i)_{i=0}^{N-1}$ over $[-1, 1]$, is defined by

$$R_N(\ell; X, Y) = \sum_{i=0}^{N-1} X_i Y_{i+\ell \text{ (mod } N)},$$

where $\ell = 0, 1, \cdots, N - 1$ and, for integers $a$ and $b (\geq 1)$, $a \ (\text{mod } b)$ denotes the least residue of $a$ to modulus $b$. The normalized cross-correlation function of time delay $\ell$ for the sequences $X$ and $Y$ is defined by

$$r_N(\ell; X, Y) = \frac{1}{N} \sum_{i=0}^{N-1} X_i Y_{i+\ell \text{ (mod } N)}.$$

If $X = Y$, we call $R_N(\ell; X, X)$ and $r_N(\ell; X, X)$ the autocorrelation function and the normalized autocorrelation function, and simply denote them by $R_N(\ell; X)$ and $r_N(\ell; X)$, respectively.

By the definition, it is easy to verify the following properties:

**Remark 1** For any $X = (X_i)_{i=0}^{N-1}$ over $[-1, 1]$, the normalized autocorrelation function $r_N(\ell; X, X)$ satisfies

$$r_N(\ell; X) = r_N(N - \ell; X)$$

and

$$r_N(0; X) = 1. \quad (2)$$

In this research, the focus is on the de Bruijn sequences, which are typical examples of full-length sequences in the discretized Markov transformations as stated in the Introduction. The de Bruijn sequences can be defined in terms of the discretized Markov transformations [3]. However, we here simply define them irrespective of the discretized Markov transformations as follows.

A (binary) cycle of length $k$ is a sequence of $k$ digits $a_1a_2\cdots a_k$ taken in a circular order. In the cycle $a_1a_2\cdots a_k$, $a_1$ follows $a_k$, and $a_2\cdots a_1a_1\cdots a_{k-1}$ are all the same cycle as $a_1a_2\cdots a_k$.

A (binary) complete cycle of length $2^n$ is a cycle of binary $2^n$-words, such that the $2^n$ possible ordered sets of binary $n$-words of that cycle are all different. Any binary $n$-word occurs exactly once in the complete cycle. A complete cycle of length $2^n$ has normality of order $n$.

**Example 1** We give examples of complete cycles of length $2^n$.

- $n = 1$, 01,
- $n = 2$, 0011, 001101,
- $n = 3$, 00010111, 00011101.

Because of the following theorem, the complete cycles are sometimes called de Bruijn sequences.

**Theorem 1 (de Bruijn [10], Flye Sainte-Marie [11])**

For each positive integer $n$, there are exactly $2^{2^n-n}$ complete cycles of length $2^n$.

In this study, we are concerned with correlational properties of the de Bruijn sequences. As we see above, a de Bruijn sequence is usually defined as a sequence over $[0, 1]$ while the correlation functions are defined for a sequence over $[-1, 1]$. Throughout this report, when we compute the values of the normalized auto-correlation functions $r_N(\ell; X)$ for a de Bruijn sequence $X$, we regard 0 in the de Bruijn sequences as $-1$. In other words, we transform a de Bruijn sequence $X$ of length $N$ over $[0, 1]$ to a sequence of length $N$ over $[-1, 1]$ by one-to-one correspondence between 0 and $-1$.

The following properties for the normalized auto-correlation functions of a de Bruijn sequences are well known. We start with this result. For a proof, consult [6] for example.

**Theorem 2** Let $N = 2^n (n \geq 1)$. For any $n (\geq 1)$, the normalized auto-correlation functions of de Bruijn sequences satisfy

$$\sum_{\ell=0}^{N-1} r_N(\ell; X) = 0, \quad (3)$$

and

$$r_N(\ell; X) = r_N(N - \ell; X), \quad 1 \leq \ell \leq n - 1. \quad (4)$$

3. Bounds of Auto-Correlation Values of de Bruijn Sequences

We set $N = 2^n (n \geq 1)$. By (2) in Remark 1, for any $n$, if $X$ is a de Bruijn sequence of length $2^n$, we always have

$$\max_{0 \leq \ell \leq N-1} r_N(\ell; X) = 1.$$

On the other hand, except for the case $\ell = 0$, we obtain

**Theorem 3 ([8])** If $X$ is a de Bruijn sequence of length $2^n$, then

$$0 \leq \max_{1 \leq \ell \leq N-1} r_N(\ell; X) \leq 1 - \frac{4}{2^n} \left[ \frac{2^n}{2^n} \right],$$

where $[x]$ denotes the greatest integer not exceeding $x$.

Similarly, (4) in Theorem 2, for any $n$, if $X$ is a de Bruijn sequence of length $2^n$, we always have

$$\min_{0 \leq \ell \leq N-1} r_N(\ell; X) \leq 0.$$

To simplify notations, we write

$$r_{\min} = \min_{0 \leq \ell \leq N-1} r_N(\ell; X), \quad r_{\max} = \max_{0 \leq \ell \leq N-1} r_N(\ell; X).$$
as well as
\[ \hat{r}_{\text{max}} = \max_{1 \leq \ell \leq N-1} r_{\text{a}}(\ell; X). \]

As mentioned above, for de Bruijn sequences, \( r_{\text{max}} \) and the bounds of \( \hat{r}_{\text{max}} \) are already clarified. Unfortunately, however, to the best of the authors’ knowledge, for de Bruijn sequences, any lower bounds of \( r_{\text{min}} \) are unknown up to now.

From Example 1, it is easy to check the following remark:

**Remark 2** For \( n = 1 \) and \( n = 2 \), if \( X \) is a de Bruijn sequence of length \( 2^n \), we have \( r_{\text{min}} = -1 \). Besides, for \( n = 3 \), if \( X \) is a de Bruijn sequence of length \( N = 8 \), we obtain \( r_{\text{min}} = -0.5 \).

Thus we are interested in the lower bounds of \( r_{\text{min}} \) for the case \( n \geq 3 \) if \( X \) is a de Bruijn sequence of length \( 2^n \).

By symmetry for \( \ell = 2^{n-1} \), we have
\[ r_{\text{a}}(2^{n-1}; X) = \frac{2}{N} \sum_{i=0}^{2^{n-1}-1} X_i X_{i+2^{n-1}}. \]

Taking account of this form, we obtain for the worst case

**Lemma 1** If \( r_{\text{min}} = -1 \), then we must have
\[ X_i = X_{i+2^{n-1}}, \quad 0 \leq i \leq 2^n - 1 \]

For \( a \in \{0, 1\} \), we use \( \overline{a} \) to denote the binary complement of \( a \), i.e. \( \overline{0} = 1 \) and \( \overline{1} = 0 \).

In virtue of this lemma, we obtain

**Theorem 4** For \( n \geq 3 \), if \( X \) is a de Bruijn sequence of length \( 2^n \), we have
\[ -1 + \frac{4}{2^n} \leq r_{\text{min}} \leq 0. \quad (5) \]

4. Experimental Results

For \( n = 4 \), the normalized auto-correlation properties for all 16 de Bruijn sequences are classified into four patterns. All the patterns are shown in Figures 1 (a) to (d).

Figure 1 (a) shows \( r_{\text{min}} = -0.75 \) and \( \hat{r}_{\text{max}} = 0.5 \), which are the worst values in four patterns. Let us call such sequences the worst sequences if their normalized auto-correlation function achieve one of these worst values. For \( n = 4, 5, \) and 6, the characteristics of the worst sequences are summarized in Table 1. For \( n = 5 \) and 6, the total numbers of de Bruijn sequences are \( 2^{41} = 2048 \) and \( 2^{16} \), respectively. The worst values of \( r_{\text{min}} \) and \( \hat{r}_{\text{max}} \), the time delays \( \ell \) that attain the worst \( r_{\text{min}} \) or \( \hat{r}_{\text{max}} \), and the number of the worst sequences are listed in Table 1.

Remark 2 and Table 1 imply the following remark:

**Remark 3** If \( n = 3 \) and \( n = 4 \), the equality holds for the lower bound of \( r_{\text{min}} \) in (5).

In this sense, the lower bound of \( r_{\text{min}} \) given by (5) is tight.

From the view point of randomness, the normalized auto-correlation functions for pseudo-random sequences are often expected to be like a delta-function. For \( n = 4 \), the sequences with the normalized auto-correlation functions in Figures 1 (b) and (c) suit this requirement. However, even except for the worst sequences, for \( n = 4 \), the sequences having the normalized auto-correlation functions in Figure 1 (d) do not satisfy this requirement.

In addition, Figures 1 (a) to (d) experimentally suggest
\[ \hat{r}_{\text{max}} \leq |r_{\text{min}}|. \]

Figure 1 (d) shows \( r_{\text{min}} = -0.5 \) that is smaller in the absolute value than the worst value \( r_{\text{min}} = -0.75 \) in Figure 1 (a) and equals in the absolute value to the worst value \( \hat{r}_{\text{max}} = 0.5 \) in Figure 1 (a). Let us call such sequences the second-worst sequences if \( r_{\text{min}} \) for the sequences is the smallest in all de Bruijn sequences except the worst sequences. For \( n = 4, 5, \) and 6, the characteristics of the second-worst sequences are summarized in Table 2. The second-worst values of \( r_{\text{min}} \), the time delays \( \ell \) that attain the second-worst \( r_{\text{min}} \), and the number of the second-worst sequences are listed in Table 2.

Experimentally Table 1 and 2 imply that the worst cases \( r_{\text{min}} \) or \( \hat{r}_{\text{max}} \) tend to occur at \( \ell = 2^{n-1} \) if \( n \) becomes large. Intuitively this is because by Lemma 1 \( r_{\text{min}} = -1 \) only if \( \ell = 2^{n-1} \).

Table 1 and 2 provide a class of the worst and the second-worst sequences in terms of the normalized auto-correlation functions. By eliminating the class, we can construct a family of good de Bruijn sequences in terms of such correlation functions.

<table>
<thead>
<tr>
<th>( n )</th>
<th>( r_{\text{min}}, \hat{r}_{\text{max}} )</th>
<th>time delay ( \ell )</th>
<th>the number of seqs</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>(-0.75, 0.5)</td>
<td>( \ell = 4, 12 )</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>(-0.75, 0.5)</td>
<td>( \ell = 8, 24 )</td>
<td>8</td>
</tr>
<tr>
<td>6</td>
<td>(-0.875, 0.625)</td>
<td>( \ell = 32 )</td>
<td>96</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( n )</th>
<th>( r_{\text{min}} )</th>
<th>time delay ( \ell )</th>
<th>the number of seqs</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>(-0.5 )</td>
<td>( \ell = 7, 9 )</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>(-0.625 )</td>
<td>( \ell = 5, 27 )</td>
<td>64</td>
</tr>
<tr>
<td>6</td>
<td>(-0.75 )</td>
<td>( \ell = 32 )</td>
<td>4728</td>
</tr>
</tbody>
</table>
5. Summary

We gave a novel lower bound of the minimum values of the normalized auto-correlation functions for de Bruijn sequences of length $N = 2^n$ ($n \geq 3$). The lower bound was tight in the sense that the equality holds for $n = 3$ and $n = 4$. For $3 \leq n \leq 6$, we experimentally characterized the worst and the second-worst sequences in all de Bruijn sequences in terms of the normalized auto-correlation function.

Acknowledgments

This study was mainly supported by the Grant-in-Aid for Scientific Research (C) under Grant No. 21560392 from the Japan Society for the Promotion of Science. The author thanks Dr. Roger R. Anderson, retired Research Scientist in the Department of Physics and Astronomy, the University of Iowa, USA for his helping the author’s English writings.

References


Figure 1: The four patterns of the normalized auto-correlation function for a de Bruijn sequence of length $2^5$. 
A Perturbation-Based Algorithm with Extremely Long Periods of Generated Cycles

Mieczyslaw Jessa

Faculty of Electronics and Telecommunication, Poznan University of Technology
ul. Polanka 3, 60-965 Poznan, Poland
Email: mjessa@et.put.poznan.pl

Abstract—A new algorithm for perturbing pseudo-chaotic orbits, solving the problem of short cycles produced by chaotic systems realized in finite-state machines, is proposed. The algorithm does not require an external system to generate uniformly distributed pseudo-random sequences. The periods of generated cycles are extremely long and independent of the precision of computations.

1. Introduction

The realization of chaotic maps in finite-state machines leads to serious degradation of chaotic dynamics. The properties of cycles observed in computers depend on the choice of the initial point, and their periods may be unexpectedly short [1, 2]. The problem of dynamical degradation has been addressed in many papers (e.g., [3] and references therein), and it is regarded as one of reasons for the weaknesses of many chaos-based cryptosystems realized in digital machines. Chaotic maps implemented in finite-state machines are known in the literature as pseudo-chaotic maps or digital chaotic maps. Up to now, three basic solutions to improve the properties of pseudo-chaotic maps have been proposed: using a higher finite precision [4, 5], cascading multiple chaotic systems [6] and using a perturbation-based algorithm [7-11]. The first solution does not solve the problem at all, and cascading multiple chaotic systems increases the length of the cycle but complicates digital realization of a chaotic system. In a perturbation-based algorithm, however, a second system is used to perturb the orbits of a pseudo-chaotic map. The use of a perturbation as a method of improving digital chaos was independently proposed by Čermák [7] and Zhou and Ling [8]. It was later improved by Sang et al. [9-10] and adopted for pseudo-chaotic ciphers by Li [11].

In a perturbation-based algorithm, cycles produced by a pseudo-chaotic map are perturbed every Δ iterations, where Δ is an integer. A perturbing sequence comes from another system that produces uniformly distributed pseudo-random sequences. The system can either be realized in the same machine or it can be physically independent of the machine used in computations. An alternative approach is to perturb a control parameter of a chaotic system. In a perturbation-based algorithm, cycles produced by chaotic systems realized in finite-state machines, is proposed. The algorithm does not require an external system to generate uniformly distributed pseudo-random sequences. The periods of generated cycles are extremely long and independent of the precision of computations.

1. Introduction

The realization of chaotic maps in finite-state machines leads to serious degradation of chaotic dynamics. The properties of cycles observed in computers depend on the choice of the initial point, and their periods may be unexpectedly short [1, 2]. The problem of dynamical degradation has been addressed in many papers (e.g., [3] and references therein), and it is regarded as one of reasons for the weaknesses of many chaos-based cryptosystems realized in digital machines. Chaotic maps implemented in finite-state machines are known in the literature as pseudo-chaotic maps or digital chaotic maps. Up to now, three basic solutions to improve the properties of pseudo-chaotic maps have been proposed: using a higher finite precision [4, 5], cascading multiple chaotic systems [6] and using a perturbation-based algorithm [7-11]. The first solution does not solve the problem at all, and cascading multiple chaotic systems increases the length of the cycle but complicates digital realization of a chaotic system. In a perturbation-based algorithm, however, a second system is used to perturb the orbits of a pseudo-chaotic map. The use of a perturbation as a method of improving digital chaos was independently proposed by Čermák [7] and Zhou and Ling [8]. It was later improved by Sang et al. [9-10] and adopted for pseudo-chaotic ciphers by Li [11].

In a perturbation-based algorithm, cycles produced by a pseudo-chaotic map are perturbed every Δ iterations, where Δ is an integer. A perturbing sequence comes from another system that produces uniformly distributed pseudo-random sequences. The system can either be realized in the same machine or it can be physically independent of the machine used in computations. An alternative approach is to perturb a control parameter of a chaotic system. In a perturbation-based algorithm, cycles produced by chaotic systems realized in finite-state machines, is proposed. The algorithm does not require an external system to generate uniformly distributed pseudo-random sequences. The periods of generated cycles are extremely long and independent of the precision of computations.

1. Introduction

The realization of chaotic maps in finite-state machines leads to serious degradation of chaotic dynamics. The properties of cycles observed in computers depend on the choice of the initial point, and their periods may be unexpectedly short [1, 2]. The problem of dynamical degradation has been addressed in many papers (e.g., [3] and references therein), and it is regarded as one of reasons for the weaknesses of many chaos-based cryptosystems realized in digital machines. Chaotic maps implemented in finite-state machines are known in the literature as pseudo-chaotic maps or digital chaotic maps. Up to now, three basic solutions to improve the properties of pseudo-chaotic maps have been proposed: using a higher finite precision [4, 5], cascading multiple chaotic systems [6] and using a perturbation-based algorithm [7-11]. The first solution does not solve the problem at all, and cascading multiple chaotic systems increases the length of the cycle but complicates digital realization of a chaotic system. In a perturbation-based algorithm, however, a second system is used to perturb the orbits of a pseudo-chaotic map. The use of a perturbation as a method of improving digital chaos was independently proposed by Čermák [7] and Zhou and Ling [8]. It was later improved by Sang et al. [9-10] and adopted for pseudo-chaotic ciphers by Li [11].

In a perturbation-based algorithm, cycles produced by a pseudo-chaotic map are perturbed every Δ iterations, where Δ is an integer. A perturbing sequence comes from another system that produces uniformly distributed pseudo-random sequences. The system can either be realized in the same machine or it can be physically independent of the machine used in computations. An alternative approach is to perturb a control parameter of a chaotic system. In a perturbation-based algorithm, cycles produced by chaotic systems realized in finite-state machines, is proposed. The algorithm does not require an external system to generate uniformly distributed pseudo-random sequences. The periods of generated cycles are extremely long and independent of the precision of computations.

1. Introduction

The realization of chaotic maps in finite-state machines leads to serious degradation of chaotic dynamics. The properties of cycles observed in computers depend on the choice of the initial point, and their periods may be unexpectedly short [1, 2]. The problem of dynamical degradation has been addressed in many papers (e.g., [3] and references therein), and it is regarded as one of reasons for the weaknesses of many chaos-based cryptosystems realized in digital machines. Chaotic maps implemented in finite-state machines are known in the literature as pseudo-chaotic maps or digital chaotic maps. Up to now, three basic solutions to improve the properties of pseudo-chaotic maps have been proposed: using a higher finite precision [4, 5], cascading multiple chaotic systems [6] and using a perturbation-based algorithm [7-11]. The first solution does not solve the problem at all, and cascading multiple chaotic systems increases the length of the cycle but complicates digital realization of a chaotic system. In a perturbation-based algorithm, however, a second system is used to perturb the orbits of a pseudo-chaotic map. The use of a perturbation as a method of improving digital chaos was independently proposed by Čermák [7] and Zhou and Ling [8]. It was later improved by Sang et al. [9-10] and adopted for pseudo-chaotic ciphers by Li [11].

In a perturbation-based algorithm, cycles produced by a pseudo-chaotic map are perturbed every Δ iterations, where Δ is an integer. A perturbing sequence comes from another system that produces uniformly distributed pseudo-random sequences. The system can either be realized in the same machine or it can be physically independent of the machine used in computations. An alternative approach is to perturb a control parameter of a chaotic system. In a perturbation-based algorithm, cycles produced by chaotic systems realized in finite-state machines, is proposed. The algorithm does not require an external system to generate uniformly distributed pseudo-random sequences. The periods of generated cycles are extremely long and independent of the precision of computations.

1. Introduction

The realization of chaotic maps in finite-state machines leads to serious degradation of chaotic dynamics. The properties of cycles observed in computers depend on the choice of the initial point, and their periods may be unexpectedly short [1, 2]. The problem of dynamical degradation has been addressed in many papers (e.g., [3] and references therein), and it is regarded as one of reasons for the weaknesses of many chaos-based cryptosystems realized in digital machines. Chaotic maps implemented in finite-state machines are known in the literature as pseudo-chaotic maps or digital chaotic maps. Up to now, three basic solutions to improve the properties of pseudo-chaotic maps have been proposed: using a higher finite precision [4, 5], cascading multiple chaotic systems [6] and using a perturbation-based algorithm [7-11]. The first solution does not solve the problem at all, and cascading multiple chaotic systems increases the length of the cycle but complicates digital realization of a chaotic system. In a perturbation-based algorithm, however, a second system is used to perturb the orbits of a pseudo-chaotic map. The use of a perturbation as a method of improving digital chaos was independently proposed by Čermák [7] and Zhou and Ling [8]. It was later improved by Sang et al. [9-10] and adopted for pseudo-chaotic ciphers by Li [11].

In a perturbation-based algorithm, cycles produced by a pseudo-chaotic map are perturbed every Δ iterations, where Δ is an integer. A perturbing sequence comes from another system that produces uniformly distributed pseudo-random sequences. The system can either be realized in the same machine or it can be physically independent of the machine used in computations. An alternative approach is to perturb a control parameter of a chaotic system. In a perturbation-based algorithm, cycles produced by chaotic systems realized in finite-state machines, is proposed. The algorithm does not require an external system to generate uniformly distributed pseudo-random sequences. The periods of generated cycles are extremely long and independent of the precision of computations.
13]. Using the method of Bays and Durham, we initially compute the L elements of sequence \( \{x_k\} \). These are written into successive cells of an auxiliary table \( T \) with size \( L \). The value of the next iteration, i.e., \( x_k = f(x_{k-1}) \), is the first number (\( u_1 \)) of the output stream. The same \( x_k \) is used to compute the address \( t_1 \) of a cell of \( T \). The number read off from this cell is the second number (\( u_2 \)) of the output sequence. We generate the next number \( x_{k+1} = f(x_k) \) and write it into \( T \) in the place of number \( u_2 \).

Next, we use \( u_2 \) to compute the next address \( t_2 \). The number read off from the cell with address \( t_2 \) is the third number (\( u_3 \)) of the output stream. We compute \( x_{k+1} = f(x_{k+1}) \) and write it in place of \( u_3 \), etc. [12, 13]. The addresses of the cells of table \( T \) are computed from the equation

\[
t_k = \left[ L \cdot u_k \right], \quad N = 1, 2, \ldots
\]

where it was assumed that \( u_n \in [0, 1) \). The period \( m_o \) of \( \{u_n\} \) is [13]

\[
m_o = O(m \cdot L)^{\varpi},
\]

where \( m_o \) is the period of sequence \( \{x_k\} \). Period (5) is achieved for

\[
L < m < L!.
\]

If \( m \) does not satisfy (6), the period of \( \{u_n\} \) either does not change or increases slightly [12]. It should be noted that if shuffling improves sufficiently the properties of pseudo-chaotic orbits, we do not need to apply any perturbation algorithm at all. In other cases, the implementation of a perturbation is a better solution.

The simplest method for perturbing the low-order bits of \( x_k \) is the computation of the XOR function of these bits as well as the bits encoding symbols \( t_k \). Generally, such an operation is dangerous because the period of \( \{t_k\} \) may be very short compared with the period of \( \{u_n\} \). In this paper, we propose another approach that exploits all bits of some elements of sequence \( \{u_n\} \), computed for \( N = (k + 1) \lambda \), \( k = 0, 1, \ldots \). Sequence \( \{u_n\} \) is obtained from a perturbed sequence \( \{y_n\} \) shuffled in table \( T \) (Fig. 1).

The constant \( c \) is computed as follows. We assume that all numbers are encoded by \( l \) bits \( b_{j,1} \), \( j = 1, 2, \ldots, l \) and that \( l \) has \( c \) divisors. For example, for \( l = 16 \) the divisors are 1, 2, 4, 8, and 16. Denoting by \( d \) one of the divisors, we can divide the sequence of \( l \) bits into \( c = l/d \) disjoint blocks \( B_{c,i} \) \( (i = 1, 2, \ldots, c) \) of bits with length \( d \). Then,

\[
\tilde{u}_n = 0.b_{c,1}, b_{c,2}, \ldots, b_{c,c-1} = 0.B_{c,1}B_{c,2} \ldots B_{c,c} \in \mathbb{I} = [0, 1),
\]

where

\[
B_{c,i} = \{b_{c,i,1}, b_{c,i,2}, \ldots, b_{c,i,c-1}\}.
\]

If \( N = (k + 1) \lambda \), then we take \( \tilde{u}_n \) to produce \( c \) digital words \( B_{c,i} \). The words are next used in \( c \) successive perturbations. A pseudo-chaotic orbit is perturbed by means of signal

\[
z_n = \begin{cases} 0 & \text{for } n \neq (k + 1) \lambda \\ B_{N-1,1+k \mod c} & \text{for } n = (k + 1) \lambda \end{cases},
\]

where \( B_{0,1+k \mod c} \) \( k = 0, 1, \ldots \) denotes \( B_{c,i} \) \( i = 1, 2, \ldots, c \), which encodes \( \tilde{u}_n \). After the perturbation, the number \( k \) is increased by unity and remains unchanged until the next perturbation. It plays the role of the perturbation counter.

The analysis of the period of a perturbed sequence is very difficult, but we can find the lower bound of its value. This is sufficient for most applications, such as in cryptography. Let us assume that \( m_i \) is the period of \( \{y_n\} \) and \( m_s \) is the period of \( \{\tilde{u}_n\} \). If

\[
L < m < L!.
\]

then

\[
m_i = O(m \cdot L)^{\varpi},
\]

where

\[
m_i > m_s.
\]

The period \( m_i \) of a pseudo-chaotic sequence perturbed every \( \Delta \) iterations of \( f \) is the least common multiple of period \( m_p \) of a perturbing sequence \( \{z_n\} \) and \( \Delta \),

\[
m_i = LCM(m_p, \Delta),
\]

If \( m_p \) and \( \Delta \) are relatively prime, then \( m_i \) is the product of \( m_p \) and \( \Delta \). In other cases, we can write that

\[
m_i \geq m_s.
\]

where \( m_s > \Delta \). In the proposed method, the perturbing signal is derived from the shuffled sequence \( \{\tilde{u}_n\} \) with period \( m_s \). The period \( m_p \) of the perturbing sequence is \( c \Delta \) times shorter than \( m_s \). Thus, from (14), we obtain that

\[
m_i \geq \frac{m_s}{c \Delta}.
\]

The results (12) and (15) are inconsistent. Consequently, the value of \( m_i \) increases with the increase in the number of iterations. Longer \( \{\tilde{u}_n\} \) yields longer \( \{y_n\} \) and longer \( \{y_n\} \) produces longer \( \{\tilde{u}_n\} \) which, in turn, increases the length of \( \{y_n\} \) and so on. The process ends when \( m_i \) stops to satisfy (10). The value of \( m_i \) becomes constant, and successive numbers begin to repeat with a constant period.

The determination of the exact \( m_i \) that can be regarded as significantly smaller than \( L! \) is practically impossible. In the technical sciences, it is usually assumed
that a number $\alpha$ is significantly smaller than a number $\beta$ if $\alpha \leq \beta / 10$. An exception to this is in the field of metrology, in which the requirement is that $\alpha \leq \beta / 100$. From a practical point of view, it is safer to assume that $m_\gamma$ stops increasing when

$$m_\gamma = m \geq \left\lfloor \frac{L!}{100} \right\rfloor.$$ (16)

For large $L$ this value is significantly greater than $O(mL \lambda)^3$ assessed by Bays and Durham [12]. To obtain periods equal to or longer than $L! / 100$, the period $m_\gamma$ of an unperturbed pseudo-chaotic sequence has to satisfy condition (6). For example, if table $T$ has 128 cells, we have the condition that $2^7 << m_\gamma << 128! \approx 1 \cdot 2^{77}$. It is easy to satisfy the condition for the unperturbed orbits of pseudo-chaotic maps.

Let us emphasize that formula (5) was obtained by Bays and Durham under the assumption that $\{x_n\}$ is an approximation of independent and uniformly distributed random numbers from unit interval $I = [0,1]$. Therefore, the analysis of the period of perturbed sequence is valid for maps $f$ that generate sequences satisfying the same assumption. If we apply the proposed self-perturbing method to a map that do not produce uniformly distributed numbers, the smallest period of sequence $\{y_n\}$ may not satisfy inequality (16).

3. Numerical Experiment

In a finite-state machine with a $l$-bit digital word, a real number is approximated by a rational number. We can perform all computations either traditionally, i.e., with the use of floating-point arithmetic, or we can express the dynamic variables of (2) as $p / 2^l$, where $p \in \mathbb{N}$, $0 < p < 2^l$.

In the ANSI/IEEE double-precision floating-point arithmetic, where the fractional part of a real number requires 52 bits, we can divide the mantissa into $e = 52 / d$ $d$-bit blocks. The number $d$ is one of six divisors: 1, 2, 4, 13, 26, 52 of $l = 52$. Next, we perturb numbers $\{x_n\}$ every $\Delta$ iterations of (2), according to the procedure described in Section 2. Another approach, which is faster and free from errors introduced by the floating-point arithmetic, exploits the transformation of $f$ into the set of integer numbers. It was first proposed in paper [14]. Formula (2) takes the form

$$p_{n+1} = \left\lfloor 2^l \cdot f(p_n / 2^l) \right\rfloor, \quad n = 0, 1, \ldots$$ (17)

In the numerical experiment, we found the period of perturbed cycles produced by the Rényi chaotic map for six initial points. For the Rényi map

$$x_{\alpha+1} = (\lambda \cdot x_\alpha) \mod 1, \quad x \in [0,1], \quad \lambda \in \mathbb{R},$$ (18)

where $\lambda$ is a real number, we obtain

$$p_{n+1} = \left\lfloor 2^l \cdot (\lambda \cdot (p_n / 2^l) \mod 1) \right\rfloor$$ (19)

or equivalently

$$p_{n+1} = \left\lfloor (\lambda \cdot p_n) \mod 2^l \right\rfloor.$$ (20)

Because the observed periods can be extremely long, it was assumed in the numerical experiment that all numbers were encoded only by $l = 16$ bits and that the size $L$ of table $T$ was small but satisfied condition (6). Table 1 shows the periods of unperturbed and perturbed orbits. The orbits were produced by a Rényi chaotic map with parameter $\lambda$ equal to 2.8. Parameter $\Delta$ was equal to 500.

During the perturbation, the XOR function between $x_n$ and $d = 8$ low-order bits of $p_n$ was computed. Because the perturbation may lead to $x = 0$ and, consequently, to a sequence of zeros, the number $x = 0$ was always replaced by the number $x = 2^{-l}$. Table 1 also contains the smallest period computed from (16) (the numbers in parentheses, below the period found experimentally).

Table 1. The period of unperturbed and perturbed pseudo-chaotic orbits as a function of initial point and the size $L$ of table $T$

<table>
<thead>
<tr>
<th>$x_0$</th>
<th>No perturbation</th>
<th>$L=9$</th>
<th>$L=10$</th>
<th>$L=11$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1/2^a$</td>
<td>715</td>
<td>12073000 (3628)</td>
<td>26565000 (3628)</td>
<td>388116000 (399168)</td>
</tr>
<tr>
<td>$10/2^a$</td>
<td>715</td>
<td>12073000 (3628)</td>
<td>27004000 (3628)</td>
<td>388116000 (399168)</td>
</tr>
<tr>
<td>$10^2/2^a$</td>
<td>715</td>
<td>680000 (3628)</td>
<td>27004000 (3628)</td>
<td>388116000 (399168)</td>
</tr>
<tr>
<td>$10^3/2^a$</td>
<td>715</td>
<td>12073000 (3628)</td>
<td>25322000 (3628)</td>
<td>98864000 (399168)</td>
</tr>
<tr>
<td>$10^4/2^a$</td>
<td>715</td>
<td>680000 (3628)</td>
<td>27004000 (3628)</td>
<td>105575000 (399168)</td>
</tr>
<tr>
<td>$10^5/2^a$</td>
<td>715</td>
<td>12073000 (3628)</td>
<td>27004000 (3628)</td>
<td>115570000 (399168)</td>
</tr>
</tbody>
</table>

The period of unperturbed orbits was constant for all initial points considered in the experiment and equal to 715. Consequently, condition (6) is satisfied for $L \geq 9$. During iterations of $f$, the value of $m_\gamma$ increases with the increase in the number of iterations. The process ends when $m_\gamma$ ceases to satisfy (10). To experimentally find the final value of $m_\gamma$, we have to omit a large number of intermediate states. We first perform some number $n_\gamma$ of initial iterations. For example, for $L = 11$, the effective searching of period 388116000 requires $n_\gamma = O(10^9)$ initial iterations. Number $n_\gamma$ is greater than $L! = 11! = 39916800$. It seems that such a long intermediate state with irregular behavior of the perturbed pseudo-chaotic orbit can be used in many applications instead of a fragment of the periodic sequence.

4. Limitations of perturbation

One of the limitations of perturbation is that perturbation may yield a sequence that terminates in a sequence of zeros or ones. As was shown in Section 3, this
can usually be eliminated without significant computational effort. A much more serious problem is the repetition of long fragments of cycles produced by \( f \) when \( f \) is implemented in the computer. If \( f \) produces many short orbits for different initial conditions, perturbation leads to orbit hopping. Because the orbits are short, we have to make perturbations frequently. If \( f \) produces long orbits, their number may be small. The probability then rapidly increases that a perturbation will repeat a long fragment of an unperturbed orbit. To overcome this serious disadvantage, we can, for example, combine signals produced by many independent pseudo-chaotic systems.

In this paper, we propose to use the shuffling of Bays and Durham to perturbed sequence \( \{y_n\} \). This is simpler and requires less computational effort. If the perturbation yields a point that was previously generated and, consequently, a long part of a previously produced sequence repeats, additional shuffling changes the order of appearances of elements of this sequence. This is true if and only if during the perturbation table \( T' \) does not contain the same numbers and in the same order as for the previous sequence. The number \( M \) of different contents of \( T' \) can be computed from the equation

\[
M = 2^l \left[ \sum_{i=1}^{L'} \left( \sum_{j=1}^{l} \frac{1}{j!} \right) \right], \tag{21}
\]

where \( l \) is the number of bits that encode numbers and \( L' \) is the size of the additional table. If numbers written into \( T' \) are equally probable, the probability \( P \) that \( T' \) contains the same numbers and in the same order as for the previous sequence is equal to \( 1/M \). For example, if \( L' = 64 \) and \( l = 16 \), we obtain \( P = 2^{-1024} \).

If \( L' \) does not satisfy condition \( L' << m, << L'! \), then the period of the sequence shuffled in \( T' \) is comparable to the period of the perturbed sequence \( \{y_n\} \). Let us emphasize that additional shuffling does not introduce security to sequence \( \{y_n\} \) [15].

**5. Conclusions**

In this paper, a new algorithm to perturb pseudo-chaotic orbits observed in finite-state machines has been proposed. In comparison with the existing algorithms, this algorithm does not require the perturbing signal to be generated by an external source. The period of the generated cycle depends on the size of the table used to shuffle a perturbed pseudo-chaotic orbit and may be a huge number, independent of the precision of computations. A method for preventing the repetition of long fragments of an unperturbed orbit observed in a perturbed signal has also been considered. The proposed algorithm can be used in applications requiring sequences with extremely long periods, e.g., in cryptography. Another application may be the generation of long-period, pseudo-random sequences in finite-state machines with a small number of states. The achievable periods can be significantly greater than periods obtained for the shuffling algorithm of Bays and Durham. The method is very fast and can be easily implemented in contemporary field programmable gate arrays. The subject of future research should be rigorous analysis of the period of sequence \( \{y_n\} \) for arbitrary map \( f \). The fact that \( \{y_n\} \) and \( \{\tilde{a}_n\} \) depend on each other makes the analysis difficult.

**References**


Spreading Sequences with Negative Auto-correlation  
Based on Chaos Theory and Gold Sequences
— Increase of Family Sizes and Performance Evaluation —

Akio Tsuneda and Taizo Sagara

Department of Computer Science and Electrical Engineering, Kumamoto University
2–39–1 Kurokami, Kumamoto, 860–8555 JAPAN
E-mail: tsuneda@cs.kumamoto-u.ac.jp

Abstract—Spreading sequences with appropriate negative auto-correlation can reduce average multiple access interference (MAI) in asynchronous DS/CDMA systems compared with the conventional spreading sequences such as Gold sequences generated by linear feedback shift registers (LFSRs). We design spreading sequences with negative auto-correlation based on Gold sequences and the chaos theory for the Bernoulli map. The family size of the proposed sequences is 6 times as large as that of the original Gold sequences. By computer simulations, we evaluate BER performances of asynchronous DS/CDMA systems using the proposed sequences.

1. Introduction

Linear feedback shift register (LFSR) sequences (e.g., M-sequences, Gold sequences, Kasami sequences) are the most well-known pseudo-random sequences and they are practically used for spreading sequences in direct-sequence code division multiple access (DS/CDMA) systems [1].

Many types of spreading sequences have been proposed for enhancement of system performance such as bit error rate (BER). Especially, it is remarkable that spreading sequences with exponentially vanishing negative auto-correlations can reduce multiple access interference (MAI) in asynchronous DS/CDMA systems compared with classical spreading codes such as M-sequences, Gold sequences and Kasami sequences [2],[3]. Such negatively auto-correlated sequences can be generated by using one-dimensional nonlinear chaotic maps, which are called chaotic sequences [2]–[4]. Their discretized version called maximal-period sequences has also been considered [5].

Furthermore, we designed binary sequences with negative (but not exponentially vanishing) auto-correlation based on the well-known Bernoulli and tent maps [6]. Theoretically, the performances of such sequences are slightly worse than the sequences with exponentially vanishing negative auto-correlations with respect to MAI reduction. However, the proposed sequences can be generated by simpler chaotic maps than the others, though our binary functions are somewhat complex. Noting that the Bernoulli/tent maps with finite bits can be realized by a class of nonlinear feedback shift registers (NFSRs) [7], we proposed NFSR-based generators of negatively auto-correlated binary sequences and revealed that the proposed sequences can also reduce BER in asynchronous DS/CDMA systems compared with conventional Gold sequences [6]. However, the circuit scale of NFSRs is, in general, much larger than LFSRs.

We also designed periodic binary sequences with negative auto-correlation based on Gold sequences generated by two LFSRs [8]. Namely, we designed such sequences based on the chaos theory for the Bernoulli map because random binary sequences can also be regarded as finite-bit approximation of the Bernoulli map [8]. By computer simulations, we revealed that the proposed sequences can reduce BER in asynchronous DS/CDMA communications [9].

In this paper, we also design periodic binary sequences with negative auto-correlation based on the chaos theory and Gold sequences, where the family size of the proposed sequences is 6 times as large as that of the original Gold sequences [10]. By computer simulations, we investigate BER performances of the proposed sequences in asynchronous DS/CDMA communications.

2. Chaos-Based Sequences and Their Performance

In asynchronous DS/CDMA systems, the average interference parameter (AIP) is defined by [11]

\[ r_{kl} = 2N^2 + 4 \sum_{i=1}^{N-1} A_k(f)A_i(f) + \sum_{i=1}^{N-1} A_k(f)A_i(f+1), \]

where \( A_k(f) \) is an aperiodic auto-correlation function of the k-th user’s spreading sequence \( \{B_n^{(k)}\}_{n=1}^{N-1} \) with period \( N \), defined by

\[ A_k(f) = \begin{cases} 
\sum_{n=0}^{N-1-f} B_n^{(k)}B_{n+f}^{(k)} & (0 \leq f \leq N-1) \\
\sum_{n=0}^{N-1-f} B_n^{(k)}B_{n+f}^{(k)} & (1-N \leq f < 0) \\
0 & (|f| > N).
\end{cases} \]
Now we briefly introduce generation of chaotic sequences and their statistical analyses. Using one-dimensional nonlinear difference equation defined by

$$x_{n+1} = \tau(x_n), \quad x_n \in I = [d,e], \quad n = 0, 1, 2, \ldots$$ \hspace{1cm} (3)

we can generate a chaotic real-valued sequence \(\{x_n\}_{n=0}^{\infty}\), where \(x_n = \tau^n(x_0)\). We transform such a real-valued sequence into a binary sequence \(\{\beta(\tau^n(x))\}_{n=0}^{\infty}\) (\(\beta(x) \in \{-1, 1\}\)). The theoretical auto-correlation function of such a binary sequence \(\{\beta(\tau^n(x))\}_{n=0}^{\infty}\) is defined by

$$C(\ell; \beta) = E[\beta(x)\beta(\tau^\ell(x))] = \int \beta(x)\beta(\tau^\ell(x))f^*(x)dx$$ \hspace{1cm} (4)

under the assumption that \(\tau(x)\) has an invariant density \(f^*(x)\), where \(E[\cdot]\) denotes the expectation. Assume that \(K\) users use chaotic binary sequences \(\{\beta(\tau^n(x^{(i)}))\}_{n=0}^{N-1}\) \((i = 1, 2, \ldots, K)\) of length \(N\) as their spreading codes, where the initial values \(x^{(1)}, x^{(2)}, \ldots, x^{(K)}\) are statistically independent of each other. The average interference parameter (AIP) for a user in such a system is given by

$$\hat{r} = 2N^2 + 4\sum_{i=1}^{N-1} (N-\ell)^2 C(\ell; \beta)^2$$

$$+ 2\sum_{i=1}^{N-1} (N-\ell)(N-\ell+1)C(\ell; \beta)C(\ell-1; \beta).$$ \hspace{1cm} (5)

Note that eq.(5) is obtained by averaging eq.(1) with the invariant density \(f^*(x)\). We also define a normalized AIP by

$$R = \lim_{N \to \infty} \frac{\hat{r}}{2N^2}.$$ \hspace{1cm} (6)

Obviously, we have \(R = 1\) for uncorrelated sequences with \(C(\ell; \beta) = 0\) \((\ell \geq 1)\).

First, consider the case \(C(\ell; \beta) = \lambda^\ell\) \((|\lambda| < 1)\), that is, chaotic sequences with exponentially vanishing auto-correlations. In this case, we have

$$R = \frac{\lambda^2 + \lambda + 1}{1-\lambda^2}$$ \hspace{1cm} (7)

which takes the minimum value \(\frac{\sqrt{5}}{2} \approx 0.875\) when \(\lambda = -2 + \sqrt{3}\) \([2],[3]\). Thus such sequences have smaller AIPS than uncorrelated sequences with \(R = 1\).

Next consider the sequences whose auto-correlation function is given by

$$C(\ell; \beta) = \begin{cases} 1 & (\ell = 0), \\ \epsilon & (\ell = 1), \\ 0 & (\ell \geq 2), \end{cases}$$ \hspace{1cm} (8)

where \(|\epsilon| < 1\). In this case, we have

$$R = 2\epsilon^2 + \epsilon + 1.$$ \hspace{1cm} (9)

The minimum value of \(R\) is \(\frac{\sqrt{5}}{2} \approx 0.875\) when \(\epsilon = -\frac{1}{2}\), which is slightly larger than \(\frac{\sqrt{5}}{2} \approx 0.866\) of the optimum case \(C(\ell; \beta) = \lambda^\ell\) with \(\lambda = -2 + \sqrt{3}\) but the difference is quite small. Of course, the sequences of this case \((\epsilon = -\frac{1}{2})\) also outperform the uncorrelated sequences.

Several types of chaotic maps which can generate chaotic sequences with exponentially vanishing auto-correlations are known \([3],[4]\). Most of them are piecewise linear Markov maps. Here, we consider the Bernoulli map \(\tau_B(x)\) defined by

$$\tau_B(x) = \begin{cases} 2x & (0 \leq x < \frac{1}{2}) \\ 2x-1 & (\frac{1}{2} \leq x \leq 1), \end{cases}$$ \hspace{1cm} (10)

which is one of the simplest piecewise linear chaotic maps with the interval \(I = [0,1]\) and \(f^*(x) = 1\). Furthermore, we define six binary functions by \([10]\)

$$B_i(x) = \begin{cases} \Theta_i(x) - \Theta_{i+1}(x) + \Theta_{i+2}(x) - \Theta_{i+3}(x) & (i = 1, 2, \ldots, 6), \end{cases}$$ \hspace{1cm} (11)

where \(\Theta_i(x)\) is a threshold function defined by

$$\Theta_i(x) = \begin{cases} 0 & (x < t) \\ 1 & (x \geq t). \end{cases}$$ \hspace{1cm} (12)

Here, we define \(B_i(x) = 2B_i(x) - 1\) \((i = 1, 2, \ldots, 6)\) for transformation \([0,1] \to [-1,1]\). By the chaos theory for the Bernoulli map, we can show that the auto-correlation function of the chaotic binary sequences \(\{B_i(\tau_B^n(x))\}_{n=0}^{\infty}\) \((i = 1, 2, \ldots, 6)\) is given by

$$C(\ell; B_i) = \begin{cases} 1 & (\ell = 0) \\ -\frac{1}{2} & (\ell = 1) \\ 0 & (\ell \geq 2). \end{cases}$$ \hspace{1cm} (13)

This implies that the sequences \(\{B_i(\tau_B^n(x))\}_{n=0}^{\infty}\) are optimal spreading codes in a class of sequences with the auto-correlation function given by eq.(8).

3. Negatively Correlated Sequences Based on Gold Sequences

3.1. Gold Sequences and Proposed Generator

Gold sequences can be generated by two \(k\)-stage linear feedback shift registers (LFSRs) generating preferred pairs of M-sequences \([1]\). Let \(\{g_n\}_{n=0}^{N-1}\) be an Gold sequence, where \(g_n \in \{0,1\}\) and \(N = 2^k - 1\). The family size of the Gold sequences is \(2^k + 1\) including the original preferred pairs of M-sequences \([1]\). If we observe \(m\) successive bits of a Gold sequence, we get a decimal integer by

$$x_n = g_{n-m} \cdot 2^{m-1} + g_{n-m+1} \cdot 2^{m-2} + \cdots + g_{n-1} \cdot 2^0.$$ \hspace{1cm} (14)
By plotting \((x_n, x_{n+1})\), we obtain a one-dimensional (1-D) map (so called, return map) of the Gold sequence. We can easily confirm that the shape of such a 1-D map (return map) is similar to the Bernoulli map [8],[9]. In this sense, the chaos theory can be applied to the Gold sequences.

Thus, we propose a sequence generator based on LFSRs generating Gold sequences as shown in Fig.1, where the output binary sequence \(b_n^{(i)}\) is obtained by

\[
b_n^{(i)} = \begin{cases} 
1 & \text{if } c_0c_{1}c_2 \in B_i \\
0 & \text{otherwise,}
\end{cases}
\]

where

\[
B_1 = [010, 011, 100, 110] \\
B_2 = [001, 010, 011, 110] \\
B_3 = [010, 011, 100, 111] \\
B_4 = [001, 010, 110, 111] \\
B_5 = [001, 011, 100, 101] \\
B_6 = [001, 100, 101, 110].
\]

which correspond to the binary functions given by eq.(11), that is, the binary sequence \(2b_n^{(i)} - 1\) is finite-bit approximation of the chaotic binary sequence \(B_i\left(\tau_n g(x)\right)_{n=0}^{\infty}\) with the correlation function of eq.(13). Hence, the proposed generator is expected to generate negatively correlated binary sequences similar to the chaotic binary sequences.

Also it should be noted that the family size of the proposed sequences is 6 times as large as that of the original Gold sequences, i.e., \((2^k+1) \times 6\), by using the 6 binary (logic) functions for every Gold sequence.

3.2. Correlation Properties

Next, we investigated correlation properties of the proposed sequences. Fig.2 (a) shows the average auto-correlation function of the proposed sequences for \(k = 7\), where the auto-correlation values are averaged for 129 sequences randomly chosen from a family of the sequences. It is shown that the average auto-correlation function is almost equal to the theoretical one.

Fig.2 (b) shows the distribution of the cross-correlation values of the proposed sequences for \(k = 7\), where all the possible pairs of the 129 sequences are taken into account. The cross-correlations of 129 original Gold sequences are also shown in the figure. The distribution is similar to the Gaussian distribution with 0 mean and the maximum cross-correlation value is larger than that of Gold sequences.

3.3. Simulations of Asynchronous DS/CDMA

We performed computer simulations of asynchronous DS/CDMA communications using the proposed sequences. In these simulations, the number of transmitted information bits per user is 1,000 and there are random delays be-
Figure 3: BER performances of the proposed sequences in asynchronous DS/CDMA communications (k = 7)

4. Concluding Remarks

We have designed spreading sequences with negative auto-correlation based on the well-known LFSR sequences (Gold sequences). The design is based on the chaos theory for the Bernoulli map. By computer simulations of asynchronous DS/CDMA communications, we have shown that the proposed sequences can reduce the BER compared with the original Gold sequences. The proposed sequence generator is obtained just by adding a combinational logic circuit with 3 inputs and 1 output to the Gold sequence generator. Furthermore, the family size of the proposed sequences is 6 times as large as that of the original Gold sequences. Hence, we can conclude that the proposed sequences are very useful for asynchronous DS/CDMA communication systems.

Acknowledgments

This work was partly supported by The Telecommunications Advancement Foundation of Japan.

References


Is Markov code superior to i.i.d. in communication systems?
Tohru Kohda†, Yutaka Jitsumatsu† and Kazuyuki Aihara‡

†Dept. of Informatics, Kyushu University
744 Motooka, Nishi-ku, Fukuoka, 819-0395, Japan.
{kohda, jitumatsu}@inf.kyushu-u.ac.jp

‡Institute of Industrial Science, The University of Tokyo
4-6-1 Komaba Meguro-ku, Tokyo 153-8505, Japan.
aihara@sat.t.u-tokyo.ac.jp

Abstract—We propose a CDMA system which allows each user's signal to have a frequency offset. Such a system can be regarded as a frequency dual of a chip-asynchronous DS/CDMA system. Time-domain Markovian codes reduce the variance of multiple-access interference (MAI) in chip-asynchronous systems, while in such a dual system, MAI can be reduced by frequency-domain Markov codes.

1. Introduction

The variance of multiple-access interference (MAI) in chip-asynchronous DS/CDMA systems with independent and identically distributed (i.i.d.) codes is smaller than that in chip-synchronous as well as symbol-synchronous DS/CDMA systems. The MAI is further reduced if we replace i.i.d. codes with Markovian spreading codes generated by a chaotic map [1, 2, 3]. This fact shows that a chaotic map is promising for a spread spectrum (SS) code generation. In this paper we consider a question: in which situation, Markov codes can outperform i.i.d. codes? This work is a preliminary study approaching to this question.

We propose a frequency dual of a chip-asynchronous DS/CDMA system. It is shown that Markov codes outperform i.i.d. ones in such a system, as well. Phase, timing, and frequency synchronization errors cause interference. Thus, a receiver which is robust to these errors is desirable. Therefore such a dual system which allows a random frequency offset is interesting. In a DS/CDMA system, symbol duration is divided into $N$ chip intervals, where $N$ is a spreading factor. In the dual system, on the other hand, frequency band is divided into several sub-bands. Therefore we refer to this system as frequency division (FD)-based CDMA, whereas DS system can be referred to as a time division (TD)-based system.

In order to investigate the condition for Markov codes to be superior to i.i.d., we compare the proposed system with orthogonal frequency division multiplex (OFDM) as well as frequency-hopping (FH) CDMA. It is suggested that the interference caused by synchronization errors may be reduced by Markov codes, if a communication system is properly designed.

This research is supported by the Japan Society for the Promotion of Science (JSPS) through its Funding Program for World-Leading Innovative R&D on Science and Technology (FIRST Program).
is the Pursley’s aperiodic cross-correlation function between \(X\) and \(Y\) [4]. As in [5], we assume that for some positive integer \(m\), \(\ell_{ij}\) ∈ \([0, 1, \ldots, N - 1]\), and \(k_{ij}\) ∈ \([0, 1, \ldots, m - 1]\), the relative time delay is expressed as

\[
\tau_i - \tau_j = \left(\ell_{ij} + \frac{k_{ij}}{m}\right)T_c
\]  

(4)

The system is said to be synchronous if \(\ell_{ij} = k_{ij} = 0\) for all \(i, j\), chip-synchronous if \(k_{ij} = 0\) but \(\ell_{ij} \neq 0\), and chip-asynchronous if \(k_{ij} \neq 0\). The multiple-access interference (MAI) is defined as

\[
I_{j,p}^{(i)} = \sum_{j=1,j\neq i}^{J} \left(\frac{d_{j}^{(i)}}{2} + \frac{d_{p+1}^{(i)}}{m} R_{\text{MA}}^{E}(\ell_{ij} m + k_{ij}; X_{\text{up}}, X_{\text{up}}) + d_{p}^{(i)} - d_{p+1}^{(i)} + \frac{1}{2} R_{\text{MA}}^{O}(\ell_{ij} m + k_{ij}; X_{\text{up}}, X_{\text{up}})\right),
\]

(5)

where \(X_{\text{up}}\) is an up-sampled sequence by a factor of \(m\), defined as

\[
X_{\text{up}} = \{X_0, \ldots, X_0, X_1, \ldots, X_1, \ldots, X_{N-1}, \ldots, X_{N-1}\}.
\]

Let us consider a CDMA system which is a frequency dual of (1) and (2). The Fourier transform of these equations are, respectively,

\[
\hat{\delta}_i(f) = \sum_{j=0}^{N-1} X_{n} T_c \text{sinc}(T_c f) e^{j\pi(1-2n)fT_c},
\]

and

\[
\hat{r}(f) = \sum_{j=1}^{K} \sum_{p=-\infty}^{\infty} d_{j}^{(i)} \hat{\delta}_i(f) e^{-j2\pi f(\tau_j+pT)} + \hat{n}_0(f) = \hat{n}_0(f),
\]

where \(\text{sinc}(t) = \sin(\pi t)/(\pi t)\). Replacing the time domain spreading codes \(X_j\), the frequency variable \(f\), the data duration \(T\), the chip duration \(T_c\), the spreading factor \(N\), and the time delay \(\tau_j\), respectively, by a frequency domain spreading code \(X_{f_j}\), a time variable \(t\), a bandwidth \(W\), a chip-bandwidth \(W_c\), a spreading factor in frequency domain \(M = W/W_c\), and a frequency offset \(v_j\), we obtain:

\[
s_j(t) = \sum_{n=0}^{M-1} X_{n,j} W_c \text{sinc}(W_c t) e^{-j2\pi n W_c t},
\]

(7)

\[
r'(t) = \sum_{j=1}^{K} \sum_{p=-\infty}^{\infty} d_{j}^{(i)} s_j(t)e^{-j2\pi (v_j+pW)c t} + n_0(t),
\]

(8)

where the prime sign (‘) is used to express a frequency version of (‘). Such a FD-based CDMA system is illustrated in Fig. 2.

**Remark 1:** Obviously, the frequency spectrum of Eq.(7) is expressed in the same form as the time domain expression of the transmitted signal in DS/CDMA systems. Such a FD-based CDMA system can be regarded as a kind of OFDM system which allows frequency offset \(v_j\), whereas standard OFDM systems do not allow it [6].

An OFDM system is illustrated in Fig. 3, where several data are transmitted in parallel with the same number of subcarriers. The spectrums of adjacent subcarriers are overlapping each other.
Figure 4: A FH/CDMA system. A rectangular baseband signal is modulated with a sinusoidal signal at frequency \( f_i \). In FH systems, only one carrier is used in one hopping duration, whereas in the proposed system, a data signal multiplied by frequency domain spreading codes uses all carriers simultaneously.

**Remark 2:** The proposed system is similar to frequency hopping (FH)/CDMA but different from it in the following sense. In FH/CDMA, each user has a hopping pattern (Fig. 4), which determines a carrier frequency, by which the user’s baseband signal is modulated. The carrier frequency changes at every hopping duration \( T_h \). In a FH system, one user occupies only one carrier within a hopping duration. Therefore, hopping pattern of different users must be disjoint. If two users share the same frequency at the same time, bit error rate increases rapidly. On the other hand, in the FD-based CDMA system, data signal is multiplied by frequency-domain spreading codes, and every user can utilize full of the sub-carriers. Then transmitted signals are overlapping each other, which causes MAI. However, as in DS/CDMA systems, BER does not increase rapidly. Due to the property called ‘graceful degradation’, BER increases gradually.

<table>
<thead>
<tr>
<th>Table 1: Correspondence Table-I</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>rect. wave</strong></td>
</tr>
<tr>
<td><strong>sinc wave</strong></td>
</tr>
<tr>
<td><strong>Gauss wave</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2: Correspondence Table-II</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>no spreading</strong></td>
</tr>
<tr>
<td><strong>random spreading A</strong></td>
</tr>
<tr>
<td><strong>random spreading B</strong></td>
</tr>
</tbody>
</table>

The differences between the proposed method and existing ones are summarised in Table 1 and 2. For simplicity, only typical waveforms are listed in Table 1, although other Nyquist waveforms, e.g. raised cosine, are widely used. Ultra wideband (UWB) based on pulse position modulation (PPM) uses an impulsive Gaussian pulse, or Gaussian monocycle [8]. Spreading codes are categorized into two types. The first type (random spreading A) is the one used in DS/CDMA and FD-based CDMA systems, i.e. it is a sequence of \([+1, -1]\)-valued random variables. The second (random spreading B) is used in FH/CDMA system, i.e. it is a sequence of vectors of the form \((0, 0, 0, 0, 0, 0)\), where the position of ‘1’ is randomly selected. Note that we have given intuitive illustrations in Figs. 1-4 and they were not well-established time-frequency representations, such as spectrogram (the square magnitude of short time Fourier transform) nor the Wigner distribution. We believe such intuitive illustrations are more appropriate to show the differences of the methods.

For frequency domain spreading codes, we have

\[
R^t_{m}(\nu, X', Y') = \sum_{n=0}^{M-1-t} X'_{n} Y'_{n+t} \quad (\nu' \geq 0). \quad (9)
\]

We replace (4) by

\[
v_j - \nu = \left( \ell'_{ij} + \frac{k'_j}{m} \right) W_c. \quad (10)
\]

The system is said to be frequency synchronous if \( \nu'_j = k'_j = 0 \), frequency chip-synchronous if \( \nu'_j \neq 0 \) and \( k'_j = 0 \), and frequency chip-asynchronous if \( k'_j \neq 0 \). Then, the duality of such a system with DS/CDMA implies that the variance of MAI in this system is expressed in the same form of Eq. (5), i.e.,

\[
l'_{ij}^{(i)} = \sum_{p_1=1}^{j} \left( \frac{d_{ij}^{(i)}}{2} + \frac{d_{i+1}^{(i)}}{2} \right) + \frac{1}{m} R_{mM}(\nu'_j, X'_{i,j}, X'_{j,i}) \quad (11)
\]

Therefore the variance of MAI in the dual system is reduced in exactly the same way of chip-asynchronous DS/CDMA systems, except that the time delay \( \tau_i \) is replaced by a frequency offset \( \nu_j \).

Assume that \( d_{ij}^{(i)} (j \neq i) \) are independent on \( d_{ij}^{(i)} \). Then without loss of generality, it suffices to consider the MAI of two-user system. We replace \( X_{up}' \), \( X_{i,j}' \) and \( k'_j \), respectively, by \( X_{up}' \), \( Y_{up}' \), \( \ell'_i \) and \( k' \). As in DS/CDMA case, for a TD-based CDMA we have

**Lemma 1:** For any \( X_{up}' \) and \( Y_{up}' \), the MAI of two-user system satisfies

\[
(l'_{2,2})^2 = \frac{d_{ij}^{(i)}}{2} \frac{d_{i+1}^{(i)}}{2} \frac{1}{m^2} R_{mM}(\nu'_j, X_{up}', Y_{up}')^2 \quad (12)
\]
We have assumed a sinc waveform with rectangular spectrum, which implies that
\[ \frac{1}{m} R_{MM}^{\text{EO}}(t' + k'; X_{up}', X_{up}') = \left( 1 - \frac{k}{m} \right) R_{MM}^{\text{EO}}(t'; X', Y') + \frac{k'}{m} R_{MM}^{\text{EO}}(t' + 1; X', Y'). \] (13)

Let \( D_{ij} \) be a \([-1, 1]\)-valued random variable for \( d_{ij} \). Eqs. (11) and (13) together with the relation \( E[D_{ij}^4 D_{p+1}^4] = 0 \) gives
\[
E_{D_p} \left[ E_{X'(Y') \mid \nu} \left[ \left( \frac{1 - \nu^2}{\sqrt{3} \nu} \right)^2 \right] \right] = \left( 1 - \frac{k'^2}{m} \right) E_r(t') + \frac{k'^2}{m} E_r(t') + 2 \left( 1 - \frac{k}{m} \right) k' \mathcal{F}_r(t'),
\] (14)

where \( E_r(t') = \frac{1}{2} (E^{\text{EO}}(t') + E^{\text{EO}}(t')) \mathcal{F}_r(t') = \frac{1}{2} (F^{\text{EO}}(t') + F^{\text{EO}}(t')) \) and
\[
E^{\text{EO}}(t') = \frac{1}{M} E_{X'Y'}[R_{MM}^{\text{EO}}(t'; X', Y')^2]
\] (15)
\[
F^{\text{EO}}(t') = \frac{1}{M} E_{X'Y'}[R_{MM}^{\text{EO}}(t'; X', Y')]
\] (16)

Remark 3: Let \( K \) be a random variable for \( k' \) taking values in \([0, 1, \ldots, m - 1]\) with its probability \( \Pr[K = k'] = 1/m \). Then for \( m \gg 1 \), we have \( E_K[(1 - K/m)^2] = 1/3, E_K[(1 - K/m)K/m] = 1/6 \). Then,
\[
E_{K} \left[ E_{D_p} \left[ E_{X'(Y') \mid \nu} \left[ \left( \frac{1 - \nu^2}{\sqrt{3} \nu} \right)^2 \right] \right] \right] = \frac{1}{3} (E_r(t') + E_r(t' + 1)) \mathcal{F}_r(t') \quad (m \gg 1).
\] (17)

This implies the variance of MAI is reduced by negative \( \mathcal{F}_r(t') \). This phenomenon is the same as the antithetic variates method in a variance reduction technique [7].

A negative \( \mathcal{F}_r(t') \) can be realized by Markov codes. \( E_r(t') \) and \( \mathcal{F}_r(t') \) are, respectively, the same as \( E(t) \) and \( F(t) \), except that variables are replaced according to Table 1. Thus, the evaluation of them for FD-based CDMA is the same as TD-based CDMA. Assume \( X_0' \rightarrow X_1' \rightarrow \ldots \rightarrow X_{N-1}' \) forms a Markov chain with a state space \( \{+1, -1\} \). Let the eigenvalue of the transition probability matrix of the Markov chain be \(-1 < \lambda < 1\). Then,
\[
E_r(t') = \frac{1 + \lambda^2}{1 - \lambda^2}, \quad \mathcal{F}_r(t') = \frac{2\lambda}{1 - \lambda^2}.
\] (18)

Therefore \( \mathcal{F}_r(t') \) is negative for \( \lambda < 0 \). The optimum \( \lambda \) is \(-2 + \sqrt{3} [3]\).

In ordinary OFDM systems, frequency offset \( v_j \neq 0 \) is not allowed. In such a case,
\[
E_{D_p} \left[ E_{X'(Y') \mid \nu} \left[ \left( \frac{1 - \nu^2}{\sqrt{3} \nu} \right)^2 \right] \right] = \frac{1 + \lambda^2}{1 - \lambda^2} \quad (m = 1).
\] (19)

Hence, \( \lambda = 0 \) is optimum, which implies we cannot reduce the variance of MAI if \( v_j = 0 \).

The phase locked loop (PLL) circuit is a fundamental component of communications, which is needed for a tracking of frequency as well as phase synchronisations. It is desirable if a fine tuning of such synchronizations is not needed. For this purpose, the receiver must be designed to be robust to synchronization errors. We have shown that the variance of MAI of frequency chip-asynchronous system is smaller than that of frequency chip-synchronous one. This implies that frequency synchronization-free CDMA system is promising.

3. Concluding Remarks

This is a preliminary work for investigating the superiority of Markov codes over i.i.d. ones in communication systems. We propose a FD-based CDMA system which can be regarded as a frequency dual of chip-asynchronous DS/CDMA systems. In such a system, the variance of MAI is reduced, if the transmitted signal have a random frequency offset. The MAI becomes even smaller if we employ Markov codes instead of i.i.d. codes. This result strongly suggests that there are still other communication systems which allows both of time and frequency offsets, where Markov codes can make the system more resistant to the synchronization errors.

References

Applications of Recurrence Plots in Road Traffic Analysis

Tomasz Grabowski 1 and Maciej J. Ogorzałek 1,2

1 Department of Electrical Engineering, AGH University of Science and Technology, Krakow, Poland
al. Mickiewicza 30, 30-059 Kraków, Poland
2 Department of Information Technologies, Jagiellonian University, Krakow, Poland
Ul. Reymonta 4, 30-059 Kraków, Poland
Email: tomasz.grabowski@poczta.fm, maciej.ogorzalek@uj.edu.pl

Abstract.
Monitoring of road traffic has been studied for many years. This is because of paramount importance of safety, economy, administration, trade and a lot of another issues. One of the road traffic analysis methods is the online identification – so-called Weight-In-Motion (WIM). Classical application of this method for calculation of selected parameters of cars-in-motion, does not give very interesting results for road traffic flow. We propose to use the recurrence plots method (RP) and recurrence quantification analysis (RQA) tool support which permit deeper insight into the properties of the flow. For the analysis we consider measured time series – electric responses of system to moving vehicles stream. We find several characteristic features of the road traffic and also compare them with properties of selected features found in web and phone intensity traffic.

1. Introduction.
This paper deals with the problem of analysis of road traffic parameters based on measurements. Increasing number of vehicles on the roads world-wide has motivated development of electronic sensing techniques for measuring and analyzing traffic conditions. Huge amount of data are gathered using these techniques. Analysis of these data is not a trivial task. Several methods have been proposed in recent years. Statistical methods rely on analysis of probabilities and the underlying mathematical formula (pattern) is in most cases not known and it may be difficult to understand. Another approach proposed is so-called direct method which relies on a system for counting vehicles and measuring their specific features, such as: velocity, length, total weight, number of axes. Implementation of such a system includes inductive loops and piezoelectric sensors placed under the road surface. Since sometimes results of measurements are not easy to understand visualization methods should be introduced. This article presents a discussion concerning description of the road traffic by the recurrence plots method. Based on time series measured using standard equipment installed on the roads near Cracow we constructed various representations of the measured signals and respective recurrence plots revealing specific patterns and properties.

2. Road Traffic Measurements.
As the number of road vehicles has rapidly increased in Poland in the last few years, estimation of road traffic parameters (e.g. traffic flow rate, time intervals between vehicles, travel times) as well as prediction, recognition and classification of cars becomes essential. One of interesting methods of identification of traffic parameters is so-called Identification (Weighting) In Motion (WIM). The WIM system consists of a group of sensors mounted on the road and a computing system. When a car runs onto detectors electric signals are generated. They are collected and processed by the computing system. In this way we generate the time series which will be used for construction of Recurrence Plots.
In the figure below typical location of detectors on the road is shown. This kind of setup is mounted eg. on the international road E-40 near Cracow. The measurement system includes 5 inductive loops (with different width: 0,1 m – 1,5 m) and 2 piezoelectric sensors. From the first group of sensors we obtain magnetic profiles of cars. The second group delivers supplementary information of specific cars features, such as: velocity, number of axles, total weight and others. Signals derived from them will be discussed in this paper.

Figure 1. Real measurement set-up
1 – 5: inductive loop sensors (suitably wide: 1,5 m; 1 m; 0,5 m; 0,3 m; 0,1 m), 6 – 7: piezoelectric sensors (1,7 meters away)
Road traffic analysis is based on detection of vehicles in motion. This process concentrates either on behavior of each vehicle separately, or a stream of cars.

There are two ways of the road traffic analysis:

a) Frequency of passing vehicles measurement
b) Measurement of time interval between two successive vehicles

An electric signal corresponding to a car crossing a particular detector, is registered in the system as a set of samples.

3. Recurrence Plot.

Natural processes can have a distinct recurrent behavior which can be periodic or irregular. The recurrence of states, understood as the states coming arbitrary close after some time, is a fundamental feature of deterministic dynamical systems. This property has been known in nature for a long time and was introduced by Poincaré in 1890.

The Recurrence Plots (RP) method was suggested in 1987 by J.-P. Eckmann to visualize the recurrences of dynamical systems [2]. This tool is giving us the possibility of the visualization of multidimensional dynamics in the new two-dimensional space. Additional tools (e.g. RQA) allow precise analysis of the structure of the plot. There are several definitions of recurrence plots (RPs) referring to their structure and methods for constructing plots (Contact Map, Dots Plot, Similarity Matrix), in most cases we use the first, most prevailing definition following Eckmann:

\[ R_{ij} = \Theta(r(i) - \|x(i) - x(j)\|) \]  

where: \( i, j = 1,2, \ldots, N; r(i) \) – size of the neighborhood; \( \|\| \) – norm, predefined metric (e.g.: Euclidean, maximal, etc.); \( \Theta(\cdot) \) – Heaviside function

\[ \Theta(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases} \]

The RP is obtained by plotting the recurrence \( N \times N \) matrix (based on equation (1)). The values one and zero in this matrix can be simply visualized by the colors black and white, using different colors for its binary entries, e.g.,plotting a black dot at the coordinates \((i, j)\), if \( R_{ij} = 1 \), and a white dot, if \( R_{ij} = 0 \). Both axes of the RP are time axes.


An elementary way of analysis of car-streams is continuous measurement of signal for many cars passing over the sensors. In this way we can analyze the intensity and specific properties of road traffic. The time series are generated in time unit by the WIM system. One can consider time intervals ranging from minutes to days or even months. The following conditions can influence the results of analysis:

- Location of the road
- Time measurement of signal
- Date of measurement (holidays, Christmas, Easter, weekend, etc., or weekday)
- Calibration of measurement system
- Environmental factors
- Repairs of road under consideration
- Repairs of another roads

Essentially we have 2 means for analysis of frequency and the type of road traffic:

a) measurement of magnetic profiles.
b) measurement of electric length of cars.

As one can notice on Figure 2, the majority of vehicles which were moving on the considered road were small cars. The second clearly appearing group was composed of big lorries. In the middle of those two dominant groups we can find midium-size cars, delivery trucks and partially buses. The same features can be noticed in the histograms in Fig.2b.

Figure 2. Electric length substitute of cars (ESC) in function of time for one hour and one day (a) as well as histograms for those (b).
On the histograms we notice two, clearly separated groups of vehicles. Another division, namely day and night division is also significant. This situation is noticed *ipso facto* only in the case of daily run. The period of reduced vehicles amount is shown by the central interval in the one day histogram of vehicles activity. After the introductory and diagnostic analysis, RP for the different lengths of the time series are considered below.

![Figure 3](image1.png)

**Figure 3.** Recurrence plot of cars-stream for 20 minutes measurement: all samples (a) and considered 1-400 samples (b).

The white lines correspond to separate cars. Thus a large number of such lines gives information about quite large traffic density. The width of line is related to the size of vehicle. The wider the line – the bigger the vehicle. One can notice also that the traffic intensity of the given vehicle class in the time unit is random. Nevertheless, we have to note that a different situation can also occur. For example in one moment a heavy column of vehicles moves on the road after a compulsory stop (for example during the heat or holidays). Such events are still marginal and in the long term they do not change the obtained results in a significant way.

Another method of measurement of the road traffic is based on magnetic profiles. Time series, derived from an induction loop sensor is shown in Figure 4.

![Figure 4](image2.png)

**Figure 4.** Recurrence plot of stream of 20 different cars passing over an inductive loop.

As can be seen in Figure 4, the structure is neither homogeneous nor periodic or stochastic. The pattern on this plot is atypical. It follows from the structure of time series. Every single vehicle has the different shape, width and height. It is being transferred to the thickness and spread of the lines on the graph.

For examining which cars generate what which shapes of magnetic profiles special time series have been constructed. Structure of the time series, containing 20 identical cars, is visible on the next RP-plot. One can see the periodicity and the image consists of parallel strips of the same breadth.

![Figure 5](image3.png)

**Figure 5.** Recurrence plot of a stream of 20 cars. Every 5 cars are separated with the same break.

If we will build the time series from vehicles of different classes (personal, delivery, lorry and buses), then our graph will assume different shapes, of course, the way they showed on the example of the Figures 5 and 6.

![Figure 6](image4.png)

**Figure 6.** Recurrence plot of stream of 20 different vehicles: cars, delivery-vans and one bus.
When measured series are more and more long, then structures which it is possible to observe on recurrence plots will also be changing. They showed will be more dynamics of the entire stream, rather than single vehicles. Figure 7 shows model built from around 200 personal vehicles.

![Figure 7](image)

**Figure 7.** Recurrence plot of stream of 200 personal cars. Every 20 separated vehicles there are the same break of 5000 samples.

When the group of 200 vehicles is divided on 10 identical groups for 20 vehicles and when these groups are separated by identical distance, then we are dealing with the periodicity. We can deal with such a situation in case of heavy traffic on the communications artery, where the cyclical nature of the traffic is a consequence for example of light signals. On the presented plot a length ratio of the break (here: 5000 samples) is essential to the length of the signal describing the vehicle (here: in case of personal cars ca 700 - 800 samples) and lengths of the entire group of 20 vehicles (here: 17680 samples). If the distance between individual groups of vehicles was too small, then the graph would become homogeneous, and distances in a RP-plot would be a unnoticeable. Such situation is shown in Figure 8.

![Figure 8](image)

**Figure 8.** Recurrence plot of stream of 200 personal cars. Every 20 separated vehicles have the same break of 1000 samples of zeroes.

We are not only interested in single vehicles, but also the time intervals between them. This variant, superficially different from previous, is natural fulfillment for its. We can focus one’s attention on the intensity of road traffic. Figure 9 shows specificity of this approach.

![Figure 9](image)

**Figure 9.** RPs of time intervals between vehicles for the time 30 minutes.

RP above are characteristic for random time series and stochastic processes [3]. It is necessary to notice, for time 30 minutes there are microstructures on the plot.

5. **Conclusions and outlook.**

Performed experiments showed, that WIM system delivers a lot of interesting information about road traffic. One can obtain for analysis both data of cars-streams and single vehicles. In contrast to statistic methods RP permits to effectively combine several features for visual inspection and characterization of dynamics of the system. RPs with RQA can be useful for classification of vehicles in motion. Results of interpretations of time series are sometimes difficult, still RPs are interesting aid for analysis cars-stream time series.

**References**

Discrete Gait Generation for the Compass-Type Biped Robot Modeled by Discrete Mechanics

Tatsuya KAI† and Takeshi SHINTANI‡

†Faculty of Information Science and Electrical Engineering, Kyushu University, JAPAN
Email: kai@ees.kyushu-u.ac.jp
‡Graduate School of Engineering, Osaka University, JAPAN
Email: sintoni@watt.mech.eng.osaka-u.ac.jp

Abstract—In this study, we consider a discrete gait generation method for the compass-type biped robot modeled by discrete mechanics. First, we derive the discrete compass-type biped robot model. Next, we define an optimal control problem on discrete gait generation and explain a sequential quadratic programming approach to the problem. A simulation is then illustrated to confirm the effectiveness of our method.

1. Introduction

Recently, a lot of work on humanoid robots have been done. Especially, the compass-type biped robot is one of simplest models of humanoid robots and has been mainly researched so far. In most studies on humanoid robots, the continuous-time mathematical model is dealt with, and Poincaré section approach is usually used to generate stable walking or gait [4, 5]. However, a humanoid robot is known to be one of most difficult mechanical systems to control, and many problems are still unsolved.

On the other hand, discrete mechanics has been focused on as a new discretizing tool for mechanical systems over the last decade [1, 2, 3]. By using discrete mechanics, we can directly obtain a discrete-time model of a mechanical system, and the discrete-time model has various advantages in terms of numerical errors and physical characteristics. We have researched discrete mechanics from the standpoint of control theory and derived some results [6, 7, 8]. In these studies, we mainly treat the cart-pendulum system as a physical example. However, it is expected that discrete mechanics can be applied to more complicated mechanical systems and discrete mechanics approach may add a fresh dimension to control problems for humanoid robots.

In this paper, we apply discrete mechanics to the compass-type biped robot and consider a gait generation problem for the compass-type biped robot formulated by discrete mechanics. This paper is organized as follows. First, some basic concepts on discrete mechanics are summed up in Section 2. In Section 3, we next derive the discrete compass-type biped robot model based on discrete mechanics and explain two modes of the system. Then, a gait generation problem of the discrete compass-type biped robot is formulated, and a solving method of it from the viewpoint of the sequential quadratic programming is developed in Section 4. In Section 5, we then show a numerical simulation to confirm the effectiveness of our method.

2. Discrete Mechanics

This section sums up basic concepts of discrete mechanics [1, 2, 3]. Let \( Q \) be a configuration manifold and \( q \in \mathbb{R}^n \) be a generalized coordinate of \( Q \). We also refer to \( T_qQ \) as the tangent space of \( Q \) at a point \( q \in Q \) and \( \dot{q} \in T_qQ \) denotes a generalized velocity. Moreover, we consider a time-invariant Lagrangian as \( L(q, \dot{q}) : TQ \rightarrow \mathbb{R} \). We first explain about the discretization method. The time variable \( t \in \mathbb{R} \) is discretized as \( t = kh \) \((k = 0, 1, 2, \cdots)\) by using a sampling interval \( h > 0 \). We denote \( \dot{q}_k \) as a point of \( Q \) at the time step \( k \), that is, a curve on \( Q \) in the continuous setting is represented as a sequence of points \( \dot{q}^d := \{q_k\}_{k=1}^N \) in the discrete setting. The transformation method of discrete mechanics is carried out by the replacement:

\[
q \approx (1 - \alpha)q_k + \alpha q_{k+1}, \quad \dot{q} \approx \frac{q_{k+1} - q_k}{h}, \quad (1)
\]

where \( q \) is expressed as an internally dividing point of \( q_k \) and \( q_{k+1} \) with a ratio \( \alpha \) \((0 < \alpha < 1)\). We then define a discrete Lagrangian:

\[
L^d(q_k, q_{k+1}) := hL\left((1 - \alpha)q_k + \alpha q_{k+1}, \frac{q_{k+1} - q_k}{h}\right), \quad (2)
\]

and a discrete action sum:

\[
S^d(q_0, q_1, \cdots, q_N) = \sum_{k=0}^{N-1} L^d(q_k, q_{k+1}). \quad (3)
\]

We next summarize the discrete equations of motion. Consider a variation of points on \( Q \) as \( \delta q_k \in T_qQ \) \((k = 0, 1, \cdots, N)\) with the fixed condition \( \delta q_0 = \delta q_N = 0 \). In analogy with the continuous setting, we define a variation of the discrete action sum (3) as

\[
\delta S^d(q_0, q_1, \cdots, q_N) = \sum_{k=0}^{N-1} \delta L^d(q_k, q_{k+1}). \quad (4)
\]

The discrete Hamilton’s principle states that only a motion which makes the discrete action sum (3) stationary is realized. Calculating (4), we have

\[
\delta S^d = \sum_{k=1}^{N-1} \left[ D_1L^d(q_k, q_{k+1})\delta q_k + D_2L^d(q_{k-1}, q_k)\delta q_k \right]. \quad (5)
\]
where \( D_1 \) and \( D_2 \) denotes the partial differential operators with respect to the first and second arguments, respectively. Consequently, from the discrete Hamilton’s principle and (5), we obtain the discrete Euler-Lagrange equations:

\[
D_1 L^d(q_k, q_{k+1}) + D_2 L^d(q_{k-1}, q_k) = 0, \\
k = 1, \ldots, N - 1.
\]

(6)

It turns out that (6) is represented as difference equations which contains three points \( q_{k-1}, q_k, q_{k+1} \), and we need \( q_0, q_1 \) as initial conditions when we simulate (6).

3. Discrete-time Compass-Type Biped Robot

This section derives a discrete-time model of a compass-type biped robot by using discrete mechanics. In this paper we consider a compass-type biped robot as shown in Fig. 1. Let \( \theta \) and \( \phi \) be the angles of Leg 1 and 2, respectively. We also use the notations: \( m \): the mass of the legs, \( M \): the mass of the waist, \( l \): the length between the waist and the center of gravity, \( b \): the length between the center of gravity and the toe of the leg, \( l (= a + b) \): the length between the waist and the toe of the leg. The Lagrangian of this system is given by

\[
L'(\theta, \phi, \dot{\theta}, \dot{\phi}) = \frac{1}{2}(I + ma^2 + ml^2 + Ml^2) \dot{\theta}^2 + \frac{1}{2}(I + mb^2) \dot{\phi}^2 - mbl \cos(\theta - \phi) \dot{\theta} \dot{\phi} - (ma + mg + M)g \cos \phi + mgb \cos \phi.
\]

(7)

Based on the problem setting above, we now derive the discrete compass-type bipedal robot (DCBR) via discrete mechanics. In general, a model of a compass-type biped robot consists of two modes: the swing phase and the impact phase. As shown in Fig. 2, it is noted that the swing phase and the impact phase occur alternately and the swing phase and the impact phase occur alternately and the swing phase, respectively.

First, we derive the swing phase model of the DCRB for the case where Leg 1 is the swing leg and Leg 2 is the supporting leg as shown in Fig. 1. For the case where Leg 1 is the supporting leg and Leg 2 is the swing leg, we can easily obtain the model by changing \( \theta_{1i}^0 \) for \( \theta_{2i}^0 \). Calculate the discrete Lagrangian \( L_{\alpha}^d \) from (7) as (2) and substitute it into the discrete Euler-Lagrange equations (6). Moreover, adding the control input to the left-hand side of the discrete Euler-Lagrange equations, we obtain the swing phase model as

\[
f_1(\theta_{1i-1}^0, \theta_{1i}^0, \theta_{1i+1}^0, \phi_{1i-1}^0, \phi_{1i}^0, \phi_{1i+1}^0, u_{1i}^0) = 0, \quad \text{(8)}
\]

\[
f_2(\theta_{2i-1}^0, \theta_{2i}^0, \theta_{2i+1}^0, \phi_{2i-1}^0, \phi_{2i}^0, \phi_{2i+1}^0, u_{2i}^0) = 0, \quad \text{(9)}
\]

where functions \( f_1 \) and \( f_2 \) are defined as (10) and (11), respectively.

Next, we consider the impact phase model of the DCRB. In this paper we assume that the swing leg has a completely-elastic collision with the ground surface. Calculating the condition that discrete momentums before and after a collision are equivalent, that is,

\[
D_2 L^d(\theta_{N-1}^0, \phi_{N-1}^0, \phi_{N}^0) = D_1 L^d(\theta_1^{(i+1)}, \theta_2^{(i+1)}, \phi_1^{(i+1)}, \phi_2^{(i+1)}),
\]

\[
D_1 L^d(\theta_{N-1}^0, \phi_{N-1}^0, \phi_{N}^0) = D_2 L^d(\theta_1^{(i+1)}, \theta_2^{(i+1)}, \phi_1^{(i+1)}, \phi_2^{(i+1)}),
\]

Fig. 2: Gait Generation for DCRB

---

- 435 -
\[ f_1 = (\text{ma}^2 + \text{Ml}^2 + \text{ml}^2 + \text{I})(\theta(k+1) + \theta(k)) + mbl(1 - \alpha)\sin((1 - \alpha)(\theta(k+1) - \theta(k))) + mbl\cos((1 - \alpha)(\theta(k+1) - \theta(k))) + (\text{ma} + \text{M} + \text{ml})gh^2(1 - \alpha) \sin((1 - \alpha)\theta(k+1) + \theta(k)) \]
\[ + (\text{ma}^2 + \text{M}^2 + \text{ml}^2 + \text{I})(\theta(k+1) - \theta(k)) + mbl\sin((1 - \alpha)(\theta(k+1) - \theta(k))) + \alpha(\theta(k+1) - \theta(k))\sin(\phi(k+1) - \phi(k)) \]
\[ - \text{mb} \cos((1 - \alpha)(\theta(k+1) - \theta(k))) + \alpha(\theta(k+1) - \theta(k))\sin(\phi(k+1) - \phi(k)) + (\text{ma} + \text{M} + \text{ml})gh^2(1 - \alpha) \sin((1 - \alpha)\theta(k+1) + \theta(k)) \]
\[ + mbl\cos((1 - \alpha)(\theta(k+1) - \theta(k))) + \alpha(\theta(k+1) - \theta(k))\sin(\phi(k+1) - \phi(k)) + mglh^2(1 - \alpha) \sin((1 - \alpha)\theta(k+1) + \theta(k)) \]
\[ + (\text{ma}^2 + \text{M}^2 + \text{ml}^2 + \text{I})(\theta(k+1) - \theta(k)) - \text{mb}(1 - \alpha)\sin((1 - \alpha)(\theta(k+1) - \theta(k))) + \alpha(\theta(k+1) - \theta(k))\sin(\phi(k+1) - \phi(k)) \]
\[ - \text{mb} \cos((1 - \alpha)(\theta(k+1) - \theta(k))) + \alpha(\theta(k+1) - \theta(k))\sin(\phi(k+1) - \phi(k)) + (\text{ma} + \text{M} + \text{ml})gh^2(1 - \alpha) \sin((1 - \alpha)\theta(k+1) + \theta(k)) \]
\[ + (\text{ma} + \text{M} + \text{ml})(\theta(k+1) - \theta(k)) + mbl\sin((1 - \alpha)(\theta(k+1) - \theta(k))) + \alpha(\theta(k+1) - \theta(k))\sin(\phi(k+1) - \phi(k)) \]
\[ - \text{mb} \cos((1 - \alpha)(\theta(k+1) - \theta(k))) + \alpha(\theta(k+1) - \theta(k))\sin(\phi(k+1) - \phi(k)) + mglh^2(1 - \alpha) \sin((1 - \alpha)\theta(k+1) + \theta(k)) \]
\[ + (\text{ma} + \text{M} + \text{ml})(\theta(k+1) - \theta(k)) + mbl\sin((1 - \alpha)(\theta(k+1) - \theta(k))) + \alpha(\theta(k+1) - \theta(k))\sin(\phi(k+1) - \phi(k)) \]
\[ - \text{mb} \cos((1 - \alpha)(\theta(k+1) - \theta(k))) + \alpha(\theta(k+1) - \theta(k))\sin(\phi(k+1) - \phi(k)) + mglh^2(1 - \alpha) \sin((1 - \alpha)\theta(k+1) + \theta(k)) \]
\[ + (\text{ma} + \text{M} + \text{ml})(\theta(k+1) - \theta(k)) + mbl\sin((1 - \alpha)(\theta(k+1) - \theta(k))) + \alpha(\theta(k+1) - \theta(k))\sin(\phi(k+1) - \phi(k)) \]
\[ - \text{mb} \cos((1 - \alpha)(\theta(k+1) - \theta(k))) + \alpha(\theta(k+1) - \theta(k))\sin(\phi(k+1) - \phi(k)) + mglh^2(1 - \alpha) \sin((1 - \alpha)\theta(k+1) + \theta(k)) \]
\[ + (\text{ma} + \text{M} + \text{ml})(\theta(k+1) - \theta(k)) + mbl\sin((1 - \alpha)(\theta(k+1) - \theta(k))) + \alpha(\theta(k+1) - \theta(k))\sin(\phi(k+1) - \phi(k)) \]
\[ - \text{mb} \cos((1 - \alpha)(\theta(k+1) - \theta(k))) + \alpha(\theta(k+1) - \theta(k))\sin(\phi(k+1) - \phi(k)) + mglh^2(1 - \alpha) \sin((1 - \alpha)\theta(k+1) + \theta(k)) \]}

we have
\[ h_1(\theta_N, \theta_{N-1}, \theta_{N-2}, \theta_{N-3}, \phi_{N}, \phi_{N-1}, \phi_{N-2}, \phi_{N-3}) = 0, \quad (12) \]
\[ h_2(\theta_{N-1}, \theta_{N-2}, \theta_{N-3}, \phi_{N}, \phi_{N-1}, \phi_{N-2}, \phi_{N-3}) = 0, \quad (13) \]
where functions \( h_1 \) and \( h_2 \) are defined as (14) and (15), respectively. Moreover, in the impact phase, the swing leg and the supporting leg replace each other, and this can be realized by the next equation:
\[ \phi_{i+1} = -\theta_{i}, \quad \phi_{i+1} = -\phi_{i}. \quad (16) \]

Therefore, the impact phase model consists of (8)–(16).

4. Discrete-time Gait Generation for DCBR

In this section we consider a gait generation problem for the DCBR derived in the previous section and we propose the new concept ‘discrete gait.’ We here deal with the following problem.

**Problem 1**: For the discrete compass-type biped robot (8)–(16), find a control input that generates a stable discrete gait.

The purpose of this section is to obtain a control input solving Problem 1. We now formulate Problem 1 as an optimal control problem whose objective function is a sum of the square of a control input. For the DCBR in the i-th swing phase, an optimal control problem can be formulated as follows:
\[ \min J = \sum_{k=0}^{N-1} [u_k^2] \]
\[ \text{s.t.} \quad (10), \quad (11) \]
\[ -[(1 - \alpha)\theta_{i+1} + \alpha\phi_{i+1}] + [(1 - \alpha)\phi_{i} + \alpha\theta_{i+1}] < 0 \quad (18) \]
\[ \theta_{i+1} = -\theta_i, \quad \phi_{i+1} = -\phi_i. \quad (19) \]

In the problem above, (18) represents a constraint on the vertical lengths of Leg 1 and 2, and (19) is a boundary condition in order to generate a stable discrete gait. In the impact phase between the i-th and (i+1)-th swing phases, we can calculate initial states of the (i+1)-th swing phases: \( \theta_{i+1}, \phi_{i+1}, \phi_{i+1}, \phi_{i+1} \) from (12)–(16).

The optimal control problem formulated above can be considered as a finite dimensional constrained nonlinear optimization problem with respect to the \( (3N - 1) \) variables \( \theta_{i}, \phi_{i}, \theta_{N}, \phi_{N}, \theta_{N-1}, \phi_{N-1}, \cdots, \theta_{1}, \phi_{1} \). Therefore, we can solve it by using the sequential quadratic programming approach and so on [3, 9].

5. Numerical Simulation

In this section we perform a numerical simulation on gait generation for the DCBR based on the method proposed in the previous section, and check the availability of our method.

First, we set parameters as follows; parameters on gait generation: \( N = 10 \), \( L = 3 \), parameters on the DCBR: \( m = 2.0 \) [kg], \( M = 10.0 \) [kg], \( I = 1.0 \) [kgm²], \( \alpha = 0.5 \) [m], \( b = 0.5 \) [m], \( l = 1.0 \) [m], \( \alpha = 1/2 \), initial states of the DCBR: \( \theta_1 = \pi/12, \phi_1 = \pi/12 \). In order to solve the optimal control problem (17)–(19), we use the sequential quadratic programming method [9].

Fig. 3–5 show the simulation results. In Fig 3, time series plots of the leg 1 and 2 (\( \theta \) and \( \phi \)) are illustrated. Fig. 4 shows the phase space of \( \theta - \phi \). In Fig. 5, a snapshot of a discrete gait is depicted. From these results, it can be confirmed that our approach can generate a stable gait for the DCBR.
6. Conclusion

In this paper we have considered a discrete gait generation problem for the discrete compass-type biped robot (DCBR) and developed a solving method for the problem from the viewpoint of the reduction to a finite dimensional optimization problem and the sequential quadratic programming method. Simulation results have indicated the effectiveness of our approach.

Our future work are as follows: discrete gait generation of the DCBR in various environments such as slopes and stairs, a transformation method of discrete-time inputs into continuous-time inputs and control of the normal compass-type biped robot, experimental validation for the normal compass-type biped robot.

References


Attitude Stabilization Control of 3D Space Robot Model with Initial Angular Momentum via Model Predictive Control

Tatsuya KAI† and Katsunori KONDOH‡

†Faculty of Information Science and Electrical Engineering, Kyushu University, JAPAN
Email: kai@ees.kyushu-u.ac.jp
‡Graduate School of Informatics, Kyoto University, JAPAN
Email: kondoh@watt.mech.eng.osaka-u.ac.jp

Abstract—This study considers attitude control of a 3D space robot of two rigid bodies with initial angular momentum. First, we explain the universal joint model with initial angular momentum. We next apply model predictive control to an attitude control problem of the universal joint model with initial angular momentum, and show a simulation result to confirm reduction of calculation amount. Then, a simulation with a model error of initial angular momentum is performed in order to check robustness of model predictive control.

1. Introduction

It is well known that for a space robot in 3-dimensional outer space, its conversation law of total angular momentum plays a role of nonholonomic constraints, and hence the robot’s attitude can be changed by transforming its shape. A lot of researches on such a space robot have been done in the fields of analytic mechanics, control theory and robotics [1, 2, 3]. In most researches on control of space robots, it is assumed that space robots do not have initial angular momentum. In realistic situations, for example, when a mother ship gives a space robot out, space robots often have initial angular momentum. Hence we have focused on 3D space robots with initial angular momentum and derived a control strategy based on the near-optimal control method [5]. However, since the model of a space robot with initial angular momentum is quite complicated and the proposed control law is feedforward-type, a huge quantities of calculation amount is needed. Moreover, the control law does not have the characteristic of robustness for the physical parameters of the system. The purpose of this study is to overcome the disadvantages mentioned above by using model predictive control that consists of feedback-type control laws.

2. 3D Space Robot with Initial Angular Momentum

First, the 3D space robot model treated throughout this paper is explained. We consider a space robot that consists of two rigid bodies and exists in 3D space as shown in Fig. 1. Two rigid bodies (Rigid Body 1 and 2) are connected by a universal joint via two links (Link 1 and 2), respectively. We represent coordinates of the inertial space, Rigid Body 1 and 2 by $C_0$, $C_1$ and $C_2$, respectively. We now assume that the origins of $C_1$ and $C_2$ correspond to the centroids of Rigid Body 1 and 2, respectively. Let $A_i \in SO(3)$ be the attitude of Rigid Body $i$ ($i = 1, 2$) with respect to the inertial space $C_0$, and $w_i \in \mathbb{R}^3$ be the angular velocity of Rigid Body $i$. Note that $\dot{v_i} = A_i^T \dot{A}_i$ holds. We use the notations; $m_i$: the mass of Rigid Body $i$ ($m = m_1 + m_2$), $l_i$: the length of Link $i$, $s_i = [00l_i]^T \in \mathbb{R}^3$: the vector showing the position of the joint with respect to $C_0$, $I_i \in \mathbb{R}^{3 \times 3}$: the inertia tensor of Rigid Body $i$ ($J_1 = I_1 + \epsilon \delta_1 \delta_1$, $J_{12} = \epsilon \delta_1 A_1^T A_2 \delta_2$), Next, we denote the angles of Link 1 and 2 of the universal joint as $\theta_1, \theta_2 \in \mathbb{R}$ ($\theta = [\theta_1 \theta_2]^T \in \mathbb{R}^2$), respectively. Then, $A_i := A_i^T A_2 = \begin{bmatrix} \sin \theta_1 \sin \theta_2 & \cos \theta_1 & -\sin \theta_1 \cos \theta_2 \\ \cos \theta_2 & 0 & \sin \theta_2 \\ \cos \theta_1 \sin \theta_2 & -\sin \theta_1 & -\cos \theta_1 \cos \theta_2 \end{bmatrix}$ represents the shape of the space robot and $w_2 = A_i^T w_1 + w$ holds for the angular velocity of the joint $w \in \mathbb{R}^3$, $\dot{w} = A_i^T \dot{A}_i$. Assuming that the space robot has initial angular momentum $P_0 \in \mathbb{R}^3$, we have the conservation law of the total angular momentum as

\[
(A_1 J_1 + A_2 J_{12})w_1 + (A_2 J_2 + A_1 J_{12})w_2 = P_0. \tag{1}
\]

Now, we set $I_o := I_1 + A_1 A_1^T + A_1 J_{12} + J_{12} A_1^T$ and parametrize $A_1$ by using the Cayley-Rodrigues parameter (3). Note that

\[
w_1 = U_1(a) \alpha, \quad U_1(a) = \frac{2(I - \alpha \alpha^T)}{1 + a^T a} \tag{2}
\]

holds for the angular velocity $w_1$ and Cayley-Rodrigues parameter $a$. Moreover, we refer angular velocities of the universal joint as control inputs, that is, $u_1 := \theta_1, u_2 := \theta_2$, then we have the next:

\[
w = \begin{bmatrix} \cos \theta_2 & 0 \\ \sin \theta_2 & 0 \end{bmatrix} u_1 + \begin{bmatrix} 0 & 1 \\ b_1 & b_2 \end{bmatrix} u_2. \tag{4}
\]

Therefore, setting $q := [\theta^T a^T]^T \in \mathbb{R}^3$, from (1)–(4) we obtain the universal joint model with initial angular momentum as (5), which is represented as a nonlinear affine map

\[
q = T v + \delta, \quad T = \begin{bmatrix} 0 & -v_1 & v_2 \\ 0 & v_1 & v_2 \\ -v_2 & v_1 & 0 \end{bmatrix}, \quad \delta = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}
\]

$T$ is the operator that changes a 3-dimensional vector $v = [v_1 v_2 v_3]^T \in \mathbb{R}^3$ into a $3 \times 3$ skew-symmetric matrix: $T = \begin{bmatrix} 0 & -v_1 & v_2 \\ 0 & v_1 & v_2 \\ -v_2 & v_1 & 0 \end{bmatrix}$.
control system with 5 states and 2 inputs and does not have any equilibrium points. For the universal joint model (5), we have shown that (5) is strongly locally accessible at any state, and if control inputs are sufficiently large, (5) is small-time locally controllable [5].

\[
A_1(\alpha) = \frac{1}{1 + \|\alpha\|^2} \begin{bmatrix}
1 + \alpha_1^2 - \alpha_2^2 - \alpha_3^2 & 2(\alpha_1\alpha_2 - \alpha_3) & 2(\alpha_1\alpha_3 + \alpha_2) \\
2(\alpha_1\alpha_2 + \alpha_3) & 1 - \alpha_2^2 + \alpha_3^2 & 2(\alpha_2\alpha_3 - \alpha_1) \\
2(\alpha_1\alpha_3 - \alpha_2) & 2(\alpha_2\alpha_3 + \alpha_1) & 1 - \alpha_1^2 - \alpha_2^2 + \alpha_3^2
\end{bmatrix}
\] (3)

\[
\begin{bmatrix}
\dot{\theta}_1 \\
\dot{\theta}_2 \\
\dot{\alpha}
\end{bmatrix} =
\begin{bmatrix}
0 & 0 & -U_1^{-1}I_u^1A_1^TP_0 \\
0 & 1 & 0 \\
U_1^{-1}I_u^1(AJ_2 + J_{12})b_1 & -U_1^{-1}I_u^1(AJ_2 + J_{12})b_2 & 0
\end{bmatrix}
\begin{bmatrix}
u_1 \\
u_2 \\
\end{bmatrix}
\] (5)

\[
J = \frac{1}{2} \int_0^{e^{T(t)}} Q(\alpha(\tau) - \alpha_d)^TQ(\alpha(\tau) - \alpha_d)d\tau + \frac{1}{2} \int_0^{e^{T(t)}} u(\tau)^TRu(\tau)d\tau + \frac{1}{2}(\alpha(t + T) - \alpha_d)^TS(\alpha(t + T) - \alpha_d)
\] (6)

3. Attitude Stabilization Control

Since the universal joint model with initial angular momentum (5) does not have any equilibrium points and cannot stand still, we cannot treat normal control problems such as a stabilization problem to the origin. Therefore, this section considers the following control problem. Problem 1 contains, for example, the situation where we move a solar panel of a space robot to the direction of the sun.

Problem 1: For the universal joint model with initial angular momentum (5), find control inputs such that the attitude of Rigid Body 1 $\alpha$ is stabilized to a desired value $\alpha_d$.

In this paper, we take the model predictive control approach in order to solve Problem 1. Especially, we use the C/GMRES method [4], which is a real-time optimization algorithm. In a simulation, we use the parameters of the universal joint model: $l_1 = l_2 = 1$, $m_1 = m_2 = 1$, $l_1 = l_2 = \text{diag}[1/2, 1/2, 1]$, initial angular momentum: $P_0 = [0.1 \ 0.1 \ -0.1 \ 1]^T$, the initial state: $q_0 = [\pi/2 \ \pi/2 \ 1 \ 1 \ 1]^T$, the desired attitude: $\alpha_d = [0 \ 0 \ 0]^T$.

For the C/GMRES method, we use the evaluation function (6) with the weight matrices $Q = \text{diag}[4.0, 1.5, 5.0]$, $R = \text{diag}[0.01, 0.01]$, $S = \text{diag}[0.8, 0.2, 0.4]$ and the evaluation interval $T(t) = T(1 - e^{-\omega t})$, $T = 6.5$, $\omega = 0.05$. Moreover, we also use the parameters of controller: the division number of the evaluation interval: $N = 50$, the stabilization parameter of the continuation method: $\zeta = 20$, the number of iterations of the GMRES method: $\zeta_{\text{max}} = 3$, the sampling time: $\Delta t = 0.05$ [s], the simulation time: 20 [s].

Simulation results are shown in Fig. 2 and 3. Fig. 2 illustrates the time series of $\theta$ and $\alpha$, and Fig. 3 depicts the snapshot of the universal joint model. From these results, it can be confirmed that the attitude of Rigid Body 1 $\alpha$ is stabilized to the desired value $\alpha_d = 0$. The computation time of this simulation is 1.45 [s], and hence we can see that the computation time is drastically reduced in comparison with the case of the near-optimal control method [5].

4. Robustness for Initial Angular Momentum

In parameters of a space robot, the mass, the inertia moment and the length can be easily measured. However, since the value of initial angular momentum changes according to circumstances, we have the difficulty to measure it. So this section verifies the availability of the model predictive control approach in the case where there exists a modeling error in initial angular momentum.

In a simulation, we use the same parameters of the universal joint model except initial angular momentum as the ones shown in Section 3. We set the measured initial angular momentum: $\tilde{P}_0 = [0.1 \ 0.1 \ -0.1 \ 1]^T$ and the real initial angular momentum: $P_0 = [0.07 \ 0.07 \ -0.01 \ 1]^T$. We also use the weight matrices of the evaluation function (6) as $Q = \text{diag}[2.0, 1.0, 3.0]$, $R = \text{diag}[0.01, 0.01]$, $S = \text{diag}[0.85, 0.2, 0.4]$. Moreover, we also use the same parameters of controller as the ones shown in Section 3.

Fig. 4 shows the time series of $\theta$ and $\alpha$, and Fig. 5 illustrates the snapshot of the universal joint model. From these results, it turns out that the attitude of Rigid Body 1 $\alpha$ is stabilized to the desired value $\alpha_d = 0$ despite model error, and hence the controller obtained by the model predictive control approach has robustness for initial angular momentum.
5. Conclusion

In this paper, we have considered attitude stabilization control of the universal joint model with initial angular momentum via model predictive control approach. Simulation results have indicated that the attitude of Rigid Body 1 is stabilized to the desired value with a reduced calculation amount compared to our previous work, and robustness with respect to initial angular momentum can be confirmed.

References

Fig. 4 Time Series of $\theta$ and $\alpha$ (with Model Error of Initial Angular Momentum)

Fig. 5 Snapshot of Universal Joint Model (with Model Error of Initial Angular Momentum)


Adaptive Stabilization of a Saddle Steady State of a Conservative Dynamical System: a Spacecraft at the Lagrange Point L2 of the Sun–Earth System

Elena Tamaševičiūtė,1 Gytis Mykolaitis,2 Skaidra Bumelienė,3 and Arūnas Tamaševičius3

1Department of Mechanical and Process Engineering, Swiss Federal Institute of Technology in Zürich (ETHZ), CH–8092 Zürich, Switzerland
2Department of Physics, Vilnius Gediminas Technical University, LT–10223 Vilnius, Lithuania
3Semiconductor Physics Institute, Center for Physical Sciences and Technology, LT–01108 Vilnius, Lithuania
Email: elenata@student.ethz.ch, gytis@pfi.lt, skaidra@pfi.lt, tamasev@pfi.lt

Abstract—An adaptive feedback method for tracking and stabilizing unknown or slowly varying saddle type steady states of conservative dynamical systems is described. We demonstrate that a conservative saddle point can be stabilized with neither unstable nor stable filter technique. Meanwhile, a controller, involving both filters in parallel, works perfectly. As a specific example, the Lagrange point L2 of the Sun–Earth system is discussed and a general second order saddle model is considered.

1. Introduction

In astrodynamics the stability of the spacecrafts at the Lagrange points L1 and L2, also in related to them the Lyapunov and the Lissajous orbits is a very challenging problem. For example, several recent space missions use the Lissajous orbit around the L2 of the Sun–Earth system. Specifically, the Lagrange points belong to the wide class of the saddle type unstable steady states (USS) of conservative dynamical systems. A straightforward idea to stabilize an USS is to apply a proportional feedback force. Classical control methods, using proportional feedback, require a mathematical model of a system or at least the location of the USS in the phase space for the reference point. However, in many practical situations neither the exact models nor the coordinates of the reference point are accessible. Moreover, the position of a reference point may slowly vary with time in an unpredictable manner because of external perturbations. Therefore adaptive, that is model-independent and reference-free methods, automatically locating the USS are preferable.

The simplest adaptive technique for stabilizing the USS is based on the derivative controller [1, 2, 3]. A feedback force in the form of a derivative $kd\dot{x}/dt$ derived from an observable $x(t)$ introduces artificial dynamical losses and consequently damps the oscillations. An important feature of the derivative is that it does not influence the position of the USS, since it vanishes when the variable $x(t)$ approaches the goal state. The method has been successfully applied to diverse nonlinear dynamical systems, for example to stabilize a laser [1], a chaotic Chua circuit [2] and an electro-chemical reaction [3]. However, the technique is rather sensitive to high-frequency noise unavoidably present in the experimental signal $x(t)$, since it requires a differentiation of the observable.

More advanced adaptive method for stabilizing the USS employs low- (high-) pass filter in the feedback loop [4, 5, 6, 7, 8, 9]. Provided the cut-off frequency of the filter is low enough, the filtered image $\tilde{x}(t)$ of the observable $x(t)$ asymptotically approaches the USS and therefore can be used as a reference point in the proportional feedback. This method has been verified in several experimental systems, including electrical circuits [4, 5, 6, 7] and lasers [8, 9].

Two more techniques, though originally designed to control unstable periodic orbits, can be used to stabilize the USS as well. The first is the time-delayed feedback method proposed by Pyragas [10, 11]. Under appropriate choice of the delay value the method is able to stabilize the steady states [5, 6, 12]. The second one is the notch-filter method [13]. Though developed to stabilize periodic orbits, it is also capable to control the steady states in the case at least two notch filters with different and incommensurate resonance frequencies are applied.

However, all the mentioned techniques, as well as the recently suggested modification with the Taylor predictor [14] and extension to strongly nonlinear regions [15], are restricted to unstable nodes and unstable spirals only. They fail to stabilize the saddle type states (the USS with an odd number of real positive eigenvalues).

To get around the problem of the odd number limitation Pyragas et al. [16, 17] proposed to use an unstable filter, that is a bold idea to fight one instability with another instability. The technique has been demonstrated to stabilize saddles in several mathematical models [16, 17, 18, 19] also in the experiments with
an electrochemical oscillator [16, 17] and the Duffing–Holmes type electronic circuit [18, 19]. Unfortunately, this advanced method is limited to dissipative dynamical systems only. It is not applicable to conservative systems. The situation is somewhat similar to the problem of the famous OGY method [20, 21], in the sense that it does not work in the Hamiltonian systems. The limitation of the Pyragas’ unstable filter method can be proved analytically using the well-known Hurwitz stability criteria. According to these criteria the necessary condition for stabilizing a saddle state is that the cut-off frequency of the applied unstable filter is lower than the damping coefficient of the system under control. While damping is zero (!) in the conservative systems under definition. Formally, to fulfill this stability criteria, the cut-off frequency could be set negative. However, this would mean that the unstable filter should become a stable one and, therefore, inappropriate to stabilize a saddle.

In this paper, we demonstrate that the conservative saddles can be successfully stabilized by means of the recently proposed conjoint filter [22], that involves both an unstable and stable subfilters. Previously such a combined filter has been employed to overcome the problem of latencies in the feedback loop when stabilizing saddles in the dissipative systems.

2. Lagrange point L2

We consider dynamics of a body of mass \( \mu \), e.g. a spacecraft at the Lagrange point L2 of the Sun–Earth system (Fig. 1). The dynamics taking into account the centrifugal force and the forces of gravity is given by

\[
\ddot{r} - \Omega^2 F(r, \xi) = 0, \tag{2}
\]

\[
F(r, \xi) = 1 + r - \frac{1}{(1 + r)^2} - \frac{\epsilon}{\Omega^2} + \xi, \tag{3}
\]

\[
\xi = \frac{P}{\mu R_0 \Omega^2}. \tag{4}
\]

The nonlinear function \( F(r, \xi) \) is depicted in Fig. 2. The steady state of the system \( r_0 \) can be found from an algebraic equation \( F(r_0, \xi) = 0 \). The value of \( r_0 \) can be roughly estimated from a simple formula \( r_0 \approx (\epsilon/3)^{1/3} \), which is valid for \( (r_0, \xi) \ll 1 \). Linearization of the system around the steady state point \( (r = r_0 + x, |x| \ll r_0) \) yields:

\[
\ddot{x} - \Lambda^2 x = 0, \quad \Lambda^2 = \Omega^2 F'(r_0, \xi). \tag{5}
\]

The insert shows that the force \( F(r) \) is nearly linear in the range from 0.009 to 0.011, that is around the Lagrange point \( r_0 = 0.01 \pm 0.001 = 0.01 \pm 10% \).

\[
\ddot{x} - \Lambda^2 x = 0. \tag{6}
\]

Figure 1: Lagrange point L2 of the Sun–Earth system. The Sun diameter, the Earth diameter, the distances \( R_0 \) and \( R \) are not in scale.

Figure 2: Nonlinear function \( F(r) \) with \( \epsilon = 3.10^{-6} \).
3. Stabilizing a body at the Lagrange point

First of all we apply the unstable filter [16, 17] trying to stabilize the saddle point:

\[ \ddot{x} - x = k_1(u - x), \quad (7) \]
\[ \dot{u} = \omega_1(u - x). \quad (8) \]

The corresponding characteristic equation is

\[ \lambda^3 - \omega_1 \lambda^2 + (k_1 - 1)\lambda + \omega_1 = 0. \quad (9) \]

There is a considerable drop of the largest \( \text{Re}\lambda \) with \( k_1 \), however it remains positive indicating instability of the closed loop (Fig. 3a).

Though the stable filter technique is not expected to stabilize a saddle steady state, we consider it here for comparison:

\[ \ddot{x} - x = k_2(v - x), \quad (10) \]
\[ \dot{v} = \omega_2(x - v). \quad (11) \]

From its characteristic equation

\[ \lambda^3 + \omega_2 \lambda^2 + (k_2 - 1)\lambda - \omega_2 = 0. \quad (12) \]

one can make sure that the result is practically the same (Fig. 3b) as for the unstable filter. The controller fails to stabilize the saddle.

However, when combined in parallel:

\[ \ddot{x} - x = k_1(u - x) + k_2(v - x), \quad (13) \]
\[ \dot{u} = \omega_1(u - x), \quad (14) \]
\[ \dot{v} = \omega_2(x - v), \quad (15) \]

the two filters give unexpectedly excellent result as evident from the solution of the characteristic equation

\[ \lambda^4 + (\omega_2 - \omega_1)\lambda^3 - (k_1 + k_2 - 1 - \omega_1 \omega_2)\lambda^2 + [k_1 \omega_2 - k_2 \omega_1 - (\omega_2 - \omega_1)]\lambda + \omega_1 \omega_2 = 0. \quad (16) \]

Indeed, the largest eigenvalue crosses zero at a certain value of the the feedback coefficient \( k_1 \) (Fig. 3c). The stability properties can be also checked using the Hurwitz criteria.

4. Concluding remarks

We have suggested using an adaptive control method for stabilizing unknown and slowly varying saddle type steady states of conservative dynamical systems. The controller is model-independent and reference-free. It does not require knowledge of either the mathematical model or the position of the steady state, but automatically tracks the state and stabilizes it. The controller involves both, the unstable filter and the stable filter in the feedback loop. While separately each of the filters seems to be useless and senseless in the case of conservative dynamical systems, when combined in one they give an excellent stabilizing result. In the nearest future we going to construct an undamped electronic circuit, imitating the dynamical behaviour of a body at the Lagrange point of the conservative Sun–Earth system. The experimental results will be published elsewhere.

Acknowledgment

The work was partially supported by the EU Structural Funds via the Research Council of Lithuania.
within the Operational Programme for Human Resources Development for 2007-2013 under the measure VP1–3.1–SMM–07–V "Support to Research of Scientists and Other Researchers (Global Grant)".

References


Suppression of spatio-temporal chaos in excitable media with nonexcitable cells

Masaki Irisawa, Keiji Konishi†, Naoyuki Hara and Hideki Kokame

Osaka Prefecture University
1-1 Gakuen-cho, Naka-ku, Sakai, Osaka 599-8531 JAPAN
† Email: konishi@eis.osakafu-u.ac.jp

Abstract—Spiral waves and spatio-temporal chaos are the main causes of a serious cardiac arrhythmia. The cardiac muscle has characteristic features of excitable media. It is well known that the fibrosis of myocardium, sets of non-excitile cells, is strongly correlated with an incidence of serious arrhythmias. The present paper examines the suppression of spiral waves and spatio-temporal chaos in excitable media with fibrosis. The boundary periodic pacing is applied to the excitable media. The influence of non-excitile cells on the suppression performance is investigated on numerical simulations. These numerical investigations suggest that the angular frequency of the boundary periodic pacing should be low for the high fibrosis ratio.

1. Introduction

Sudden death is mainly caused by fatal arrhythmias such as the ventricular tachycardia and the ventricular fibrillation. In order to eliminate the ventricular fibrillation, a high-voltage electric shock is applied to a patient. Although this method succeeds in suppressing the fibrillation, it may induce burns and aftereffects to a patient. In recent years, the advanced defibrillation scheme with a low-voltage electric shock is required to avoid such problems.

It is well accepted that the ventricular tachycardia and the ventricular fibrillation would be caused by electrical spiral waves and spatio-temporal chaos occurred in cardiac tissue [1]. As the tissue can be considered as excitable media, the elimination of spiral waves and spatio-temporal chaos in mathematical models of excitable media corresponds to a defibrillation.

There has been growing recognition that such elimination would be a key phenomenon for medical care. It is therefore necessary to find effective methods for the elimination. So far, many researchers have proposed the various methods for eliminating spiral waves and spatio-temporal chaos in mathematical models of excitable media. These are classified into the global and non-global methods: for the global methods, an input signal is applied to the whole media [2, 3, 4]; on the contrary, for the non-global methods, it is applied to some parts of the media [5, 6, 7, 8, 9].

It is generally known that the real cardiac muscle is not homogeneous: the real muscle includes the fibrotic tissue which consists of non-excitile cells. In healthy hearts, the percentage of fibrotic tissue makes up only 5% of the total tissue. During aging and in cardiac diseases, the percentage may increase up to 35% [10]. An increased amount of fibrotic tissue is strongly correlated with an increased incidence of serious arrhythmias [11, 12, 13]. Therefore, the presence of fibrosis should be taken into consideration in developing the defibrillation scheme. Recently, the excitable media with fibrosis are described by the mathematical models: the effect of diffuse fibrosis to the wave propagation has been investigated [14, 15, 16].

The purpose of this work is to investigate the elimination of spiral waves and spatio-temporal chaos in the excitable media with fibrosis. It is shown that the spiral waves and spatio-temporal chaos can be suppressed by applying periodic pacing on media’s boundary. This method does not require the measurements and the feedback control mechanism; thus, it would be easily realized in practical situations, such as the implantable cardioverter-defibrillator. The influence of fibrosis on the suppression performance is investigated by numerical simulations.

2. Excitable media with nonexcitable cells

The excitable media can be described by simple mathematical models. The present paper employs a two-dimensional Bär model [17]:

$$\frac{\partial u}{\partial t} = -\frac{1}{\varepsilon} u(u-1) \left( u - \frac{v + b}{a} \right) + \nabla^2 u, \quad (1a)$$

$$\frac{\partial v}{\partial t} = f(u) - v, \quad (1b)$$

$$f(u) = \begin{cases} 0 & u < \frac{1}{4} \\ 1 - 6.75u(u-1)^2 & \frac{1}{4} \leq u \leq 1 \\ 1 & u > 1 \end{cases} \quad (2)$$

This work was partially supported by KAKENHI(20560425).
Figure 1: Initial condition and development into spiral waves ($\varepsilon = 0.06$): (a) initial condition; (b)-(c) spiral waves.

Here $\nabla^2 := \partial^2 / \partial x^2 + \partial^2 / \partial y^2$ is the Laplace operator. $u \in \mathbb{R}$ and $v \in \mathbb{R}$ are the activator and inhibitor variables, respectively. The parameters $a$ and $b$ are related to the excitation threshold, and they are fixed at $a = 0.84$, $b = 0.07$ throughout this paper. As the parameter $\varepsilon$ is a small positive value, $u$ is considered as a fast variable compared with the slow variable $v$. As shown in Fig. 1, the suitable initial conditions lead to a stable spiral wave and a meandering spiral for $\varepsilon < 0.069$. The spiral wave breaks up and then spatio-temporal chaos occurs for $\varepsilon > 0.069$. In our numerical simulations, the time step $\Delta t = 0.002$ and the space step $\Delta x = \Delta y = 0.1$ are used. Further, model (1) has the cell size, $x \in [0, 400]$ and $y \in [0, 400]$, with the no-flux boundary.

The diffuse fibrosis in cardiac tissue was modeled by the presence of nonexcitable obstacles of size $0.1 \times 0.1$. These were randomly distributed over the medium, which mainly expresses the aging fibrosis [14]. The ratio of nonexcitable obstacles corresponds to that of fibrosis.

To suppress spiral waves and spatio-temporal chaos, the periodic pacing is applied to the media’s boundary [8]. Equation (1a) is modified as

$$\frac{\partial u}{\partial t} = -\frac{1}{\varepsilon} u(u-1) \left( u - \frac{v+b}{a} \right) + \nabla^2 u + F(x,t).$$

The input signal for elimination, $F(x,t)$, is given by

$$F(x,t) = \begin{cases} \frac{15 \delta(x) \cos \omega t}{\varepsilon} & t \in [0, 500] \\ 0 & \text{otherwise} \end{cases},$$

where $\omega$ is the angular frequency. The delta function $\delta(x)$ is 1 only at the boundary $x = 0$. The initial condition (i.e., medium state at $t = 0$) is set to a spiral wave or spatio-temporal chaos. The input signal is applied for the period $t \in [0, 500]$. We judge the successful elimination by the fact that the entire medium settles into the rested state by $t = 550$.

3. Numerical simulations

Figure 2 shows the time-space patterns for $\varepsilon = 0.08$ and $\omega = 1.45$. The shaded area represents the excited state ($u \geq 1/3$) and the white area represents the rested state ($u < 1/3$). The suppression occurs with 0% fibrosis as shown in Figs. 2(a)-(d); however, it does not occur with 20% fibrosis (see Figs. 2(e)-(h)). For $\varepsilon = 0.08$ and $\omega = 1.20$, the suppression does not occur with 0% fibrosis as shown in Figs. 3(a)-(d); in contrast, it occurs with 20% fibrosis (see Figs. 3(e)-(h)). From these results, we notice that the suppression depends on the angular frequency $\omega$ and the fibrosis ratio.
It was reported that, without fibrosis (i.e., 0%), the controllable angular frequency $\omega$ strongly depends on the parameter $\varepsilon$ [8]. The present paper shall investigate the influence of the fibrosis ratio on the controllable angular frequency $\omega$. Figure 4 indicates the controllable angular frequency in $\varepsilon$-$\omega$ plane for fibrosis ratios 0, 10, 20%. The upper (lower) bound of the controllable angular frequency are denoted by $\omega_{\text{up}}$ ($\omega_{\text{low}}$). The suppression occurs when $\omega$ is set to $\omega \in [\omega_{\text{low}}, \omega_{\text{up}}]$. Here, $\Delta \omega$ is called controllable range. The ranges $\Delta \omega$ for the ratios 0% and 10% are almost the same. On the other hand, for the ratio 20%, $\omega_{\text{up}}$ decreases greatly and $\omega_{\text{low}}$ decreases slightly with an increase in the fibrosis ratio. Further, it is observed that, for the ratio 30%, input signal (4) cannot suppress the spiral waves and the spatio-temporal chaos (i.e., $\Delta \omega = 0$). From these results, we notice that input signal (4) is effective to the excitable medium with fibrosis up to about 20%.

4. Discussion

This section considers the reasons why the upper and lower bounds of the controllable angular frequency, $\omega_{\text{up}}$ and $\omega_{\text{low}}$, decreases with an increase in the fibrosis ratio.

To begin with, let us consider the reason $\omega_{\text{up}}$ decreases. It is obvious that input signal (4) induces the plane waves with the angular frequency $\omega$ as indicated in Figs. 2 and 3. Figure 5 shows the time-space pattern on the medium with fibrosis ratio 20% and $\varepsilon = 0.08$ for $\omega = 1.35$, which is under the upper bound $\omega_{\text{up}}$. Although the waves are disturbed due to the 20% fibrosis, they still keep their distances. It is clear that these distances become smaller with increasing $\omega$. For over the upper bound $\omega_{\text{up}}$, $\omega = 1.40$, the waves’ distances become smaller as shown in Fig. 6. Eventually, the disturbed plane waves touch together and collapse their formation. On the other hand, it was reported that the velocity of traveling waves decreases as the fibrosis ratio increases [15]. This report implies that, with increasing the fibrosis ratio, the plane waves’ distances become smaller. The disturbance effect and the smaller waves’ distances would explain the reason $\omega_{\text{up}}$ decreases with an increase in the fibrosis ratio.

It is known that if several waves with different frequencies exist in the medium, a wave with the highest frequency dominates the entire medium [18]. Hence, if the input signal frequency is higher than the spiral/chaotic waves frequencies, the plane wave induced by the input signal dominates the entire medium. This is the fundamental mechanism of suppression with the boundary periodic pacing. Since the frequencies of spiral/chaotic waves decrease with an increase in fibrosis ratio, even low-frequency input signal can suppress them with the high fibrosis ratio. This would be the reason $\omega_{\text{low}}$ decreases as the fibrosis ratio increases.

![Figure 4: Controllable ranges in $\varepsilon$-$\omega$ plane for fibrosis ratio 0, 10, 20%.](image)

![Figure 5: Time-space pattern on the medium with fibrosis ratio 20% and $\varepsilon = 0.08$ for $\omega = 1.35$.](image)

![Figure 6: Time-space pattern on the medium with fibrosis ratio 20% and $\varepsilon = 0.08$ for $\omega = 1.40$.](image)
5. Conclusion

The present paper dealt with the suppression of spiral waves and spatio-temporal chaos in excitable media with fibrosis. The influence of non-excitable cells on the suppression performance has been investigated by numerical simulations. According to these numerical investigations, we might suggest that the angular frequency of boundary periodic pacing has to be low for high fibrosis ratio (i.e., aged tissue).

References


Optimal Backstepping Control for Genesio–Tesi chaotic system
Using Genetic Algorithm

Mohammad Reza Modabbernia†, Ali Reza Sahab‡, Masoud Taleb Ziabari†
And Seyed Amin Sadjadi Alamdari

† Department of Electronics, Chamran Community Collage of Technology, Rasht, Iran.
‡ Faculty of Engineering, Electrical Group, Islamic Azad University, Lahijan Branch, Iran.
+ Faculty of Engineering, Mechatronic Group, Islamic Azad University, Qazvin Branch, Iran.

Email: m_modabbernia@afr.ac.ir, Ali.reza.sahab@gmail.com, m.t.ziabari@gmail.com, asadjadi@gmail.com

Abstract—This paper has presented chaos synchronization in the Genesio-Tesi using the backstepping approach. Backstepping approach consists of parameters which accept positive values. The parameters are usually chosen optionally. The system response is different for each value. It is necessary to select proper parameters to obtain a good response because the improper selection of the parameters leads to inappropriate responses or even to instability of system. Genetic algorithm can select appropriate and optimal values for the parameters. GA by minimizing the fitness function can find the optimal values for the parameters. This selected fitness function is for minimizing the least square error. Fitness function forces the system error to decline to zero rapidly that causes the system to have a short and optimal setting time. Fitness function also makes an optimal controller and causes overshoot to reach its minimum value. This hybrid makes an optimal backstepping controller.

1. Introduction

Chaos is a very interesting nonlinear phenomenon and has applications in many areas. One of the important problems in chaotic systems is synchronization. A robust adaptive PID controller for chaotic systems had been presented in [1]. Linear feedback for controlling chaos and Routh–Hurwitz criteria based on stability analysis has been done in [2]. Chaos suppression of Genesio system is achieved to use adaptive feedback linearization-based controller in [3]. Exponential Synchronization in the Genesio Tesi via a novel feedback control has been presented in [4]. Considerable effort has been also done to design control systems using feedback linearization and backstepping design technique for deterministic as well as uncertain chaotic systems [5-10]. Synchronization in the Genesio Tesi via Backstepping Approach has been presented in [11]. Backstepping design based on synchronization of two Genesio chaotic systems is proposed in [12]. Genetic algorithms(GAs) have been extensively applied to the off-line design of controllers [13].

Until now for controlling synchronization, Genesio-Tesi chaos has been used. In many of these methods, the designed controller has high overshoot and the system is reached stability after a long time. In some other controllers, the system is reached stability in appropriate time, but the system error is too much for a while.

In the backstepping controllers, the parameters of controller are chosen arbitrarily. In these controllers if we change some of these values, the system will be led to instability and it won’t respond well. In these controllers, the system’s behavior may not be good and they may have high overshoot. In some other controllers, the system has mush setting time and they may have oscillation behavior and show bad errors.

The paper is organized as follows: Section 2 describes Genesio-Tesi chaotic system. In section 3, a backstepping controller for synchronization is designed. Section 4 describes GA and algorithm used here. In section 5 a backstepping controller is designed for step tracking. Section 6 provides the conclusion.

2. Genesio-Tesi Chaotic System

Genesio–Tesi chaotic system can be represented by following set of nonlinear differential equations [14]:

\[
\begin{align*}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= x_3 \\
\dot{x}_3 &= -ax_3 - bx_2 - cx_1 + dx_1^2 + ku
\end{align*}
\]

where \(x_1, x_2\) and \(x_3\) are state variables, and \(a, b\) and \(c\) are positive real constants satisfying \(ab < c\). For instance, the system is chaotic for the parameters \(a = 1.2, b = 2.92\) and \(c = 6\). Here, \(a, b\) and \(c\) are linear parameters and \(d\) is nonlinear parameter which is taken as one without loss of generality. Constant scalar \(k\) is assumed to be known and \(u\) is the control input to the model. The initial condition for the states is taken as \(x(0) = [0.1;0.2;0.2]^T\).

Before controlling \((u=0)\) the nonlinear Genesio-Tesi given by equation (1) exhibits varieties of dynamical behavior including chaotic motion - displayed in Figure 1.
3. Design Controller with Backstepping

The backstepping is used to bring the states $x_1, x_2, x_3$ to the desired references via the torque $u$ calculated with four steps.

Step 1.
Consider the first subsystem of equation (1).
\[ \dot{x}_1 = x_2 \]
Construct the joint Lyapunov function.
\[ V_1(x_1) = \frac{1}{2}x_1^2 \]
Take $x_2$ as a virtual control input and choose.
\[ x_2 = \Phi_0(x_1) = -x_1 \]

Step 2.
Consider $(x_1, x_2)$ of equation (1):
\[ \dot{x}_1 = x_2 \]
\[ \dot{x}_2 = x_3 \]
Take $x_3$ as a virtual control input and choose:
\[ x_3 = \Phi_1(x_1, x_2) = -(x_1 + x_2)(1 + k_1) \]
And take the Lyapunov function as
\[ V_2(x_1, x_2) = V_0 + \frac{1}{2}(x_2 - \Phi_0)^2 \]

Step 3.
Consider all system
\[ \dot{x}_1 = x_2 \]
\[ \dot{x}_2 = x_3 \]
\[ \dot{x}_3 = -ax_1 - bx_2 - cx_3 + dx_1^2 + ku \]
Take $u$ as an actual control input and choose:
\[ u = \Phi_2(x_1, x_2, x_3) = \frac{1}{k}[\frac{\partial \Phi_1}{\partial x_1}x_2 + \frac{\partial \Phi_1}{\partial x_2}x_3 - (x_1 + x_2) - k_1(x_3 - \Phi_1) - (-c_1 - bx_2 - ax_3 + dx_1^2)] \]

4. Genetic Algorithm

In its most general usage, genetic algorithms refer to a family of computational models inspired by evolution. These algorithms start with many initial points in order to
cover all search intervals and encode a potential solution to a specific problem on a simple chromosome like data structure and apply recombination operators to these structures so as to preserve critical information. An implantation of genetic algorithms begins with a population of chromosomes randomly bred. We evaluate each chromosome by using the objective function called Fitness function. In order to apply the genetic reproductive operations called crossover and mutation, we select, randomly, two individuals called parents and we apply the crossover operation, if its probability reaches, between parents by exchanging some of their bits to produce two children. A mutation is the second operator applied on the single children by inverting its bit if the probability reaches. After this stage we obtain two population: a parent population and a children population, the individual who has a goodness solution is preserved [16].

The genetic algorithms are used to search the optimal parameters \( k_j \) in order to guarantee the stability of systems by ensuring negativity of the Lyapunov function and having a suitable time response. The fitness function used is

\[
f = \frac{1}{N} \sqrt{\sum_{i=1}^{N} (x_i - x_d)^2}
\]

(11)

\( x_i \) is system state and \( x_d \) is favorite mood for \( x_i \). Based the system purpose for placing the states at zero value; \( x_d = 0 \). By the training, can be obtained optimal parameters as \( k_1 = 0.708 \) and \( k_2 = 0.17 \).

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size population</td>
<td>100</td>
</tr>
<tr>
<td>Maximum of generation</td>
<td>300</td>
</tr>
<tr>
<td>Prob.crossover</td>
<td>75</td>
</tr>
<tr>
<td>Prob.mutation</td>
<td>0.001</td>
</tr>
<tr>
<td>( k_i ), search interval de</td>
<td>[0.1 10]</td>
</tr>
</tbody>
</table>

After using the Genetic Algorithm the result estimated are showed in figure (3). Figure 3a–c shows the state trajectory variation for Genesio system.

5. Step Response Tracking

Suppose, the \( x_1 \) state would be output of the system and it would track the input response. In this case by using the change of variable \( y = 1 - x_1 \), equation (1) would be converted to the equation (12)

\[
\begin{align*}
\dot{y} &= -x_2 \\
x_2 &= x_3 \\
\dot{x}_3 &= -ax_1 - bx_2 - c(1 - y) + d(1 - y)^2 + ku
\end{align*}
\]

(12)

Take \( u \) as an actual control input and choose:

\[
u = \Phi_2(x_1, x_2, x_3) = \frac{1}{k}[-\frac{\partial \Phi}{\partial y} x_2 + \frac{\partial \Phi}{\partial x_1} x_3 - (x_2 - y)] - k_2(x_3 - \Phi_1) 
\]

(13)

After using GA for the controller equation (13) obtain \( k_1 = 9.965, k_2 = 5.114 \). The setting time of optimal backstepping controller obtain 4.785s. The system response of step tracking is shown in figure 4.
5. Conclusion

This paper has presented a new hybrid backstepping approach with genetic algorithm demonstrated to have more optimal behavior when compared with previous methods. This approach is used for chaos synchronization in the Genesio-Tesi by using backstepping method to control the Genesio-Tesi chaos. The designed controller consists of parameters which accept positive values. The controlled system presents different behaviors for different values. Improper selection of the parameters causes an improper behavior which may cause serious problems such as instability of system.

Genetic algorithm optimizes the controller to gain optimal and proper values for the parameters. For this reason, GA minimizes the fitness function to find minimum current value for it. On the other hand, fitness function finds minimum value minimizing least square errors.

By this approach, the setting time and overshoot reach their minimum values demonstrated to have more optimal values when compared with previous methods. Also by selecting different fitness functions can have other appropriate results.

References

Recovering Piecewise Constant Signals from Noisy Time Series

Max A. Little† and Nick S. Jones†

†Oxford Complex Systems, Oxford Centre for Integrative Systems Biology, Dept. of Physics, University of Oxford, Oxford OX1 3PU, UK
Email: littlem@physics.ox.ac.uk, n.jones1@physics.ox.ac.uk

Abstract-- Piecewise constant (PWC) signals exhibit flat regions with abrupt jumps. These occur in many physical situations, including jump-diffusion in financial markets, atomic diffusion in crystal lattices, step-like motion of molecular machines, and single-molecule nanopore sequencers. Time series containing PWC signals typically observed in experimental setups are often contaminated by significant noise. The abrupt transitions make recovery of the PWC signal from the noisy time series a major challenge for classical linear time series analysis. In this talk, I will describe several new techniques that exploit concepts such as sparsity, shrinkage and fusion, to provide more accurate estimates of the underlying PWC signal. I will describe some applications in the analysis of the dynamics of molecular machines.
Multivariate synchronization analysis of EEG recordings from epilepsy patients
Ralph G. Andrzejak1 Georgia E. Polychronaki2, Andreas Schulze-Bonhage3 and Konstantina S. Nikita2

1 Department of Information and Communication Technologies, Universitat Pompeu Fabra, Barcelona, Spain
2 School of Electrical and Computer Engineering, National Technical University of Athens, Athens, Greece
3 Epilepsy Center, University Hospital, Freiburg, Germany
Email: ralph.andrzejak@upf.edu

1. Introduction

We compute multivariate synchronization measures from electroencephalographic (EEG) recordings of epilepsy patients. We extract alarms from the temporal profiles of these synchronization measures and study their potential for the prediction of epileptic seizures. The predictive performance of these alarms is assessed using analytical performance estimates and alarm time surrogates.

2. Methods

We applied a multivariate synchronization analysis to EEG recordings from epilepsy patients. For a single window of EEG signals recorded simultaneously with a number of k electrodes, at first two symmetric interdependence matrices, of dimension k times k, are extracted. One matrix contains all pairwise covariance values [1, 2] and the other all pairwise values of the mean phase coherence [3, 4]. Here the phases are extracted based on the Hilbert transform. Subsequently, the k eigenvalues of both matrices are determined, and an entropy value for each of the eigenvalue spectra is calculated [5]. We denote the entropy extracted from the covariance by SC and the one extracted from the mean phase coherence by SR. A moving window technique results in temporal profiles of SC and SR.

As EEG recordings we used a database that largely overlaps with one previously studied by Schad and colleagues [6]. This database consists of continuous long-term EEG recordings from six patients comprising a total duration of 460h and including 28 seizures of various types (simple partial, complex partial and generalized tonic-clonic seizures). All patients were undergoing presurgical epilepsy diagnostics at the Epilepsy Center, University Hospital of Freiburg. A distinctive feature of this database is that all recordings consist of simultaneous scalp and intracranial EEG. The scalp electrodes were placed according to the international 10-20 system, and the intracranial EEG was recorded from the surface of the cortex or from deeper structures of the brain. This recording setup allows us to probe different compositions of electrode groups as input to the multivariate synchronization analysis. In particular, we compare results derived from groups of scalp EEG recordings versus those derived from different groups of intracranial EEG recordings.

3. Results

Common features of the SC and SR profiles are reported, such as a strong temporal modulation reflecting the sleep-wake cycle of the patients. In addition, distinct differences such as a higher robustness of SR against artefacts in the recordings are shown. It is studied whether temporal variations of these profiles can serve to derive alarms to predict seizures. In particular, we use a simple threshold crossing to produce these alarms. We show that the sensitivity and specificity of the alarms for impending seizures largely depends on parameters such as the prediction horizon, smoothing of the profiles, or the value used for the threshold. Depending on these parameters, a very high sensitivity can be obtained for moderate false positive rates. We use analytical performance estimates [7] as well as seizure predictor surrogates [8] to compare this performance against the one expected under different well-defined null hypotheses. These tests provide strong evidence that rather than being indicative for a true predictive power of the extracted alarms this performance is consistent with the one expected by chance.

[6] Schad A, Schindler K, Schelter B, Maiwald T, Brandt A, Timmer J, Schulze-Bonhage A. Application of a multivariate seizure detection and prediction method to...
non-invasive and intracranial long-term EEG recordings. 
*Clin Neurophysiol* 2008; 119:197-211.


New nonlinear markers and insights into speech signal degradation for effective tracking of Parkinson’s disease symptom severity

Athanassios Tsanas\textsuperscript{1,2}, Max A. Little\textsuperscript{1,2,3}, Patrick E. McSharry\textsuperscript{1,2,4} and Lorraine O. Ramig\textsuperscript{5,6}

\textsuperscript{1} Systems Analysis, Modelling and Prediction (SAMP) group, Mathematical Institute and Department of Engineering Science, University of Oxford, Oxford, UK, \textsuperscript{2}Oxford Centre for Industrial and Applied Mathematics (OCIAM), Mathematical Institute, University of Oxford, Oxford, UK, \textsuperscript{3}Oxford Centre for Integrative Systems Biology, Department of Physics, University of Oxford, Oxford, UK, \textsuperscript{4}Smith School of Enterprise and the Environment, University of Oxford, UK, \textsuperscript{5}Speech, Language, and Hearing Science, University of Colorado, Boulder, Colorado, USA, \textsuperscript{6}National Center for Voice and Speech, Denver, Colorado, USA

Email: (A. Tsanas) tsanas@maths.ox.ac.uk, tsanasthanasis@gmail.com, (M. Little) littlem@maths.ox.ac.uk, (P. McSharry) patrick@mcsharry.net, (L. Ramig) Lorraine.Ramig@colorado.edu

Abstract— We have recently shown that speech signal degradation can be used to quantitatively predict average Parkinson’s disease (PD) symptom severity, which is typically evaluated on the Unified Parkinson’s Disease Rating Scale (UPDRS). In this study, we demonstrate the potential of wavelets to reveal changes in fundamental frequency variations with PD progression. We develop a set of new measures based on wavelets, energy, and entropy, which form robust indicators of the UPDRS. These results demonstrate that PD leads to dissimilar speech patterns in males and females, tentatively taken to indicate different patho-physiological mechanisms.

1. Introduction

Parkinson’s disease (PD) is a neurodegenerative disorder affecting approximately 100 people for every 100,000 in the population [1]. Since the probability for PD onset increases steeply over the age 50 [2] and given that the population worldwide is growing older, these estimates could increase further in the near future. Although PD is progressive and ultimately fatal, pharmacological and surgical treatments are available to alleviate some of the symptoms and slow down disease progression. Therefore, to optimize treatment, early diagnosis and frequent PD progression tracking are essential [3]. PD symptoms include tremor, rigidity and general deterioration of muscle control. The diagnosis of Parkinsonism is given when these symptoms can be attributed to neurotoxins or drugs; when the aetiology is unknown, the disease is termed idiopathic.

At present, PD symptoms are physically assessed by clinical experts using empirical tests and guidelines. The clinical rater’s assessment is typically expressed using the gold standard metric Unified Parkinson’s Disease Rating Scale (UPDRS) [4]. For untreated patients the UPDRS spans the numerical range 0-176, with 0 representing symptom-free (healthy person) and 176 total disability. The UPDRS consists of three components: (1) Mentation, Behavior and Mood; (2) Activities of daily living; and (3) Motor (muscular control). Collectively, the three components are known as total UPDRS. The third component is known as motor UPDRS, and ranges from 0-108, with 0 indicating no motor symptoms and 108 denoting total lack of motor control. The UPDRS assessment by clinical experts is costly to national health systems relying on human expertise, and often cumbersome to patients who have to physically visit the clinic over regular intervals. These factors impose the need for accurate, objective, remote tracking of average PD symptom severity.

Degraded speech performance has been qualitatively related with PD at least since the beginning of the 1960s [3], however, only recently strong evidence has emerged linking speech degradation with PD progression [5], [6]. Those studies prompted us to investigate the statistical mapping of a range of classical speech signal processing algorithms (known as dysphonia measures in the jargon of the speech literature) to UPDRS [7], [8]. In this study, we explore the effectiveness of wavelets to reveal changes in fundamental frequency variations with PD progression.

2. Methods

2.1. Data

We used data from the original study of Goetz et al., in which 52 subjects with idiopathic PD diagnosis within the past five years were recruited to a clinical trial [6]. Subjects were given a PD diagnosis if they had at least
two of the following symptoms: rest tremor, bradykinesia or rigidity, with no evidence of Parkinsonism. Their symptom severity was expressed using the UPDRS at three intervals: baseline, three-months and six-months into the trial. The recruited subjects were followed for six months during which they were asked to complete a series of tests weekly, using the Intel At Home Testing Device (AHTD). Among these tests, the subjects were required to sustain the vowel ‘ahh …’ for as long as possible and as steadily as possible. Four sustained vowel phonations were recorded at a level of loudness that was comfortable for the subject and two at twice the level of loudness that was comfortable on each day the recruits took the test. We used data from subjects that had completed at least 20 valid study sessions. Table 1 summarizes the details of the 42 PD subjects used in this study. After initial screening to remove flawed phonations (too short, patient coughing), we processed 5,875 signals using dysphonia measure signal processing algorithms implemented in the Matlab software package.

Table 1: Summary of the AHTD data

<table>
<thead>
<tr>
<th></th>
<th>MALES (28 subjects)</th>
<th>FEMALES (14 subjects)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age (years)</td>
<td>64.8 ± 8.1</td>
<td>63.6 ± 11.6</td>
</tr>
<tr>
<td>Weeks since PD diagnosis</td>
<td>63.0 ± 61.9</td>
<td>89.7 ± 81.2</td>
</tr>
<tr>
<td>Motor-UPDRS (baseline, 3-months, 6-months)</td>
<td>20.3 ± 8.5, 21.9 ± 8.7, 22.0 ± 9.2</td>
<td>17.6 ± 7.4, 21.2 ± 10.5, 20.1 ± 9.4</td>
</tr>
<tr>
<td>Total-UPDRS (baseline, 3-months, 6-months)</td>
<td>27.5 ± 11.6, 30.4 ± 11.8, 31.0 ± 12.4</td>
<td>24.2 ± 9.1, 27.4 ± 12.1, 26.8 ± 10.8</td>
</tr>
</tbody>
</table>

Figures are given in the form mean ± standard deviation.

2.2. Discrete wavelet transform and wavelet decomposition

Wavelets have the property of quantifying regularity effects (scale aspects) and transient processes (time aspects), qualities which make them well suited for detecting scale and time deviations from an expected pattern. The rationale is that a healthy person is expected to be able to sustain a vowel with minimal deviation from exact periodicity, whilst people with pathological voices cannot [9]. Moreover, wavelet decomposition is well adapted to the study of fractal properties and self-similarity of signals, properties of speech signals used previously in developing dysphonia measures [10]. The discrete wavelet transform (DWT) expresses the initial signal using approximation and detail coefficients. The wavelet decomposition then is successive expressing the approximation coefficients using subsequent layers to extract new approximation and detail signals. The layers of the wavelet decomposition are known as levels. Practically speaking, the resulting wavelet coefficients can be thought of as similarity (resemblance) indices between the selected wavelet and the signal in each level, where large coefficients denote large resemblance. For more details regarding wavelets we refer to [11].

2.3. Extracting features based on wavelets

As a first pre-processing step, we extracted the fundamental frequency $F_0$ from each of the 5,875 signals. Algorithms extracting $F_0$ focus on a time window of the original signal (the window can be either pre-specified, or it can be determined by the $F_0$ algorithm, e.g. using zero-crossing). Then, for each of those windows, the $F_0$ is estimated giving an $F_0$ series vector. There are many algorithms to compute $F_0$ and this is in itself a topic of intense research [12]; in this study, we used the robust RAPT algorithm [13]. Then, the input signal vector for wavelet processing is the $F_0$ series.

We applied wavelet decomposition of the $F_0$ series in 10 levels, and extracted the wavelet coefficients experimenting with three wavelet families (Daubechies, Symlets, Coiflets). Then, we computed the energy, entropy (using both Shannon’s and the log energy definitions), and the Teager-Kaiser Energy Operator (TKEO) for the wavelet coefficients at all levels. The TKEO can be thought of as a nonlinear measure of energy, taking into account both the amplitude and the frequency of the input signal (in this case the wavelet coefficients). It was first proposed in [14] and is defined as:

$$\Psi(x_n) = x_n^2 - x_{n+1} \cdot x_{n-1}$$  \hspace{1cm} (1)

where $n$ denotes the index of the input vector. Then, the TKEO vector gives rise to two features using its mean and its standard deviation for each level. Recently, Little et al. have shown that the transformation of the fundamental frequency into its logarithmic perceptual semitone can enhance robustness to confounding factors such as smooth vibrato prior to further processing [15]. Therefore, in addition to the features extracted using the raw $F_0$ series, we computed the log transform of the $F_0$ series and then followed the methodology already outlined to obtain additional features.

The results of all the algorithms using both the raw $F_0$ series and the log transformed $F_0$ series are appended in a feature vector, which is used to characterize each phonation. Essentially, the feature vector reduces the initial vector space with elements equal to the length of the $F_0$ series to a reduced space, where each element of the feature vector space can be thought of as a distinctive feature. This process was repeated for all the 5,875 phonations where each phonation was characterised by 180 features, resulting in a 5,875×180 design matrix.

2.4. Statistical mapping

The UPDRS values in the AHTD study were obtained at baseline, three-month and six-months, whilst the voice recordings were obtained at weekly intervals; therefore we
used a straightforward piecewise linear interpolation to obtain weekly motor-UPDRS and total-UPDRS scores. In doing so, we assumed that the UPDRS did not fluctuate wildly within the three-month intervals between the actual assessments by clinical experts. We have argued in [7], [8] that linear UPDRS progression is the physiologically most plausible disease course on average, an assertion supported by other recent studies on early PD [16], [17].

Using the interpolated UPDRS scores, we have the classical supervised learning problem where we want to develop a learner maximizing the accuracy of predicting the response variable \( y \) (UPDRS) given the features \( X \) (wavelet vector). We have used Breiman’s Random Forests (RF) [18], a powerful, nonlinear, non-parametric learner for the statistical mapping of features to UPDRS.

2.5. Feature selection

The use of a large number of features can potentially lead to poor population of the feature space occluding the detection of relevant patterns useful to predict the response variable. This well known problem is known as the curse of dimensionality, and is typically addressed by either transforming the initial feature space \( M \) into a new space \( m \) (where \( m < M \)), or by selecting \( K \) of the initial features in the \( M \)-dimensional feature space \( (m \) or \( K \) are determined by trial and error, for example using cross-validation). One of the advantages RF have over alternative learners, is that they rank features internally as an integral part of the statistical mapping process, effectively acting as feature selection wrappers. Therefore, we used the ranked sequence of features to decide on the most parsimonious model with performance within one standard error from the optimal (as defined in section 2.6).

2.6. Model validation (cross-validation)

The generalization performance of the proposed model was assessed using 10-fold cross validation with 100 runs. In each run, we randomly split the \( N \) phonations (4,010 for males and 1,865 for females). The training set comprises 0.9-\( N \) phonations and the testing set comprises 0.1-\( N \) phonations. We assess the performance of the learning scheme using the mean absolute error (MAE):

\[
MAE = \frac{1}{L} \sum_{i \in Q} |\hat{y}_i - y_i|
\]

where \( U_i \) is the true UPDRS value, \( \hat{y}_i \) is the predicted value, and \( L \) is the number of phonations in the dataset denoted by \( Q \), containing the indices of the particular set in each cross-validation run. The MAE over all cross-validation runs was averaged.

3. Results

Having extracted the \( F_0 \) series vector, we have experimented using different wavelet families in the DWT step. We found that the wavelets from the Daubechies, Coiflet and Symlet families had similar performance. The out of sample MAE results are summarized in Table 2, and were obtained using the dysphonia measure subsets of Table 3 and the Symlet 4 wavelet.

### Table 2: Out-of-sample mean absolute error (MAE) results

<table>
<thead>
<tr>
<th></th>
<th>Motor UPDRS</th>
<th>Total UPDRS</th>
<th>Motor UPDRS</th>
<th>Total UPDRS</th>
</tr>
</thead>
<tbody>
<tr>
<td>males</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>5.36 ± 0.34</td>
<td>6.93 ± 0.36</td>
<td>4.72 ± 0.38</td>
<td>5.47 ± 0.51</td>
</tr>
</tbody>
</table>

Figures are given in the form mean ± standard deviation.

### Table 3: Feature subsets selected using the Random Forests’ internal feature ranking property

<table>
<thead>
<tr>
<th>Males</th>
<th>Females</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log entropy 3(^{rd}) detail coef.</td>
<td>Log entropy 3(^{rd}) detail coef.</td>
</tr>
<tr>
<td>Log entropy 2(^{nd}) detail coef.</td>
<td>Log entropy 2(^{nd}) detail coef.</td>
</tr>
<tr>
<td>Log entropy 1(^{st}) detail coef.</td>
<td>Log entropy 1(^{st}) detail coef.</td>
</tr>
<tr>
<td>Log entropy 4(^{th}) detail coef.</td>
<td>Log entropy 4(^{th}) detail coef.</td>
</tr>
<tr>
<td>Shannon entropy 2(^{nd}) approximation coef. (with prior ( F_0 ) transform)</td>
<td>Shannon entropy 2(^{nd}) approximation coef. (with prior ( F_0 ) transform)</td>
</tr>
<tr>
<td>Shannon entropy 1(^{st}) approximation coef.</td>
<td>Shannon entropy 1(^{st}) approximation coef.</td>
</tr>
<tr>
<td>Log entropy 5(^{th}) detail coef.</td>
<td>Log entropy 5(^{th}) detail coef.</td>
</tr>
<tr>
<td>Shannon entropy 3(^{rd}) approximation coef.</td>
<td>Shannon entropy 3(^{rd}) approximation coef.</td>
</tr>
<tr>
<td>Shannon entropy 4(^{th}) approximation coef.</td>
<td>Shannon entropy 4(^{th}) approximation coef.</td>
</tr>
<tr>
<td>Log entropy 3(^{rd}) detail coef. (with prior ( F_0 ) transform)</td>
<td>Log entropy 3(^{rd}) detail coef. (with prior ( F_0 ) transform)</td>
</tr>
<tr>
<td>Shannon entropy 2(^{nd}) approximation coef.</td>
<td>Shannon entropy 2(^{nd}) approximation coef.</td>
</tr>
</tbody>
</table>

### 4. Discussion

We have introduced a number of new measures based on the use of wavelets to investigate how speech performance degradation can be mapped to UPDRS. We focused on the analysis of the fundamental frequency because this is the single most important characteristic of speech [9], and our previous exploration of the AHTD database [7], [8] has confirmed studies’ reporting that \( F_0 \) is adversely affected in PD. Our findings demonstrate that the new measures enable replicating the clinical raters’ assessments to within 7 UPDRS points for males and within 5.5 UPDRS points for females. This is notable improvement over previous results in [7] and [8] where we used broadly accepted speech signal processing algorithms.

The notable difference observed between the UPDRS estimation performance in males and females suggests that patterns associated with \( F_0 \) may be more indicative of PD in female subjects, an argument supported by other \( F_0 \) related measures in the AHTD database. Given that higher fundamental frequencies tend to have lower perturbations [19], and that women have on average higher \( F_0 \) [9], it is plausible that failure to sustain \( F_0 \) periodicity indicates
pronounced voice pathology in females whilst similar deviation from $F_0$ periodicity could be (at least partly) due to normal vibrato in males. Furthermore, studying Table 3 we can tentatively deduce interesting insights on the most useful PD patterns for both genders. One particularly interesting characteristic is that many of the features in the selected subset for females stem from prior log-transformation of the $F_0$.

The results of this study support the argument that features extracted based on wavelets are competitive alternatives to the classical dysphonia measures which are currently widely used to analyse dysphonic speech signals.

Acknowledgments

We are grateful to Ralph Gregory for medical insight and to Mike Deisher, Bill DeLeeuw and Sangita Sharma at Intel Corporation, and Paul Moore for fruitful discussions and comments on early drafts of the paper. We also want to thank James McNames, Lucia M. Blasucci, Eric Dishman, Rodger Elble, Christopher G. Goetz, Andy S. Grove, Mark Hallett, Peter H. Kraus, Ken Kubota, John Nutt, Terence Sanger, Kapil D. Sethi, Eejaz A. Shamim, Helen Bronte-Stewart, Jennifer Spielman, Barr C. Taylor, David Wolff, and Allan D. Wu, who were responsible for the design and construction of the AHTD device and organizing the trials in which the data used in this study was collected.

Declaration

A. Tsanas is funded in part by Intel Corporation and in part by the Engineering and Physical Sciences Research Council (EPSRC).

References

Detecting system state transitions in environmental time-series using non linear time series analysis

T.E. Karakasidis, A. Fragkou and A. Liakopoulos

Laboratory of Hydromechanics and Environmental Engineering, Department of Civil Engineering, University of Thessaly, Volos, Greece
Email: thkarak@uth.gr, fthanos@uth.gr, aliakop@uth.gr

Abstract– Environmental systems and time series emanating from such systems present a particular interest. System state transitions can occur in time and/or in space and the detection of such transitions can be particularly useful in the design related to such systems. Since the majority of physical systems present non linear behavior the use of appropriate tools is necessary. Recurrence Plots (RPs) and Recurrence Quantitative Analysis (RQA) along with Cross Recurrence Plots (CRPs) are some tools that permit to extract the underlying system dynamics [1-3].

In the present work we analyze time series from two different environmental systems. In the first case we study records of daily values of the Nestos river (Greece) water level at various measurement stations. Transitions from “periodicity” to “chaos” and “chaos” to “laminarity” are identified in time. In the second part we analyze temperature fluctuations in a horizontal round heated turbulent jet where instantaneous temperature time series were recorded at several points on the jet cross section. The temperature time series are analyzed in a first place using RQA. The variation of RQA measures is related with and interpreted via the transitions of the physical state of the fluid from the fully-turbulent flow near the jet centerline to the transitional flow near the boundary of the jet [9]. In a second phase the CRP analysis reveals correlations between the various parts of the jet.

1. Introduction

The analysis of time-series is of particular importance in understanding the dynamics of various environmental systems and in forecasting their behavior. The majority of methods for the analysis of environmental time series and forecast presuppose the linearity of the underlying dynamical system and the presence of stochastic noise. Classical linear methods (autocorrelation function, cross-correlation function, spectral analysis, etc.) have been proved exceptionally successful in the analysis of time series. However, they fail to distinguish chaotic behavior resulting from nonlinear deterministic systems. Moreover, they fail to detect correlations between two signals. For the detection of chaotic deterministic behavior and its discrimination from stochastic behavior, so as for correlations between two time series, nonlinear methods have been proposed which are based on the conceptual framework of time-lag embedding. Nonlinear methods include correlation-dimension-based characterization, average mutual information Recurrence Quantification Analysis and Cross Recurrence Quantification Analysis [1-3]. In this study, we focus on Recurrence Quantification Analysis and Cross Recurrence Quantification Analysis.

In a first place RP and RQA are applied on a time series of Nestos river water level daily measurements, recorded over a period of 16 years and 4 months. From the global inspection of the RP as well as from the use of epoch RQA analysis different areas of transition were identified among “laminarity”, “chaos” and “periodicity” states.

In the second part we consider a horizontal round heated turbulent jet. We analyze the macroscopic local temperature fluctuation measurements along a horizontal line, located in a cross section of the jet, so that some of our time series correspond to conditions of fully-developed turbulence (these are the time series obtained close to the centerline of the jet) while other time series have both laminar and turbulent flow characteristics (intermittency, etc.) i.e., fluctuations measured close to the boundary between the heated jet and ambient water. Considering the interrelations between the time series by means of CRP analysis we observe strong indication of non linear relationship between the data near the center line of the jet in contrast with the boundaries of the jet.

2. Theoretical Background

2.1. Recurrence plots

Recurrence Plot is a graphical tool introduced by Eckmann et al. [1] in order to extract qualitative characteristics of time series. It is a powerful tool with the great advantage it can be applied on non-stationary data. It is based on the reconstruction of phase space and on the recurrence of states (eqn. 1).

\[ R_{ij}^{m,\varepsilon} = \Theta(\varepsilon - \| \hat{x}_i - \hat{x}_j \|) \quad i, j = 1, \ldots, N \]  

(1)

\( R_{ij} \) is the recurrence matrix where the corresponding RP is based on, \( m \) the embedding dimension, \( \varepsilon \) the cutoff distance for the points considered to be recurrent and \( \Theta \) is the Heaviside function. If the points are located at smaller distances than the considered distance \( \varepsilon \) are recurrent and \( \Theta=1 \), else \( \Theta=0 \). The RP is obtained by plotting the recurrence matrix using different colours for its binary
entries, e.g. plotting a black dot at coordinates \((i,j)\) if the corresponding element of the recurrence matrix is \(R_{ij} = 1\) and a white dot for \(R_{ij} = 0\). By definition \(R_{ij} = 1\) for every \(i\) thus the RP has a black main diagonal line called line of identity. Moreover RPs are symmetric by definition with respect to the diagonal \(R_{ji} = R_{ij}\). When computing an RP a norm must be chosen. The most widely employed norms are the \(L_1\) norm, the \(L_2\) norm (Euclidean norm) and the \(L_\infty\) norm (maximum or supremum norm) [3]. In the present study the Euclidean norm was used since it results in an intermediate number of neighbours compared to \(L_1\) and \(L_2\) norms [3].

### 2.2. Recurrence Quantification Analysis

Webber & Zbilut [2, 4] and Marwan et al. [5] extended the idea of Recurrence Plots and defined a number of measurable quantities that can be extracted from the RPs giving rise to Recurrence Quantification Analysis (RQA). We present briefly some of the RQA indices that have been proposed [2, 5] and we have employed in the present study:

#### 2.2.1. %Recurrence

It gives the ratio of the number of recurrence points (pixels) to the total number of points (pixels) of the plot.

\[
\%REC = \frac{\sum_{i=1}^{N} R_{ij}}{N^2} \tag{2}
\]

#### 2.2.2 %Determinism

It represents the ratio of the number of recurrence points forming upward diagonal lines to the total number of recurrence points

\[
\%DET = \frac{\sum_{j=1}^{N} P(l)}{\sum_{j=1}^{N} P(l)} \tag{3}
\]

#### 2.2.3 MaxLine

Maxline is the length of the longest diagonal line segment in the plot, excluding the main diagonal line of identity. This is a very important recurrence variable because it is related to the Lyapunov exponents [1, 6]

\[
L_{max} = \max(\|\mathbf{y}_i - \mathbf{y}_j\|) \tag{4}
\]

#### 2.2.4. Trapping Time

It shows the average length of the vertical lines.

\[
TT = \frac{\sum_{t=1}^{N} tP(t)}{\sum_{t=1}^{N} P(t)} \tag{5}
\]

Trapping Time represents the average time that the system has been trapped in the same state.

### 2.3. Cross Recurrence Plots

In order to analyze the dependencies between two different systems by comparing their states the idea of Cross Recurrence Plots was proposed [8, 10]. This concept, based on Recurrence Plots considers a phase space with two trajectories \(x_i\) and \(y_j\) of length \(N_x\) and \(N_y\) respectively.

\[
CR_{x_\varepsilon}^{y_\varepsilon} = \Theta(\varepsilon - \|\mathbf{x}_i - \mathbf{y}_j\|) \quad \mathbf{x}_i, \mathbf{y}_j \in \mathbb{R}^m \tag{6}
\]

Analogous to the RPs the philosophy of the CRPs is that, if the points of the two trajectories are close enough, a black point will be marked onto the CR matrix at \((i,j)\) location. In CRP instead of the Line of Identity, the Line of Synchronization makes its appearance.

### 2.4. Cross Recurrence Quantification Analysis

Zbilut et. al. [8] gave an extension in quantification of CRPs, by introducing the Cross Recurrence Quantification Analysis. This analysis based on the fact that long diagonal lines in the CRP reveal similar time evolution of the trajectories of both processes. This similarity is quantified by introducing some quantitative measures which consider the frequency distributions of the diagonal line lengths \(P(l) = \{i; j = 1\ldots N\}\) for each diagonal parallel to the main diagonal \(CR_{x_\varepsilon}^{y_\varepsilon} (i = j)\). Considering that for \(k = 0\) the line is the LOI, \(k > 0\) the diagonals above and \(k < 0\) the diagonals below the LOI, the quantitative measures are the following.

#### 2.4.1. Recurrence Rate (RR\(_k\))

\[
RR_k = \frac{1}{N - k} \sum_{i=1}^{N-k} P(l) \tag{7}
\]

It represents the probability of occurrence of similar states in both systems with a certain delay \(t = k\Delta t\).

#### 2.4.2 Determinism (DET\(_k\))

The proportion of recurrence points forming long diagonal lines to all recurrence points. However, it is constrained to the considered diagonal.
2.4.3. Averaged Line Length ($L_k$)

It quantifies the duration of having two deterministic processes similar time evolution in the phase space.

$$
DET_k = \frac{\sum_{i=1}^{N-k} I_{P}^i (l)}{\sum_{i=1}^{N-k} I_{P}^i (l)}
$$

(8)

The corresponding time series of water level appears in Fig.1A and the corresponding RP in Fig. 1B. From the global inspection of the RP of Nestos river, 16 lines parallel to the main diagonal are observed (periodicity) so as the Airplane Structure of the RP (sign of trend in our time series). Moreover from a closely inspection of the RP and the RQA (Fig.1C) we can observe transitions from Periodicity to Chaos, Chaos to Laminarity, Laminarity to Chaos, Chaos to periodicity and Periodicity to Chaos. We note that these terms do not have the same meaning as in fluid dynamics. In “Chaotic” regions (650-1300), (2750-3900), (5450-6000) deterministic lines are very small thus we observe that Maxline drops down to 25 and sometimes even to zero. In Periodic areas (0-650), (3900-5450) values of %determinism and maxline are very high, 92.8% and 62 respectively. In Laminar region (1300-2750) we observe high values of %determinism, almost 94%, but no large deterministic lines (parallel to the main diagonal) appear, so the maximum line is low (25 to 30). State remains “trapped” in time, Trapping time has high values (6.6 and 6.7), while in the other regions the values are smaller.

3. Results and Discussion

Time series analyses were performed using CRP toolbox [11], Visual Recurrence Analysis (VRA) [12], and TISEAN [13] packages.

3.1. RP and RQA in Nestos environmental system time series

The corresponding time series of water level appears in Fig.1A and the corresponding RP in Fig. 1B. From the global inspection of the RP of Nestos river, 16 lines parallel to the main diagonal are observed (periodicity) so as the Airplane Structure of the RP (sign of trend in our time series). Moreover from a closely inspection of the RP and the RQA (Fig.1C) we can observe transitions from Periodicity to Chaos, Chaos to Laminarity, Laminarity to Chaos, Chaos to periodicity and Periodicity to Chaos. We note that these terms do not have the same meaning as in fluid dynamics. In “Chaotic” regions (650-1300), (2750-3900), (5450-6000) deterministic lines are very small thus we observe that Maxline drops down to 25 and sometimes even to zero. In Periodic areas (0-650), (3900-5450) values of %determinism and maxline are very high, 92.8% and 62 respectively. In Laminar region (1300-2750) we observe high values of %determinism, almost 94%, but no large deterministic lines (parallel to the main diagonal) appear, so the maximum line is low (25 to 30). State remains “trapped” in time, Trapping time has high values (6.6 and 6.7), while in the other regions the values are smaller.

3.2. CRP and CRQA in a round heated jet

In order to find the center line of the jet in the second environmental system, instead of recurrence Plots and Recurrence Quantification Analysis methods of spectrum analysis and mutual information were applied [9]. According to all above methods, we found that the center of the jet is located at $x=17.5$ cm. So the CRP analysis gives the correlations between $x=17.5$cm and the other parts of the jet. Figure 3 shows CRPs of $x=17.5$ and $x=16.5$cm (near the center line of the jet) and CRPs of $x=17.5$cm and $x=9.5$cm (left boundary of the jet). It can be clearly seen that as we approach the boundary of the jet ($x=9.5$cm) the lines parallel to the main diagonal tend to became very small. On the other hand CRP which is close to the center line seems to have lines parallel to the main diagonal big enough to reveal the correlation between the states.
around 16 for RR/L+. Moreover, the same states give maxima around 12 (x=16.5cm) and 18 (x=18.5cm) for RR/L-. Those maxima for + and – measures are a strong indication of a non-linear relationship between the data [3].

The solid lines between the maximum lags of RR and L measures show the differences between those lags. So as we approaching the boundaries (x=9.5cm and x=23.5cm) those differences seem to become larger. The quantification analysis of CRPs was able to detect nonlinear relations between time series.

![](image1)

**Fig.3** Cross Recurrence Plots of x=17.5cm with (A) x=16.5cm and (B) x=9.5cm.

![](image2)

**Fig.4** Cross Recurrence Quantification Analysis of x=17.5cm with all the states across the jet from x=9.5cm to x=23.5cm (left and right boundaries of the jet)

### 4. Conclusions

From the analysis of the first environmental system we can conclude that characteristic times of the dynamics of the system were revealed both from the visual inspection, as well as from the quantitative analysis of RQA. The system presents transition among “periodicity”, “chaotic” and “laminar” states. We believe that these transitions are related to the variation of other environmental time series, mainly total precipitation. However, river water levels and stream flow are naturally characterized by smoother variations in time compared to the driving process of precipitation. RQA-Epoch analysis identifies multiannual periods where specific characteristics appear persistently.

The Cross Recurrence Quantification Analysis of the second environmental system shows interdependencies between the states of the system which gives as information about the turbulence effect on the flow state and is suitable in order to find the nonlinear relation between the considered data series. Further research is under progress in order to extract characteristic features of the system.

### Acknowledgments

Part of this research has been funded by the program «PYTHAGORAS-IV», grant uth-51908.04, jointly funded by the European Community and the Greek Ministry of Education.

### References

Associative Dynamics of Color Images in a Chaotic Neural Network

Makito Oku† and Kazuyuki Aihara‡,

†Department of Mathematical Informatics, Graduate School of Information Science and Technology, The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
‡Institute of Industrial Science, The University of Tokyo
4-6-1 Komaba, Meguro-ku, Tokyo 153-8505, Japan
Email: oku@sat.t.u-tokyo.ac.jp

Abstract—
Recently, we have succeeded in numerically simulating a large-scale chaotic neural network with 1 million units. This success will open new possibilities for applications of chaotic neural networks from the viewpoints of academic interest as well as new media art. In this paper, we report a way to deal with color images by using a large-scale chaotic neural network. In the proposed method, color images are converted to binary sequences, modified slightly by inverting some bits, and stored in the network. The results of numerical simulations show that chaotic alternations among stored patterns as well as their reverse patterns can be observed within a certain range of parameters. We also compare four different coding schemes of color information, which changes the appearance of chaotic dynamics.

1. Introduction

The recent development of computing technologies allows us to handle much larger neural networks than before. One of the leading studies in this direction is the “Blue Brain Project” [1]. The first subgoal of this project, that is, the simulation of a single cortical column, has been achieved; in this simulation, 10,000 biologically detailed neuron models are used. On the other hand, Izhikevich has performed a simulation of the entire neocortex and thalamus by using 1 million neuron models that require few computational resources but retain the realistic physiological properties of neurons [2].

Last year, our group succeeded in simulating a large-scale chaotic neural network with 1 million units [3]. A characteristic feature of the chaotic neural network model [4, 5, 6] is that the network’s state chaotically wanders among multiple attractor states. The model is used for the phenomenological modeling of the dynamical associative memory or the endogenous perceptual alternation as well as for efficient optimization methods [6, 7, 8, 9]. Although large-scale chaotic neural networks have been investigated in the context of optimization problem solving [10] or electrical circuit implementation [11], no one has investigated a large-scale dynamical associative memory model.

We found in our previous study [3] that the network exhibits chaotic itinerancy among the stored patterns if we preprocess the patterns to modify their statistics. We also found that even an incompletely retrieved image—the network output differs from the correct image at many pixels—can evoke a clear perception of the original image. Because of these features, a large-scale dynamical associative memory model will have new possibilities for applications from the viewpoints of academic interest as well as new media art.

In this study, we store color images in the network and investigate whether chaotic itinerancy occurs. In addition, we are also interested in the appearance of the output of the network. Since an associative memory model can store only binary patterns, the conversion of color images to binary sequences is required. We first attempt the most simple and direct way of conversion and then carry out other three conversion methods.

2. Chaotic Neural Network Model

First, we will explain the chaotic neural network model [4, 5, 6]. In this study, we consider a recurrent neural network with \(N\) units. The external input is constant in both the time and the spatial domains. Each unit has two internal state variables \(\eta\) and \(\zeta\) and one output variable \(x\). If we adopt the vector representation, \(\eta = [\eta_1, \ldots, \eta_N]^T\), \(\zeta = [\zeta_1, \ldots, \zeta_N]^T\), and \(x = [x_1, \ldots, x_N]^T\), the system’s dynamics can be described as follows:

\[
\eta(t+1) = k_f \eta(t) + W x(t),
\]

\[
\zeta(t+1) = k_r \zeta(t) - ax(t) + a,
\]

\[
x(t+1) = f(\eta(t+1) + \zeta(t+1)).
\]

Here, \(W\) denotes the \(N \times N\) weight matrix; \(k_f, k_r \in [0, 1]\), the time constants; \(a \geq 0\), the strength of the refractoriness; and \(a\), which is a vector of constant values \(a \geq 0\), the external input. The first internal state variable \(\eta\) changes in response to the input from other units. The other internal state variable \(\zeta\) depends on the output of each unit. The output \(x\) is defined by a nonlinear function of the summation of the internal state variables. Here, \(f\) is an operation that applies the sigmoid function below to each element of the argument vector:

\[
f(y) = \frac{1}{1 + \exp(-y/\epsilon)}.
\]
Next, we explain the associative memory model. In general, the associative memory model has two phases: the encoding phase and the retrieval phase. In the encoding phase, \( K \) binary patterns \( s^1, \ldots, s^K \in \{-1, 1\}^B \) are given. For simplicity, we assume that each pattern contains equal numbers of 1 and -1. Then, the weight matrix is determined by the autocorrelation matrix of the patterns:

\[
W = \frac{1}{K} \sum_{k=1}^{K} s^k(s^k)^T.
\]

In the retrieval phase, one of the stored patterns with a slight perturbation is given as an initial state. Then, the corresponding stored pattern is recovered in finite steps. This phenomenon is called pattern completion. Each stored pattern corresponds to an equilibrium point in the phase space.

The dynamical associative memory model differs from the associative memory model in that the system’s state is attracted to a stored pattern for a short period, but leaves the pattern after a while, and then visits other patterns. The chaotic neural network model exhibits such behavior in some parameter regions. Neither the order of visiting nor the duration of each stay is predictable although the system’s dynamics is deterministic. This phenomenon is called chaotic itinerancy.

Many previous works on the dynamical associative memory model use approximately 100 units. In order to perform large-scale simulations, one of the major problems is the requirement of a considerably large amount of memory capacity for representing the weight matrix, which increases in \( O(N^2) \). However, the all-to-all connection regime used in the associative memory model has high redundancy; even if we remove some of the network connections, the qualitative property of the dynamics can be retained. Therefore, we use a partially connected network in our simulations. The weight matrix \( W = \{w_{ij}\} \) is changed as follows:

\[
w_{ij} = \left\{ \begin{array}{ll} \frac{1}{K} \sum_{k=1}^{K} s^k_j s^k_i & e_{ij} \in E \\ 0 & \text{otherwise} \end{array} \right.,
\]

where \( e_{ij} \) denotes a connection from unit \( i \) to unit \( j \), and \( E \) denotes the set of all the connections. Notice that if \( K \) is even, we can get rid of the connections whose values are 0.

### 3. Conversion of Color Images to Binary Sequences

In our previous study [3], binary images of 1,000 \( \times \) 1,000 pixels were transformed to \( 10^6 \)-dimensional vectors by concatenating all rows and stored in the network. Although binary images are useful for investigating the dynamical associative memory model, applications of the model to multivariate images such as gray-scale images or color images should be appreciated from a practical point of view. Henceforth, we only consider color image processing since the same techniques as those proposed here can be applied to gray-scale images in a straightforward manner.

The basic color image representation is \( RGB \) (red, green, and blue). \( RGB \) values are normally represented by integer values from 0 to 255. Each value is encoded in 8 bits; 24 bits represent the complete information of a pixel. By concatenating these 24-bit sequences, we can directly convert the 24-bit \( RGB \) color images to binary sequences of length \( 24 \times (\text{the number of pixels}) \).

The obtained binary sequences of 1 and 0 are transformed to those of 1 and -1, and then their statistics are adjusted by inverting the minimum number of bits (see Table 1). This preprocessing balances the numbers of 1 and -1 in each pattern as well as equalizes the amount of overlaps among patterns.

In addition to the abovementioned conversion method, we investigated two other color spaces. First, we investigated \( YIQ \), which is used by the TV system in the U.S. and Japan. The \( YIQ \) values of an image are obtained from the \( RGB \) values of the image by using the linear transformation below [12]:

\[
\begin{pmatrix}
Y \\
I \\
Q
\end{pmatrix} =
\begin{pmatrix}
0.2990 & 0.5870 & 0.1140 \\
0.5957 & -0.2745 & -0.3213 \\
0.2115 & -0.5226 & -0.3111
\end{pmatrix}
\begin{pmatrix}
R \\
G \\
B
\end{pmatrix},
\]

where \( RGB \) values are rescaled from 0 to 1. \( Y \in [0, 1], I \in [-0.5958, 0.5957], \) and \( Q \in [-0.5226, 0.5226] \). The first component \( Y \) provides the luminance value, and the other two components describe the chrominance information.

In our simulations, the \( YIQ \) values that are converted from the original \( RGB \) values are rescaled from 0 to 255 and digitized. Then, we can obtain a 24-bit representation per pixel as in the case of \( RGB \) images.

Next, we investigated \( HSB \) (hue, saturation, and value), also known as \( HSB \) (hue, saturation, and brightness), which is commonly used in image processing because it is more natural to human vision. In the determination of \( HSB \), we consider a cylindrical coordinate because hue is a circular quantity. Let \( M = \max(R, G, B), m = \min(R, G, B) \), and \( C = M - m \). If \( R \neq G \neq B \) holds (thus \( M, C \neq 0 \)), the \( HSB \) color space is defined as follows [13]:

\[
H = \begin{cases} 
60^\circ (G \mod B)/C & (M = R) \\
60^\circ (B \mod G)/C + 120^\circ & (M = G) \\
60^\circ (R \mod G)/C + 240^\circ & (M = B)
\end{cases},
\]

\[
S = C/V,
\]

\[
V = M.
\]

In addition, \( S = 0 \) if \( C = 0 \), \( V \neq 0 \), and \( V = 0 \) if \( M = 0 \), \( H \in [0^\circ, 360^\circ) \), \( S \in [0, 1] \), and \( V \in [0, 1] \). Usually a certain value is assigned if a value is undefined (\( H = 0 \) if \( C = 0 \), for example). As in the case of the \( YIQ \) values, \( HSB \) values are rescaled from 0 to 255 and digitized in our simulations in order to obtain a binary representation.

We also investigated another encoding method. The color space is \( RGB \), but gray code is used instead of binary
code for transforming the 0–255 integer values into 8-bit binary sequences. In gray code, every pair of successive values differs in only one bit. A binary code representation of an integer \( n \) can be converted to a gray code representation by \( n \oplus \lfloor n/2 \rfloor \), where \( \oplus \) denotes an XOR operation.

Network output can be decoded in the opposite manner. If the recovered \( RGB \) values are outside the specified range, we reset the values larger than 255 to 255 and the negative values to 0.

### 4. Simulations and Results

In the following simulations, we set \( k_f = 0.8 \), \( k_r = 0.9 \), \( a = 6.4 \), \( \alpha = 12 \), and \( \epsilon = 0.015 \). Each unit receives input from 100 units selected at random. These connections include those of value 0, which are removed before carrying out the simulations. As an initial condition, \( \eta_i \) takes a random value that is uniformly distributed in \([0, 1]\), and \( z_i \) is set to 0.

The source code of the program is written in the C programming language with Message Passing Interface (MPI). The program is run on a cluster of eight Linux server machines that have a single 3.2–3.6 GHz processor and 2.0 GB RAM each. It takes approximately 1 s to compute one step of simulation.

First, we use four color images represented in \( RGB \) space, as shown in Fig. 1, and convert them to binary sequences by using binary code. All images have 256 × 256 pixels. The \( RGB \) values range from 0 to 255. The number of units, or the length of the binary sequences, is 1,572,864.

After the preprocessing, 1.1–3.9\% (Ave. 2.6\%) of the bits in each binary sequence are inverted. The root-mean-square (RMS) error per color component of a pixel is 0.67.

Figure 2 shows a typical time series of the decoded network output. The four stored patterns as well as their reverse patterns are retrieved alternately. A few cycles of oscillation between a stored pattern and its reverse one is also observed.

To analyze the time series of the network output \( x(t) \), we calculate the Hamming distances between the digitized network output and the four binary sequences (see Fig. 3). The downward and upward peaks correspond to the retrievals of the stored patterns and the reverse ones, respectively. A similar chaotic behavior is observed even in the run of \( 10^5 \) steps. The maximum Lyapunov exponent is estimated to be 0.67.

Next, we apply the other three coding schemes: \( YIQ \) space with binary code, \( HSV \) space with binary code, and \( RGB \) space with gray code. After preprocessing, the RMS error is 1.58, 1.23, and 0.70 for the \( YIQ \), \( HSV \), and gray code cases, respectively. A chaotic behavior similar to the previous case is observed in all the three cases. The appearance of the retrieved images decoded from the network output is also similar to that in the previous case for the stored patterns, but qualitatively different for the reverse patterns except in the \( YIQ \) case (see Fig. 4).

### 5. Discussions

The results of numerical simulations, as shown in Figs. 2 and 3, confirmed that chaotic alternations among stored patterns as well as their reverse patterns can be observed if we store color images in the network. The positive maximum Lyapunov exponent is another evidence for the existence of chaos, and the lifetime of chaotic behavior in the considered network appeared to be significantly longer than that of a smaller network [6].

![Figure 1: Original color images for stored patterns. From left to right, “Mandrill”, “Lena”, “Peppers”, and “Tree”.](image1)

![Figure 2: An example of time series of the decoded network output displayed with 10 step intervals.](image2)
In addition, it appeared that the color images decoded from the network output, as shown in Fig. 2, can evoke a clear perception of the original images. These observations are also consistent with the results of our previous work in which binary images were used as stored patterns [3].

We found that the use of different color coding schemes can change the appearance of chaotic dynamics, as shown in Fig. 4. This would be because the bit-wise reverse operation may have a qualitatively different effect under different coding schemes in general case.

The reason why \( YIQ \) does not change the appearance can be explained as follows. To rescale the \( YIQ \) values obtained by using Eq. 7, we multiply these values with a scaling matrix and add a translation vector. The reverse operation applied to the rescaled values has the same effect as in \( RGB \) space. The same is true for few other linear transformations of \( RGB \) space such as \( YUV \) and \( I_1I_2I_3 \). However, a linear transformation in general does not have such a property.

6. Conclusion

In conclusion, we stored 24-bit RGB color images of 256x256 pixels in a large-scale chaotic neural network with approximately 1.6 million units. To convert the color images represented by integer values from 0 to 255, we used an 8-bit binary representation. We observed chaotic alternations among the stored images as well as their reverse images. We also found that the use of different color coding schemes could change the appearance of chaotic dynamics. Finally, as the future work, we plan to store as many images as possible in the network.

Acknowledgments

This research is partially supported by the Japan Society for the Promotion of Science, a Grant-in-Aid for JSPS Fellows (21-937).

References

Nonlinear Image Processing for Multiple Object Tracking on Cellular Hardware Platform

Takao Matsui†, Tomohiro Fujita†, Mamoru Nakanishi‡, Takeshi Ogura†

†Ritsumeikan University, 1–1–1 Noji-Higashi, Kusatsu, 525–8577
†E-mail: ri008067@ed.ritsumei.ac.jp, {tfujita, togura}@se.ritsumei.ac.jp
‡NTT Microsystem Integration Laboratories, 3–1, Morinosato Wakamiya, Atsugi, 243–0198
‡E-mail: mamoru@aecl.ntt.co.jp

Abstract: We developed a nonlinear image processing method for multiple object tracking on cellular automata on content addressable memory (CAM²), which is a cellular hardware platform that can attain pixel-order parallelism on a single PC board. We successfully implemented edge detection, hole filling, and center point detection, all of which are based on Cellular Automata (CA) processing, on CAM². Evaluation results show that real-time multiple object tracking can be performed on CAM² and that its performance is superior to that of conventional methods.

1. Introduction

Multiple object tracking is one of the most useful image processing applications in the security field. For multiple object tracking, various image processes—such as noise reduction, edge detection, hole filling, and center point detection—are necessary. Nonlinear image processing techniques, based on morphology and discrete-time cellular neural networks (DTCNN) processing have already been proposed [1] on a cellular hardware platform "called cellular automata on content addressable memory" (CAM²).

On CAM², each cell is connected to the nearest neighbor cells and all cell values can simultaneously change by referring to the values of these neighbors. This makes CAM² a good fit for Cellular Automata (CA)-based techniques, such as morphology and DTCNN [2][3][4], but although DTCNN has already been successfully implemented on CAM², its performance is inadequate in terms of processing time and program code size.

We propose CA-based image processing for multiple object tracking and implementation on CAM² in place of DTCNN. Our method performs edge detection, hole filling, and center point detection at a lower processing time than DTCNN processing, demonstrating its potential for multiple object tracking at high speeds.

2. Cellular Hardware Platform: CAM²

2.1. CAM² Features

CAM² was established on Content Addressable Memory (CAM) and CAM-based system technologies five years ago. At that time a highly parallel integrated circuits and system (HiPIC) [5] was proposed as a CAM-based system model for real-time image processing. The HiPIC is an application-specific system that achieves high performance and flexibility, and we have developed various practical real-time image processing and cellular type processing systems [6] [7] based on it. Figure 1 shows a block diagram of CAM².

Fig. 1. Block diagram of CAM².

According to the HiPIC concept, CAM² consists of a highly parallel PE array, an FPGA that controls the array, a RISC processor or DSP that executes serial data processing, and a specified amount of memory. Each is implemented using only digital LSI technology. The most prominent features of the configuration are dedicated CAMs for the highly parallel PE array. Each CAM performs various types of parallel data processing for CA in each word. Since the memory-based structure of the CAM is the most suitable for
LSI technology, several hundred thousand CA cells (or “processing elements”) can be created on a single PC board using deep sub-micron CMOS technology. Moreover, CAM\(^2\) can be easily controlled by command sequences generated from the FPGA, which is a reconfigurable logic element. In short, CAM\(^2\) can effectively perform CA-based image processing.

2.2. Cellular Automaton Processing on CAM\(^2\)

CA processing using CAM\(^2\) is carried out by iterative operations of CA-value transfer and update. In the CA-value transfer process, the values of an original cell are transferred to the nearest neighbor’s cell. In the CA-value update process, the next value of the original cell is calculated by a transition rule using the original and nearest neighbors’ cell values.

To perform this processing, the following functions are absolutely essential:

- Maskable OR search
- Partial & parallel write
- Hit-flag shift up & down

For the search, the hit results are accumulated into hit-flag registers by means of OR logic. For the write, the data are written into specific bit positions of multiple word locations. For the shift, the hit-flag registers are shifted to up or down words. Through the iteration of these functions, CA-value transfers and updates can be performed in a bit-serial, word-parallel manner. The drawback is that the processing takes too long for complex operations such as the multiplication of longer bits. Moreover, the processing time for CA-value transfer is in proportion to the transfer bit length. Thus, low-bit CA-value updates are required to shorten the processing time.

3. Morphology and DTCNN Processing for Multiple Object Tracking

Morphology and DTCNN processing have already been proposed for multiple object tracking [1]. An example of image processing for multiple object tracking using CAM\(^2\) is shown in Fig 2. In the first step of this process, object outlines are extracted by edge detection performed by DTCNN processing. Next, hole filling is performed by DTCNN processing followed by noise reduction performed by dilation of morphology. Finally, center points of the objects are extracted by DTCNN processing.

![Fig. 2. Example of image processing.](image)

4. CA-based Processing on CAM\(^2\) for Multiple Object Tracking

4.1. CA-based Edge Detection

In CA-based edge detection, the transition rule for the CA-value update is quite simple, only when all the nearest neighbor cells are black should the next CA-value be changed (form black to white), as shown in Fig. 3. The results of CA-based and DTCNN-based edge detection are the same.

![Fig. 3. Example of CA-based edge detection.](image)

4.2. CA-based Hole Filling

In CA-based hole filling, the output image is generated using the original image and a transition rule. The original image is preserved through the processing, and the initial output image is all black, as shown in Fig. 4(a). The output image is then moderated in a wave propagation manner by the CA-value update, as shown in Fig. 4(b) to (f). This update is repeated iteratively based on the pixel size. The transition rule is quite complicated, and the process converges after the pixel size steps.
4.3. CA-based Center Point Detection

CA-based center point detection is composed of four processing elements: upper-side, lower-side, left-side, and right-side. The transition rule of the upper-side processing is shown in Fig. 5. When the upper cells are all white, and one or more lower cells are black, the center cell changes from black to white.

![Upper-side processing](image)

The transition rule of under-side, right-side, and left-side processing is similar to that of upper-side processing but not quite the same. The four processes are repeated iteratively until no processed cells remain. CA-based center point detection is shown in Fig. 6.

![Center point detection](image)

5. Experiments and Evaluation

5.1. Processing Performance

Fig. 7 shows a photograph of CAM². A highly parallel PE array (a CAM LSI) is mounted on a daughterboard, while a motherboard contains an FPGA and some memory. CAM² can handle a 128 × 128 image at 40 MHz system clock. The processing time of CAM² is not influenced by image size when the image size is smaller than 128 × 128.

Processing times are summarized in Tables 1, 2, and 3. CA-based edge detection is about 12 times faster than that with DTCNN processing, CA-based hole filling is about 40 times faster, and CA-based center point detection is about 13 times faster. In total, CA-based processing is 16 times faster than DTCNN processing.
Daughter board
Mother board
Fig. 7. CAM² board with 128 × 128 CA cells.

Table 1. Edge detection processing time

<table>
<thead>
<tr>
<th></th>
<th>Number of cycles</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DTCNN</td>
<td>2350</td>
<td>58.750 μ</td>
</tr>
<tr>
<td>CA-based</td>
<td>189</td>
<td>4.725 μ</td>
</tr>
</tbody>
</table>

Table 2. Hole filling processing time

<table>
<thead>
<tr>
<th></th>
<th>Number of cycles</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DTCNN</td>
<td>162305</td>
<td>4.0576 m</td>
</tr>
<tr>
<td>CA-based</td>
<td>4416</td>
<td>0.1104 m</td>
</tr>
</tbody>
</table>

Table 3. Center point detection processing time

<table>
<thead>
<tr>
<th></th>
<th>Number of cycles</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DTCNN</td>
<td>458008</td>
<td>11.450 m</td>
</tr>
<tr>
<td>CA-based</td>
<td>3536</td>
<td>0.8840 m</td>
</tr>
</tbody>
</table>

5.2. Example of CA-based Image Processing

Fig. 8 shows an example of CA-based image processing for multiple object tracking. Various types of objects are processed, and finally the object center points are successfully extracted. This result is the same as the one achieved with DTCNN processing.

6. Conclusion

We described a CA-based image processing technique on CAM² for multiple object tracking. In place of DTCNN processing, CA-based edge detection, hole filling, and center point detection are proposed and successfully implemented on CAM². Experimental results demonstrated that CA-based processing takes less time than DTCNN processing. To develop a multiple object tracking system for actual use, we plan to study post-processing techniques, such as judgment for correspondence between objects, for use with the CA-based algorithm.

References

New Method of Sequential Symbolic Analysis of Biomedical Signals

Robert Stepien, Wlodzimierz Klonowski

Nalecz Institute of Biocybernetics and Biomedical Engineering Polish Academy of Sciences
4 Ks.Trojdena St., 02-109 Warsaw, Poland
Email: step@ibib.waw.pl, wklon@ibib.waw.pl

Abstract – We apply sequential symbolic analysis to the first range difference (‘derivative’) of signal. We encode it using two-elements alphabet - symbol ‘1’ when the signal is growing or remains unchanged and symbol ‘0’ when the signal is decreasing. After such symbolization sequences of the same symbol (‘1’ or ‘0’) give us information for how long periods the signal is monotonous (growing or decreasing) and indexes characterizing distributions turn out to be good characteristics of the analyzed signal.

1. Introduction

There are many different techniques for symbolic time series analysis, based on conversion of the amplitude of measured signal into a few possible symbols, corresponding to chosen amplitude ranges [1-6]. The effect of such coarse-graining (partition of the data space) is such that large-scale feature are captured, while noise is reduced. The choice of the data space partition affects the characteristics of symbolic description of the data. From this point, the techniques start to differ, for example the alphabet i.e. the set of used symbols may be different. Two-symbols alphabet {0,1} is often used. The greater the alphabet the more details of the original signal may be captured but the tradeoff is diminished reduction of noise. For two-symbols alphabet the data median or data mean are often used as the threshold for data space partition, but in non-stationary signals mean and median often abruptly change. We apply sequential symbolic analysis to the first range difference (‘derivative’) of signal. In this case value 0 is the natural threshold.

2. Methods

We calculate the first range difference of time series \( x(i) \) and we build series of symbols, \( s(i) \) (cf. [7]):

\[
x(i) = \begin{cases} 
1 & \text{if } [x(i+1) - x(i)] \geq 0 \\
0 & \text{if } [x(i+1) - x(i)] < 0
\end{cases} \quad i = 1, \ldots, (I-1)
\]

Symbol series is divided into \( K \) windows of width \( W \) symbols each. Sliding window technique is used for further analysis. If the shift of the window is smaller then the windows are overlap. We count monosequences \( N \times 0 \) (or \( N \times 1 \)) in consecutive windows; mono-sequence of length \( N \) \((N = 1, \ldots, W)\) is a homogeneous sequence containing only one type of symbol, ‘0’ or ‘1’. In such a way we obtain the number of mono-sequences consisting of \( N \) symbols ‘0’ in the \( k \)-th window, \( L_k \{N \times 0\} \) (or, equivalently, the number \( L_k \{N \times 1\} \)). We repeat this for all possible values of \( N \).

Using distribution of all detected monosequences in the \( k \)-th window, \( L_k \{N \times 0\} \), we calculate Shannon entropy [7]:

\[
S(k) = -\sum_{N} \frac{L_k \{N \times 0\}}{Q_k} \log \frac{L_k \{N \times 0\}}{Q_k}
\]

where \( Q_k = \sum L_k \{N \times 0\} \).

Next we normalize entropy.

\[
S_n = S / S_{\text{max}} \leq 1
\]

The entropy reaches maximum when all possible monosequences in the window have the same probability of occurrence

\[
S_{\text{max}} = -\log \left( \frac{1}{n} \right)
\]

where \( n \) is the maximum number of different, not repeated monosequences in the window.

Finally, the monotony is defined as

\[
M = 1 - S_n
\]

The monotony shows level of diversity of signal in single window and changeability of diversity in time (from window to window). In fact, diversity of intervals in signals is counted -- the intervals are monotonic parts of signals (amplitude is growing or amplitude is decreasing). The monotony is the measure of statistical repetition in the signal. It reaches minimum when the monosequences with same lengths do not repeat in the window. Monotony reaches maximum when the same monosequences (sinus signal) or a single monosequence (constant signal) fill up the whole window.
3. Results

Even comparison of number of monosequences like $L\{8x0\}$ on a single EEG channel demonstrates differences between normal EEG and pathological EEG that can be seen by naked eye (Fig. 1 and Fig. 2).

![Fig. 1. $L\{8x0\}$ calculated from a single-channel EEG for a normal case, for a case with weak ictal activity, and for a person with strong ictal activity.](image1)

![Fig. 2. $L\{8x0\}$ calculated from a single-channel sleep-EEG in a case of normal physiological sleep (upper curve) and in a case of a person suffering of insomnia (lower).](image2)

Such differences between normal and pathological cases may be seen even much better when monotony of sleep-EEG is considered (Fig. 3).

![Fig. 3. Monotony of sleep-EEG.](image3)

One can easily observed disturbances in quasiperiodicity of sleep. For deeper sleep stadiums, the values of monotony drop to level of average correlated noises. It means, that the brain in deeper sleep generates more various monosequences. This result is a consequences of the greater contribution of long monosequences in deeper stages of sleep. The long monosequences are more sensitive to noise so the brain ‘makes’ more longer sequences.

![Fig. 4. Monotony of sleep-ECG.](image4)

![Fig. 5. Monotony of sleep-EMG.](image5)
4. Conclusions

Sequential symbolic analysis of the first range difference of biosignals, in particular the method that makes use of a newly defined characteristic called monotony [7] are very promising in biomedical applications, for example in screening for pathological conditions.

Acknowledgments

We acknowledge support by statutory activity 4.4/st/10 of the Nalecz Institute of Biocybernetics and Biomedical Engineering Polish Academy of Sciences, Warsaw.

References


Speaker Identification with Voiced Speech Variability Modeling using Phase Space Reconstruction

Łukasz Bronakowski and Krzysztof Ślot

Technical University of Lodz, Institute of Electronics
ul. Wołczanska 211/215, Lodz, Poland
Email: lukasz.bronakowski@p.lodz.pl, kslot@p.lodz.pl

Abstract—The following paper examines a possibility of applying concepts and methods of chaotic system analysis for speech variability modeling in speaker identification task. The proposed descriptor comprise a set of parameters, which are derived for reconstructed phase spaces of voiced-speech segments. The proposed method for analysis of attractor convergence is based on correlation sum vectors, summarised by vector quantization technique. It has been shown that the presented approach appears to be promising means for speaker discrimination.

1. Introduction

In recent years one can observe an increasing interest in exploiting new features that can improve performance of biometric identity verification systems. Speaker recognition is one of the most prominent research directions within this field, as it pertains to speech, which is the basic behavioral biometric modality. Majority of commonly used methods in speaker recognition that have been proposed so far exploit short- and long-term spectral and energy features, such as Linear Prediction Coefficients - LPC, Mel-Frequency Cepstral Coefficients - MFCC and corresponding regression coefficients (delta-cestral, delta-delta-cestral coefficients) [1]. These features reflect only the behavioral aspect of speech production, which comprises processes involved in phone articulation.

Majority of vocal tract components are highly controllable, yielding a wide variability of possible ways of speech articulation, which is a drawback from the point of view of biometrics. However, a voluntary control over vocal tract is only partial in case of vocal folds and their excitation mechanisms, which are strongly determined by anatomical factors. Therefore, these components of the speech production system seem to naturally fit biometric purposes. Among the pool of features that are widely used in research on speaker recognition a characteristic that is directly related to vocal fold operations is the fundamental frequency of speech. However, this descriptor provides information on the speech-production aspect, which can be voluntarily controlled, revealing no evidence on more complex speech-phonation mechanisms. The phonation process, which is determined by anatomic structure of vocal folds and glottis, is generally omitted in speaker recognition methods.

The reason for it could be the difficulty and complexity of speech production, which involves aerodynamic, biomechanical, and acoustic factors that are still not fully understood. One of the open issues in speech signal exploration is a phenomenon of short-time variability in voiced speech production.

This phenomena can be interpreted as bifurcations and low-dimensional chaos [2], therefore non-linear theory can be applied to perform discussed problem. Descriptors of nonlinear speech behavior that account for vocal folds individual anatomy could provide a promising basis for identity resolution, as physiological features are known to be the most reliable biometric characteristics (as it is in case of retina, iris or fingerprints).

An objective of the following paper is to examine, whether short-time variability in speech-production can be exploited as a useful feature for speaker discrimination. We hypothesize that the speech-phonation aspect, reflected by speech signal non-stationary behavior, can be as important in speaker-modeling as the commonly used spectral characteristics of the human vocal tract. To verify the formulated hypothesis we propose a novel speech signal descriptor: a measure of convergence of reconstructed phase space attractors, derived for voiced speech segments.

The organization of the paper is as follows: the proposed descriptor of speech signal variability is introduced in section 2; speaker identification procedure is outlined in section 3 and its experimental evaluation is discussed in section 4.

2. Voiced-speech Variability Descriptor

Due to a limited rate of speech organ dynamics, one can assume that the speech signal is stationary within approximately 30 ms intervals. Small signal perturbations, at the order of no more than one percent, which are always present over this quasi-stationary background, heavily contribute to an individual appearance of speech. Furthermore, these perturbations can provide information about psycho-physiological state of the speaker, because they can originate from changes in tension of articulatory apparatus muscles and fluctuations of the air pressure exhaled from lungs during speaking [2].

Short-term speech variability is involuntary - one cannot
control a vocal folds tension at millisecond-long rates (as opposed to intentional control over changes of fundamental frequency over long-term intervals). As a result, it is an invaluable source of information on physical rather than behavioral properties of vocal tract, which can be exploited for identity recognition.

Speech signal analysis tools that have been used throughout the reported research for examining short-term perturbations have been adopted from nonlinear system theory and include Poincare mapping, reconstructed phase space and fractal dimension analysis. High speaker-discrimination potential that is offered by the present signal description perspective can be easily noticed from plots presented in Fig. 1, where reconstructed phase space is used to present the same utterance (a vowel “a”) spoken by three different speakers.

![Speech waveform](image)

Figure 1: Speech waveform of the vowel /a/ spoken by three different speakers and corresponding reconstructed phase spaces obtained from four consecutive speech frames.

2.1. Reconstructed Phase Space

The reconstructed phase space (RPS) can be considered as a plot of the time-lagged version of a signal. Structural patterns that occur in such phase space are commonly referred to as trajectories or attractors.

For time series \( x(n) \), where \( n = 1 \ldots N \) is a time index, each RPS trajectory point is a vector [3]:

\[
X = \begin{bmatrix} x_0 & x_{0+\tau} & \cdots & x_{0-(d-1)\tau} \\
\end{bmatrix}
\]

where \( \tau \) is a time lag and \( d \) is an embedding dimension. The RPS trajectory of the whole signal can be presented as a matrix composed of time-delayed vectors \( x_m \):

\[
X = \begin{bmatrix} x_1 & x_2 & \cdots & x_{d+1} \\
\end{bmatrix}
\]

RPS representation of a signal captures full dynamics of the underlying system and includes nonlinear information, which is not preserved by commonly used in speaker-recognition spectral-based speech-representation techniques. Several methods have been developed for estimation of phase-space trajectory distribution. These are e.g. Bayesian modeling of scatter of samples [4] or correlation dimension [5]. In the reported research, the correlation sum has been adopted as a statistical descriptor of the underlying attractor [2]. Correlation sum measures a trajectory divergence rate and is given by:

\[
C(\varepsilon) = \sum_{ij} \Theta(\varepsilon - ||x_i - x_j||)
\]

where \( \Theta(x) \) is a Heaviside function and sum indices refer to trajectory points subject to testing \((i)\) and points of their neighborhoods \((j)\). To get a more complete information on a structure of an attractor, we propose to characterize each frame \( m \) of an input signal using the following descriptor:

\[
C^m = [C(\varepsilon_1)^m, C(\varepsilon_2)^m \ldots C(\varepsilon_p)^m] \tag{4}
\]

where \( p \) is the adopted maximum size of neighborhood of interest and \( C(\varepsilon_k)^m \) is given by:

\[
C(\varepsilon_k)^m = \frac{1}{M N} \sum_{i=1}^{M} \sum_{j \neq i}^{N} \Theta(\varepsilon_k - ||x_i - x_j||) \tag{5}
\]

The sum \( C(\varepsilon_k)^m \) is computed at \( M \) trajectory samples per frame and \( N \) is the total number of trajectory points. The procedure of attractor descriptor derivation has been schematically depicted in Fig. 2. As a result of its application is derivation of a \( p \)-element vector (4), which will be used for speech signal frame representation in subsequent classification.

3. Speaker Identification Procedure

Data classification methodology that has been adopted for recognition (presented schematically in Fig. 3) assumes no temporal ordering of frames, which is a common approach for text-independent speaker recognition tasks. All vectors (4) extracted from frames of input training sequences (voiced-speech segments of sentences) uttered by a given speaker are used to build a reference model for this speaker. The model is a set of \( q \)-vectors that are codebook elements derived from vector-quantization of the corresponding distribution of training samples.
Each speaker $s$ is therefore represented by a unique set of $q$-vectors $\{\psi^s_i\}$. Speaker recognition is a procedure of confronting subsets of $L$ test vectors $\{c_i\}$ extracted from consecutive frames of a voiced-speech segment, with all available models (codebooks). A measure of fit is an overall distortion between codebook vectors of a given model and the test sequence vectors:

$$D_s = \frac{1}{L} \sum_{i=1}^{L} d(\tilde{\psi}^s_i, c_i)$$

(6)

The vector $\tilde{\psi}^s_i$ is the closest match between the test vector $c_i$ and all codebook elements from a model of the considered speaker $s$:

$$\tilde{\psi}^s_i = \min_{1 \leq j \leq q} d(\psi^s_j, c_i)$$

(7)

Euclidean distance has been used as a distance measure $d(\ldots)$ throughout the reported research.

4. Experimental Evaluation of the Proposed Procedure

The presented procedure has been verified using publicly available database CSLU: Speaker Recognition Version 1.1 [6]. Five different sentences, repeated twice by ten speakers (5 females and 5 males) during twelve recording sessions were subject to analysis. Each of the sentences comprised several voiced-speech segments, yielding sets of about 14000 voiced phonemes (frame sequences) per speaker. This set was evenly split into training and testing parts.

Pronunciation variability has been assessed for speech segments of about 30 ms length with 10 ms overlap. The time lag $\tau$ used for generate RPS is in general empirical, but we adopted a commonly used measure based on autocorrelation function: $C(\tau) = 0.5 \cdot C(0)$. The embedding dimension $d$ is estimated using the false neighbours method [7]. Eight values of the neighborhood size $\varepsilon$ were used for frame-descriptor construction: 0.001, 0.002, 0.004, 0.008, 0.016, 0.032, 0.064, 0.128 (the magnitude of input signal was normalized within the range $[-1 \cdots 1]$).

A value of $M = 10$ (equation (5)) has been arbitrarily adopted, which means that 10 points are selected along a trajectory as reference points for correlation sum computation. The points are selected in such a way that each of the
resulting between-point intervals contains approximately 10% of data points. To determine length of the trajectories, an estimation of the fundamental frequency is made according to the algorithm which is based on the computation of autocorrelation of speech in time-domain [8].

K-means method [9] is used to generate codebooks (speaker models) from correlation sum vectors. The number of means have been varied between 8 and 128. Speaker recognition performance shown as a function of a codebook-size (separately for male and female speakers) has been presented in Fig. 4. As it can be seen, the best results - over 80% correct recognition - have been obtained for a 32-element codebook. We consider this result to be a very good one, as we apply for speaker recognition a completely different basis than commonly used cepstral coefficients. For example, a use of MFCCs features in combination with GMM modeling and SVM classification [10] yields recognition rates between 66.37% (for 10s test and training speech duration) and 91.87% (for 10s test and 24min training speech duration) for the same CSLU database as in presented paper.

Speech signal descriptors that have been used in the reported research have little in common with spectral characteristics that dominate current techniques. As such, there exist an expectation to substantially increase speaker-recognition performance if both diverse signal analysis directions are appropriately combined.

Figure 4: Speaker-recognition performance, shown as a function of codebook-size.

5. Conclusion

The presented paper shows that short-time variability of speech is a source of important clues for speaker recognition. Although the analysis of the problem is still in its early stage - the variability descriptor can be certainly refined - quite good recognition rates can be attained.

The adopted approach to speaker recognition is complementary to the commonly used strategies. Therefore, a natural way of continuation of the reported research is to combine short-term variability with conventional ways of speech-signal characterization and to verify, whether such a combination could result in noticeable improvement in speaker recognition performance.

Acknowledgments

This research has been supported by Polish Ministry of Science and Higher Education under a grant no. N N515 343336.

References

Residue to Weighted Converter for the Quinary Moduli Set 
\{5^n – 2, 5^n – 1, 5^n\}

Hasan Amin Oseily† & Ali Massoud Haidar††
†Electrical Department, †† Computer Department, Faculty of Engineering
Beirut Arab University, Beirut-Lebanon
E-mail: sasoha@yahoo.com, ari@bau.edu.lb

Abstract— The residue number system (RNS) is a carry-free number system which can support high-speed and parallel arithmetic. One of the major issues in efficient design of RNS systems is the residue to weighted conversion which is an important issue concerning the utilization of RNS numbers in digital signal processing (DSP) applications. We present here an efficient design of residue to weighted converter for the newly introduced quinary moduli set \{5^n – 2, 5^n – 1, 5^n\}, based on mixed-radix conversion (MRC) algorithm. The proposed residue to weighted converter is adder-based and memory-less which can result in high-performance hardware. The proposed residue to weighted converter has better performance and also eliminates the use of multiplier, compared to the last work [10].

1. Introduction

The usage of Residue Number System (RNS) in Digital Signal Processing (DSP) applications has received considerable attention due to its attractive carry-free property which yields arithmetic processors that are inherently parallel, modular and fault isolating [1],[2],[3]. For successful application of RNS, data conversion must be very fast so that the conversion overhead doesn’t nullify the RNS advantages [3]. The residue number system (RNS) is a non-weighted number system which speeds up arithmetic operations by dividing them into smaller parallel operations. Since the arithmetic operations in each moduli are independent of the others, there is no carry propagation among them and so RNS leads to carry-free addition, multiplication and borrow-free subtraction [4]. The RNS is mostly used in encryption and decryption techniques for its advantages in the computation process. One of the major issues in efficient design of RNS systems is the residue to weighted conversion. The algorithms of residue to weighted conversion are mainly based on Chinese remainder theorem (CRT), mixed-radix conversion (MRC) [4] and new chinese remainder theorems (New CRTs) [5]. In addition to these, novel conversion algorithms [6] which are designed for some special moduli sets have been proposed. Multiple-valued logic (MVL) has been proposed as a means for reducing the power, improving the speed, and increasing the packing density of VLSI circuits [7]. In MVL, the number of discrete signal values or logic states extends beyond two. Arithmetic units implemented with MVL achieve more efficient use of silicon resource and circuit interconnections [8]. There is a clear mathematical attraction of using multiple-valued number representation in RNS. The modular arithmetic that is inherent in MVL can be match with modular arithmetic needed in RNS. The first MVL-RNS system was introduced by Soderstrand [9] to design a high speed Finite Impulse Filter (FIR). The residue to weighted converter proposed in [9] is based on chinese remainder theorem (CRT) and implemented with read-only memories (ROM’s). This converter is practical to implement small and medium RNS dynamic ranges and it is not appropriate for large dynamic ranges. This paper develops a two-level MRC algorithm for designing an efficient residue to weighted converter for the moduli set \{5^n – 2, 5^n – 1, 5^n\}. The proposed hardware architecture for residue to weighted converter has better performance in terms of area and delay since it is multiplier-free and memory-less in comparison with the residue to weighted converter proposed in [10].

2. Background

A residue number system is defined in terms of a relatively-prime moduli set \{P_1, P_2, ..., P_n\} that is greater common divisor GCD (P_i, P_j)= 1 for i ≠ j and i, j = 1,2,...,n . A weighted number X can be represented as X = (x_1,x_2, …, x_n), where

\[ x_i = X \mod P_i = \lfloor X \rceil P_i, \ 0 \leq x_i < P_i \]  \hspace{1cm} (1)

Such a representation is unique for any integer X in the range [0, M-1], where M=P_1.P_2....P_n is the dynamic range of the moduli set \{P_1,P_2,...,P_n\} [10].

Then, the equivalent representation of X=32 is (x_1, x_2, x_3) = (2, 2, 4). Addition, subtraction and multiplication on residues can be performed in parallel without any carry propagation among the residue digits. Hence, by converting the arithmetic of large numbers to a set of the parallel arithmetic of smaller numbers, the RNS representation yields significant speed up.

The algorithms of residue to weighted conversion are based mainly on Chinese remainder theorem (CRT) and mixed-radix conversion (MRC).
1) **Chinese Remainder Theorem**: by CRT, the number $X$ is calculated from residues by

$$X = \left| \sum_{i=1}^{k} (x_i n_i) / p_i M_i \right|_M$$

where $M_i = M / p_i$ and $N_i = |M_i^{-1}|_p$ is the multiplicative inverse of $M_i$ modulo $p_i$.

2) **Mixed-Radix Conversion**: the weighted number $X$ can be computed by

$$X = a_n P_n + \ldots + a_3 P_2 P_1 + a_2 P_1 + a_1$$

Where $a_i$s are called the mixed-radix coefficients and they can be obtained from the residues by

$$a_n = \left| \left( (x_n - a_1) / P_1 \right) / P_2 \right|_P$$

where $n > 1$ and $a_1 = x_1$.

3. **RNS with Moduli Set \{5^n – 2, 5^n – 1, 5^n\}**

In [10], a ternary moduli set \{3^n – 2, 3^n – 1, 3^n\} was introduced for RNS. Here we will introduce also the moduli set with quinary numbers set \{5^n – 2, 5^n – 1, 5^n\}. This moduli set contains pair-wise relatively prime and balanced moduli which can offer large dynamic range and fast internal RNS processing. Because of using of high radix ($r = 5$), this RNS can be simply realized in quinary-valued logic (QVL). Addition circuits for moduli set \{5^n – 2, 5^n – 1, 5^n\} can be obtained by using the same method of [11]. If we consider three numbers $A$, $B$, and $C$ as the residues in respect of the modulo $m$, then addition of these numbers in modulo $m$, can be performed as

$$A + B + C < m \Rightarrow A + B + C$$

$$A + B + C \geq m \Rightarrow A + B + C - m$$

In other words, if the result is greater than or equal to the moduli, we add it to the complement of the moduli and ignore the carry out.

**Example 1**: If we perform the addition operation on the residues of the two number $X(2,2,4)$ and $Y(0,4,3)$, we found that

$2+0 = 2$ because $2+0 < 3$ (then the residue is 2)

$2+4=6$ then the residue is $2+4-5 = 1$

$4+3=7$ then the residue is $4+3-7=0$

The final residue of addition is (2,1,0).

We propose a two-level conversion algorithm for the residue to weighted conversion of the moduli set \{5^n – 2, 5^n – 1, 5^n\}. In the first level we use a MRC block for combining the two residues. The second level consists of another MRC block combining the result of the first level with the third residue. Figure 4 shows the block diagram of the proposed residue to weighted converter.
Figure 4. Block diagram of the proposed converter

The following propositions are needed for the derivation of our algorithm.

**Proposition 1**: the multiplicative inverse of \(5^n-2\) modulo \(5^n\) is \(k_0 = (5^n-1)/2\).

**Proof**: it is clear that \(5^n - 2 \equiv s^n \mod 5^n\), so
\[
|k_0 \times (5^n - 2)|_{5^n} = \left|\frac{(5^n - 1)}{2} \times (5^n - 2)\right|_{5^n} = \frac{1}{2} \times -2 = 1
\]

**Proposition 2**: the multiplicative inverse of \(5^n - 1\) modulo \(5^n + 1\) is \(k_1 = -1\).

**Proof**: Since \(|s^n|_{5^n+1} = 1\) and \(|s^n-1|_{5^n+1} = -1\), we have
\[
|k_1 \times s^{n+1}|_{5^n+1} = -1 \times s^{n+1} = 1
\]

Consider the three-moduli set \(\{5^n - 2, 5^n - 1, 5^n\}\) and let the corresponding residues of the integer \(X = (x_1, x_2, x_3)\) be \(x_1, x_2, x_3\). Using the MRC conversion algorithm (5), \(Z\) can be calculated by

\[
Z = x_1 + (5^n-2) \times k_0 (x_3-x_1)
\]

Where
\[
|k_0 (5^n-2)|_{5^n} = 1
\]

Substituting the value of \(k_0\) from proposition (1) into (9) gives

\[
Z = x_1 + (5^n-2) \times ((5^n-1)/2) \times (x_3-x_1)
\]

The above equation can be rewritten as

\[
Z = x_1 + (5^n-2)T
\]

Where
\[
T = \left|\frac{(5^n-1)/2}{(x_3-x_1)}\right|_{5^n}
\]

We know that
\[
(5^n-1)/2 = 2x (5^n + 5^{n-1} + \ldots + 5^{1})
\]

Therefore (13) can be written as,

\[
T = 2x (5^n + 5^{n-1} + \ldots + 5^{1}) \times (x_3-x_1)
\]

The equation (12) can be rewritten as

\[
Z = x_1 + 5^n T - 2T
\]

Now, consider the moduli set \(\{5^n(5^n-2), 5^n-1\}\) and \(X = (Z, x_1)\). Using the derivation like before, \(X\) can be calculated by

\[
X = Z + 5^n (5^n - 2) \times k_1 (x_2 - Z)
\]

Where
\[
|k_1 \times 5^n (5^n - 2)|_{5^n} = 1
\]

By substituting the value of \(k_1\) from proposition 2, we have

\[
X = Z + 5^n (5^n - 2) \times Z - x_1
\]

So, (20) can be rewritten as

\[
X = Z + 5^n D - 5^n 2D = (Z + 5^n D) + 5^n (-D - D)
\]

Where
\[
D = \left|Z - x_1\right|_{5^n}
\]

Since \(Z\) is a \(2n\)-digit number, we can write

\[
D = \left|Z + Z_0 - x_2\right|_{5^n} = \left|Z_1 + Z_0 - x_2\right|_{5^n}
\]

Where \(Z_1\) and \(Z_0\) have digit level representation as

\[
Z_1 = (z_{2n-1} \ldots z_{0} Z_0)
\]

\[
Z_0 = (z_{0} \ldots z_{1} Z_0)
\]

**Example 2**: Given the moduli set \(\{5^n - 2, 5^n - 1, 5^n\}\) where \(n = 2\). The residue number \((1,4,2)\) is converted into its equivalent weighted number as follows: For \(n = 2\) the moduli set is \(\{23, 24, 25\}\). So, by substituting values in (11) and (20) we have

\[
Z = 1 + 23 \times 12 \times 25 = 277, X = 277 + 25 \times 23 \times 277 - 4 \times 24 = 5452
\]

To verify the result, we have

\[
x_1 = 5452 \times 23 = 1, x_2 = 5452 \times 2 = 4, x_3 = 5452 \times 25 = 2
\]

Therefore, the weighted number 5452 has RNS representation as \((1,4,2)\) in the RNS with moduli set \(\{23,24,25\}\).

### 4. Hardware Implementation

The MRC block of the first level is represented by equations (15)–(17), whereas equations (21) and (23) represent the MRC block of the second level. Details on the first-level and second-level are as follow.

1) **The First Level**: Equation (16) can be calculated by a regular \(n\)-digit quinary adder. Then, (15) is implemented

\[
V = \left|x_{i} x_{j}\right|_{5^n}
\]
by an $n$-digit quinary multi-operand adder which is consists of a $n$-digit quinary carry save adder (CSA) tree followed by a regular $n$-digit quinary adder. Finally, (17) can be calculated by a 2$n$-digit regular quinary adder. It should be noted that since $x_1$ is an $n$-digit number, no extra hardware is needed for computation of $x_1 + 5T$. The desired result can be obtained by concatenating $x_1$ with $T$. Figure 5(a) shows the hardware implementation of the first level of the residue to weighted converter.

2) The Second Level: Equation (23) can be performed by an $n$-digit modulo $(5^n - 1)$ quinary adder which is shown in Fig. 2. Calculation of equation (21) relies on an $n$-digit quinary adder followed by a 5$n$-digit regular quinary adder. Like before, since $Z$ is a 2$n$-digit number, no extra hardware is needed for computation of $Z + 5^nD$. Figure 5(b) shows the hardware implementation of the second level of the residue to weighted converter.

As shown in Figure 5(a) and (b), the proposed residue to weighted converter for the moduli set $\{5^n - 2, 5^n - 1, 5^n\}$ is multiplier-free and consists of quinary adders.

5. Results

The residue to weighted converter for the moduli set $\{3^n - 2, 3^n - 1, 3^n\}$ which is presented in [10], is based on direct implementation of the CRT algorithm and requires $n$-digit ternary multipliers and a modulo $(3^n - 2) (3^n - 1) (3^n)$ ternary adder for final reduction. So, as a result, the converter of [10] achieve long conversion delay and high hardware cost. But the larger modulo adder used in our converter is a modulo $(5^n - 1)$ adder and also the proposed design eliminates the use of multiplier. Therefore, the proposed residue to weighted converter has better performance than the residue to weighted converter of [10] due to the reduction of delay and hardware cost for more than 100% for the same converted number. For larger moduli set than $\{5^n - 2, 5^n - 1, 5^n\}$, the same procedures will be followed to conclude the conversion method but with some differences in the equation of $Z$. The proposed hardware that can implement this conversion method is FPGA (field programmable gate array)

6. Conclusion

In this chapter an efficient design of the residue to weighted converter for the moduli set $\{5^n - 2, 5^n - 1, 5^n\}$ is presented. The proposed hardware implementation of the residue to weighted converter is multiplier-free and memory-less, which can be efficiently implemented in VLSI. In comparison with the last residue to weighted converter for the moduli set $\{3^n - 2, 3^n - 1, 3^n\}$, the proposed design has better performance especially that quinary is easier than other systems for conversion into decimal (decimal is multiplier of quinary).

References

Basic Learning Characteristics of Digital Spike Maps

Takashi Ogawa† and Toshimichi Saito †
†EE Dept., Hosei Univ., Tokyo 8584, Japan

Abstract—This paper studies learning algorithm of the digital spike maps. The map is equivalent to a simple one-dimensional cellular automaton and can generate various digital spike-trains. In order to approximate a class of spike-trains, we present a learning algorithm with self-organizing function. Performing a basic numerical experiment, we have clarified that the map can learn a typical class of teacher signals. The results contribute to bridge between spiking neural systems and digital dynamical systems.

1. Introduction

Spiking signals play important roles in a variety of neural systems [1]-[9]. In biological/artificial neural systems, roughly speaking, spiking signals are generated by integrate-and-fire dynamics. Analysis of them is basic to understand information processing function in the brain [2][3]. Such neural systems can exhibit chaos, synchronization and related rich bifurcation [4]-[8]. Analysis of these phenomena is recognized as meaningful. The spiking signals are simple, low power and suitable for various engineering applications including image segmentation [9], A/D converters [10], UWB communication [11] [12] and neural prosthesis [13].

Inspired by such neural systems, a variety of spiking neurons have been presented [2] [7]. The digital spiking neuron (DN) is one of them [14] [15]. The DN is constructed by coupling plural shift registers and can exhibit rich digital spike-trains. Adjusting the wiring pattern between plural shift registers, the DN can learn (approximate) a class of teacher spiking signals.

This paper studies the digital spike map (DSM) and its learning algorithm. The DSM can be regarded as a simple class of cellular automata (CAs, [16] [17]) and can output various spike-trains. The learning algorithm is simple and includes a self-organizing function. Although there exists various candidates of teacher spike signals, we use the teacher signal constructed by the Izhikevich neuron [2]. The Izhikevich neuron is a dynamical systems inspired by biological spiking neuron. Performing a basic numerical experiment, we have clarified that the DSM can learn typical teacher signals of the Izhikevich neuron. A basic method is used to measure distance among spike-trains [14]. Please note the following for novelty and significance.

1. The DSM is a simple model to describe digital spike-trains and can be regarded as a variant of cellular automaton (CA) with rich dynamics/applications [16] [17]. The DSM can bridge between spiking neural systems and CA.

2. The DSM learning is simple, successive and self-organizing. It is different from the DN learning that is based on change of the wiring pattern and the genetic algorithm [14].

2. Digital Spike Map

In this section, we define the DSM. The domain \( I_D \) of the DSM is a set of lattice points in the unit circle \( I = [0, 1) \),

\[
I_D \equiv \{ \alpha_1, \alpha_2, \cdots, \alpha_M \}, \quad \alpha_i = \frac{2i - 1}{2M} \tag{1}
\]

where \( i = 1 \sim M \) and \( M \) is the number of the lattice points. The \( i \)-th lattice point \( \alpha_i \) is the center of the \( i \)-th subinterval:

\[
\Delta_i = \left[ \frac{i - 1}{M}, \frac{i}{M} \right) \tag{2}
\]

The DSM is a discrete map from \( I_D \) to itself:

\[
\varphi_{n+1} = Q(\varphi_n), \quad \varphi_n \in I_D \tag{3}
\]

where \( \varphi_n \) denotes the \( n \)-th digital spike phase where \( n \) is a positive integer. For an initial value \( \varphi_1 \), the DSM outputs a
sequence of lattice points
\[ \{\varphi_1, \varphi_2, \cdots, \varphi_N\}. \]
This phase sequence corresponds to the spike-train of \( N \) spike positions
\[ \{\tau_1', \tau_2', \cdots, \tau_N'\} = \{\varphi_1, \varphi_2 + 1, \cdots, \varphi_N + (N-1)\} \quad (4) \]
where \( \tau_k' \in [k-1, k) \), \( \varphi_k \) satisfies the condition \( \varphi_k = \tau_k' \mod 1 \). The spike-train is described by
\[ y(\tau) = \begin{cases} 1 & \text{for } \tau = \tau_n' \\ 0 & \text{otherwise} \end{cases} \quad (5) \]
where \( n = 1 \sim N \). We refer to \( y(\tau) \) as a digital spike-train (DST) hereafter. Fig.1 shows an example of the DSM and corresponding DST.

3. Learning algorithm

We introduce the learning algorithm. First, the teacher signal is a sequence of digital spike phases \( \{\theta'_1, \cdots, \theta'_N\} \), \( \theta'_n \in I_D \), corresponding to a spike-train in Eq. (4).

In the learning a pair of the phases is necessary. We define the pairs are presented successively: let the first pair of the digital spike phases \( (\theta'_1, \theta'_2) \) be presented at step \( s = 1 \) and let the \( s \)-th pair \( (\theta'_s, \theta'_{s+1}) \) be presented at step \( s \).

**Step 1:** Let \( s = 0 \). \( Q(a) \) is initialized by
\[ Q(a_i) = \frac{2i-1}{2M}, \quad i = 1 \sim M. \quad (6) \]

**Step 2** (update of the teacher signal.): The \( s \)-th pair of the teacher signal \( (\theta'_s, \theta'_{s+1}) \) is presented. Then the DSM of \( a_s \) is updated as shown in Fig.2, black circles:
\[ Q(a_s) = a_{s+1}. \quad (7) \]
We refer to the black circle as "winner". This output is permanent and can not change afterward. It causes a restriction for the teacher signals: for any phase \( a_i \in I_D \), the next phase \( a_{i+1} \) is given uniquely.

**Step 3** (Update of set of lattice points.): We refer to the winner before step \( s \) as "past winner". Let \( N_r \) (respectively, \( N_l \)) be the sets of lattice points between the winner \( a_s \) and the right-closest past winner \( a_r \) (respectively, the left-closest past winner \( a_l \)). For the lattice points,
\[ Q(a) = \begin{cases} F_r(a) & \text{for } a \in N_r \\ F_l(a) & \text{for } a \in N_l \end{cases} \quad (8) \]
where \( F_r \) (respectively, \( F_l \)) implies the linear interpolation between \( a_r \) and \( a_l \) (respectively, \( a_i \)). This update is temporal and \( Q(a) \) can change if some teacher signal is applied to the position of \( a \).

Note that this interpolation relates to self-organizing and the DSM can learn by insufficient teacher signals for \( s < M \). Examples are shown in Fig.2 (c), right side past winner \( a_r \) and left side past winner \( a_l \) are shown.

**Step 4:** Let \( s = s + 1 \). Go to **Step 2** and repeat until the maximum step limit \( s = N \).

Figs.2 and 3 show shapes of the DSM and DST in the learning process where the teacher signal is generated by DSM in Fig 1. We can see that the DST tends to mimic the teacher signal as the learning step \( s \) increases.

![Figure 2: Learning process and the DSM for \( M = 16 \). The black circles denote learned points "winner" and grey circles denote interpolation points.](image)

\( (a) \) s=1, \( (b) \) s=2, \( (c) \) s=3, \( a_r \) and \( a_l \) denote the right side past winner and the left side past winner, respectively. \( (d) \) s=15.

4. Numerical experiments

In order to evaluate the algorithm efficiency, we have performed a basic numerical experiment. The teacher signal is constructed by the spike-train of the Izhikevich neuron whose dynamics is described by
\[ \begin{align*}
\dot{v} &= 0.04v^2 + 5v + 140 - u + I \\
u &= a(bv - u)
\end{align*} \quad (9) \]
with the auxiliary after-spike resetting
\[ \begin{align*}
v &\leftarrow c \\
u &\leftarrow u + d
\end{align*} \quad (10) \]
where, \( v \) and \( u \) are dimensionless variables. After trial-and-errors, we fix the dimensionless parameters \( a = 0.1, b = 0.2, c = -53, d = 4 \) and \( I = 10 \). Fig. 5 (a) shows an example of spike-trains: this is between "Fast Spiking (FS)"
and "Intrinsically Bursting (IB)". The neuron fires and a spike is generated if \( \nu \geq 30 \). Let \( p_n \) be the \( n \)-th spike position, and let \( \Delta p_n \) be the \( n \)-th inter-spike-interval (ISI): \( \Delta p_n = p_{n+1} - p_n \) where \( n = 1 \sim N - 1 \). We use average of \( ISI \) of the teacher signal: \( ISI_{ave} \) for normalization of position. The teacher signal is normalized as \( \tau_n = p_n / ISI_{ave} \) and the phase is extracted \( \theta_n = \tau_n (mod 1) \) where time-axis is adjusted to satisfy \( \tau_1 \in [0, 1) \). In order to make the teacher signal, we quantize the spike phase: a sequence of spike phases: \( \{\theta_1, \cdots, \theta_N\} \) is converted to a digital spike phases: \( \{\theta'_1, \cdots, \theta'_M\} \); if the \( n \)-th teacher spike phase \( \theta_n \) is included in the \( i \)-th subinterval \( \Delta_i \), the \( n \)-th teacher spike phase is \( \theta'_n = \frac{\theta_n}{\Delta_i} \). As mentioned in the algorithm, the \( s \)-th pair of spike-phases is presented at learning step \( s \):

\[
(\theta'_s, \theta'_{s+1}), \quad s = 1 \sim N - 1
\] (11)

For simplicity, we consider a DST of 16 (= \( N \)) spikes for \( 0 < \tau < N \). Let \( \varphi_n \) denote the \( n \)-th spike phase of the DST \( y(\tau) \). The distance between the DST and the teacher signal is measured by

\[
S_{TD} = \frac{1}{M} \sum_{n=1}^{M} |\theta'_n - \varphi_n|
\] (12)

Note that \( S_{TD} = 0 \) if \( \theta'_n = \varphi_n \) for all \( n \).

Figs. 4 and 5 show shapes of the DSM and DST in the learning process where the teacher signal is generated by the Izhikevich neuron. Fig. 6 shows the approximation characteristic. Note that, even if the leaning is not terminated for \( s < 15 \), we have used the DSM in the learning process at step \( s \) to generate a DST of 16 spikes. We have measured the distance between the teacher signal and DST. In Fig. 6, we can see that the closest distance between teacher signal and DST decreases as \( s \) increases. For \( s = 9 \), the distance seems to converge to a small value:

\[
s = 9 \text{ seems to be sufficient for the DSM to approximate the teacher signal. Our algorithm is applicable to many other systems including the BN [19].}
\]

5. Conclusions

DSM and its basic learning algorithm are studied in this paper. The learning algorithm is simple and includes a self-organizing function. Performing basic numerical experiments, we have confirmed the DSM can approximate a typical class of the Izhikevich neuron even if the number of spikes is not sufficient.

Future problems are many including

(1) detailed analysis of learning process,
(2) learning wider class of spike-trains,
(3) relation between DSM and CA,
(4) engineering application and
(5) building hardware.


Figure 5: Learning process and the spike-trains. The teacher signal is constructed by the Izhikevich neuron, (a) teacher signal is constructed by the Izhikevich neuron, (b) analog spike position of the Izhikevich neuron (a), (c) digital teacher signal is based on (b), (d) \( s = 3 \), the distance between DST and the teacher signal is \( S_{TD} = 0.357 \), (e) \( s = 7 \), \( S_{TD} = 0.177 \), (f) \( s = 11 \), \( S_{TD} = 0.072 \), (g) \( s = 15 \), \( S_{TD} = 0.021 \).

Figure 6: Approximation characteristics for learning step \( s \). The plot mean the distance between the teacher signal is constructed by the Izhikevich neuron and DST.

References

Bifurcation Analysis of Coupled Nagumo-Sato Models

Kazutoshi Kinoshita†, Tetsushi Ueta††, Jun’ichi Imura‡, and Kazuyuki Aihara‡‡

†Department, of Information Science and Intelligent Systems, Tokushima University
††Center, for Advanced Information Technology, Tokushima University
‡Dept. of Mechanical and Environmental informatics, Tokyo Institute of Technology
‡‡Institute of Industrial Science, University of Tokyo
Email: {†kino, ††tetsushi}@is.tokushima-u.ac.jp, ‡imura@mei.iits.ac.jp, ‡‡aihara@sat.tu-yokai.ac.jp

Abstract—The Nagumo-Sato model is one of mathematical neuron models described by a piecewise linear difference equation. Since there is a conditional character which is discontinuous at the threshold value, the system can be classified as a hybrid dynamical system. Bifurcation phenomena are occurred by changing internal parameters and chaotic attractors are also given. The dynamical properties were exactly studied analytically.

In this paper, we investigate the bifurcations of diffusively-coupled Nagumo-Sato models. By using complementarity a shooting algorithm and brute-force method, complete bifurcation diagrams are obtained. In spite of the discontinuities inside the coupled system, our shooting method can solve bifurcation problems. A period-locking regions edged by border-collision bifurcations are found, and chaotic regions are distinguished by a tangent bifurcation. We discuss on changing bifurcation structures with parameter variations.

1. Introduction

A Nagumo-Sato model is one of mathematical neuron models[1, 2] written as follows:

\[ x_{k+1} = f(x_k) \] (1)

\[ f(x) = \begin{cases} 
    ax - b + 1 & (x < C) \\
    ax - b & (x \geq C)
\end{cases} \] (2)

This model is included in discrete-time piecewise affine systems. In these models, a flow as a solution of a difference equation is suddenly switched to another flow by getting across the system border. In control engineering field, if a state space has some non-smooth characteristics, the system is called a hybrid system. Here, discrete-time piecewise affine systems is categorized in hybrid systems. Thus the Nagumo-Sato model is also regarded as one of hybrid systems[3].

Bifurcation problems on limit cycles observed in a hybrid system is computable if the Poincaré section is defined on the manifold given by the condition of non-smoothness characteristics and a suitable transformation (projection) of the state into local coordinate system[4] is provided. However, as far as authors know, less discussion has been done on bifurcations in hybrid discrete systems.

Various coupled systems has been researched in neuron models. Especially, the gap junction is an important things of the structure of the neuron.

In this paper, we analyze bifurcations of a coupled system of Nagumo-Sato models in the sense of a natural extension for continuous-time gap junction systems as follows:

\[ \begin{align*}
    x_{k+1} &= f(x_k) + k(x_k + y_k) \\
    y_{k+1} &= f(y_k) + k(y_k + x_k)
\end{align*} \] (3)

We compute bifurcation parameter values by the shooting method, and a lot of border-collision bifurcations are found[5]. We compare bifurcations of the single Nagumo-Sato model with coupled Nagumo-Sato model, and discuss the characteristics of the model.

2. Bifurcations

2.1. Variational equations

Even the characteristics of the system contains non-smoothness such as hysteresis, break points, we can compute bifurcation parameter value numerically by using a shooting method unless the derivative of the characteristics is not defined, i.e., a smoothness of the characteristics for the state is not required. Rewrite the system (3) as

\[ x_j(k+1) = f_j(x_j(k)) \] (4)

where \( j = 1, 2, \ldots, m \), \( k \) is a discrete time, \( x(k) = (x(k), y(k)) \), and each \( f_j : \mathbb{R}^2 \rightarrow \mathbb{R}^2 \) is smooth. There are borders given by \( q_j(x) = 0 \) and it switches the system from \( f_j \) to \( f_{j+1} \) if the state \( x(k) \) is given by exceeding any border from \( x(k-1) \).

Let us denote the solution of Eq.(4) as \( x(k) = \varphi(x_0, k) \). It satisfies the initial condition \( x(0) = \varphi(x_0, 0) = x_0 \). The non-smoothness of Eq. (4) affects the derivative of the solution. In fact, a global derivatives cannot be obtained, however, according to the position of \( x \), we can split the system (4) into two linear systems. Similarly, the derivatives of period-\( n \) solutions, i.e., the solutions of the first variational equations can be evaluated by solving the following:

\[ \frac{\partial \varphi}{\partial x_0}(k+1) = \frac{\partial f}{\partial x} \cdot \frac{\partial \varphi}{\partial x_0}(k), \frac{\partial \varphi}{\partial x_0}(0) = I \] (5)
where \( \frac{\partial f^n}{\partial x} \) is a Jacobian matrix, but a careful evaluation is required to compute it since it depends on locations of periodic points thus it is not a fix matrix.

2.2. Border-collision bifurcations

Border-collision bifurcations are occurred regardless of local characteristics of fixed/periodic points. So borders-collision bifurcations are not obtained by the method based on eigenvalues. When the an orbit of the attractor hits with the border in the system, the border-collision bifurcation is occurred. The conditions are written as:
\[
\begin{align*}
\varphi(x_0, \lambda) - x_0 &= 0 \\
f(x_0, \lambda) - C &= 0
\end{align*}
\]
\hspace{1cm} (6)

The point of border-collision bifurcation \( x_0 \) and the parameter value \( \lambda \) can be obtained by solving Eq. (6).

2.3. Bifurcation diagrams

Figure 1 shows one dimensional bifurcation diagram and the maximum Lyapunov exponent of Eq. (3), when parameters are \( b = 0.5, C = 0.5 \) and \( k = 0.1 \), the initial value is \( (x, y) = (0.6, 0.1) \). In this figure, \( a \) is increased from \( a = 0 \), the orbit of \( x \) changes from period-2 to period-3, when it is occurred a bifurcation. However, Lyapunov exponent is not 0 when this bifurcation occurs, therefore it has a possibility of the border-collision bifurcation. In Fig. 1, zero Lyapunov exponent is happened when \( x = 0.8 \), and further increment of \( a \) there, chaos is observed. It is related with the tangent bifurcation.

Figure 1: One dimensional bifurcation diagram (red) and Lyapunov exponent (blue)

Figure 2 shows example of the situation of border-collision bifurcation in the vicinity of \( a = 0.61 \) in Fig. 1. Red points is a period-3 attractor and green points is a period-8 attractor. If the period-3 attractor moves along the arrow by a parameter variation, by hitting the border \( x = C \), we have suddenly the period-8 attractor. This border-collision bifurcation does not have a bistable situation.

We calculate several bifurcation curves Fig. 3. We observe that a lot of border-collision bifurcations are occurred from the Eq. (3) and they are computable solving Eq. (6). There is an big island of period-1 on the upper left, and period-2 on lower left. When \( a \) increases, it is occurred border-collision bifurcations and consists of high-periodic areas. It is noteworthy that border-collision bifurcations \( B_{31}, B_{33}, B_{32}, B_{35} \) do not terminate on the tangent bifurcation \( T \), and they lie on chaotic area. Thus they are regarded as bifurcations for unstable periodic points.

3. Characteristics

We compare the bifurcations of single Nagumo-Sato model with coupled it. Figure 4 is bifurcation diagram of Nagumo-Sato models, there are (a) single and (b) coupled, and Fig. 5 is enlarged diagram of Fig. 4 that the vicinity of tangent bifurcation. In Fig. 4, we use brute-force method to compute bifurcation diagrams, because hi-periodic fields have been overcrowded. In this figure, the horizontal and vertical axes are \( a \) and \( b \) respectively, where \( 0.0 < a < 1.0, 0.0 < b < 1.0 \). The blue region indicates the parameter region which shows a period-1 (fixed point) in the state space. In the same way, the red region is period-2, the green is period-4, and the black region shows a chaotic area, if over...
period-13 when coloring by color of the remainder divided by 13.

In Figure 4 and 5, as for the coupled model, the total number of border-collision bifurcations decreases compared with the single model, because a lot of border-collision bifurcations are occurred by parameter $a = 0.8$, however the tangent bifurcation is occurred with $a = 0.8$. Where the Jacobian matrix of Eq. (3) are:

$$
\frac{\partial f_1}{\partial x_1} = \begin{pmatrix} a + k & -k \\ -k & a + k \end{pmatrix}, \quad \frac{\partial f_2}{\partial x_2} = \begin{pmatrix} a + k & -k \\ -k & a + k \end{pmatrix},
$$

and the eigenvalue of Eq. (7) is:

$$
\mu = a + k \pm \sqrt{(a + k)^2 - a^2 + 2ak}.
$$

The position of tangent bifurcation is decided depending on the coupling factor $k$, because is decided by the eigenvalue $\mu$. Figure 6 is the bifurcation diagram that parameter $k = 0.2$ and $k = 0.4$, and Fig. 7 is the bifurcation diagram that the horizontal and vertical axes are $a$ and $k$. The tangent bifurcation where is border of the chaotic area is changed as coupling factor $k$.

Figure 8 is the phase portrait of the chaotic area. In Fig. 8 (a) when parameter $a = 0.9$, the chaos appears to both sides across $y = x$, however (b) when $a = 1.1$, the chaos appears that is wide range and unsteady.

4. Conclusions

We compute bifurcation sets of the coupled Nagumo-Sato model by the shooting algorithm as the hybrid system, and we show a lot of border-collision bifurcations and the tangent bifurcation. It is clarified that the size of the chaotic region is radically depended on the value of parameter $a$.

Acknowledgement

This research is supported by the Japan Society for the Promotion of Science (JSPS) through its “Funding Program for World-Leading Innovative R&D on Science and Technology (FIRST Program).”

References

Figure 5: Enlarged diagram of bifurcation diagram, the vicinity of tangent bifurcation. (a) single model, $0.7 < a < 1.1$, $0.3 < b < 0.7$, $C = 0.5$. (b) coupled model, $0.5 < a < 0.9$, $0.3 < b < 0.7$, $C = 0.5$, $k = 0.1$.

Figure 6: Bifurcation diagrams (a) $0.0 < a < 1.0$, $0.0 < b < 1.0$, $k = 0.2$. (b) $0.0 < a < 1.0$, $0.0 < b < 1.0$, $k = 0.4$.

Figure 7: Bifurcation diagram, $0.0 < a < 1.0$, $0.0 < k < 0.5$, $b = 0.5$ $C = 0.5$.

Figure 8: Phase portraits of chaos fields (a) parameters are $a = 0.9$, $b = 0.5$, $C = 0.5$, $k = 0.1$ (b) $a = 1.1$ other is the same.


Synchronization Phenomena of Globally Coupled Logistic Maps with Time-Varying Parameters

Hironori Kumeno†, Yoshifumi Nishio† and Daniele Fournier-Prunaret‡

†Dept. of Electrical and Electronic Eng., Tokushima University, 2-1 Minami-Josanjima, Tokushima, 770-8506 JAPAN
‡LATTIS-INS, Universite de Toulouse 135 avenue de Rangueil, 31077 Toulouse, France
Email: {kumeno, nishio} @ee.tokushima-u.ac.jp, daniele.fournier@insa-toulouse.fr

Abstract—Synchronization phenomena in globally coupled logistic maps whose parameters are forced into periodic varying are investigated when four and five maps are coupled. Various synchronization phenomena are observed by choosing a coupling intensity in both cases. The observed synchronization phenomena are fall into five general categories, which are asynchronous, self-switching phenomenon of synchronization, coexistence phenomena of synchronization states, synchronization of the total number of the coupled maps minus one map and synchronization of all the maps.

1. Introduction

Synchronization is one of the fundamental phenomena in nature, and one of typical nonlinear phenomena. Therefore, studies on synchronization phenomena of coupled systems are extensively carried out in various fields, physics [1], biology [2], engineering and so on. Parametric excitation circuit is one of resonant circuits, and it is important to investigate various nonlinear phenomena for future engineering applications. In a simple oscillator including parametric excitation, Ref. [3] reports that the almost periodic oscillation occurs if nonlinear inductor has saturation characteristic. Additionally the occurrence of chaos is referred in Refs. [4] and [5]. The network of chaotic elements can be modeled by a system of coupled one-dimensional maps. Behavior generated in coupled system of chaotic one-dimensional map is investigated in Refs. [6]-[8]. In particular, Coupled Map Lattice (CML) and Globally Coupled Map (GCM) are well known as mathematical models in discrete-time system. The research into CML and GCM is important for not only modeling of nonlinear systems of multiple degree of freedom but also application to biological networks and engineering. In the past we have observed effects of parametric excitation of coupled van der Pol oscillators [9]. Previously, we investigate synchronization phenomena in three coupled logistic maps involving parametric force [10]. In this study, we propose now to investigate synchronization phenomena generated by a larger number of coupled maps with globally coupling. A typical scheme for global coupling is given by:

\[ x_i(n+1) = (1 - \varepsilon) f[x_i(n)] + \frac{N}{i=1} \sum_{j=1}^{N} f[x_j(n)] \]

where \( \varepsilon \in [0, 1] \) is the coupling intensity. The globally coupled maps are a scheme such that an average number of all the maps affect each of the map. The one-dimensional map used in this study is a logistic map, since the map can be described by a simple discrete equation. Mathematically, the logistic map is written as

\[ x(n+1) = ax(n)(1 - x(n)). \]

In this study, we investigate synchronization phenomena in the globally coupled logistic maps whose parameters are forced into periodic varying when four or five maps are coupled. The paper is organized as follows. In the next section, we present the parametrically forced logistic map. Synchronization phenomena observed in the globally coupled maps are in Section 3. The last section is devoted to the conclusion.

2. Parametrically forced logistic map

A parametrically forced logistic map used in this study is described as:

\[ x(n+1) = \alpha_f(n)x(n)(1 - x(n)), \]

and

\[ \alpha_f(n) = \begin{cases} \alpha_1, & \text{for each even value of } "n" \\ \alpha_2, & \text{for each odd value of } "n" \\ (n = 0, 1, 2, ...) \end{cases} \]

where \( \alpha_f(n) \) is a term of the parametric force and time-varying. In this system, two kinds of parameters, \( \alpha_1 \) and \( \alpha_2 \), are alternately replaced every update. Figure 1 shows an example of a return map of the parametrically forced logistic maps. For the original logistic map, two-periodic solution is observed for \( a = 3.0 \), while, three-periodic solution is observed for \( a = 3.83 \). These two solutions are
periodic, whereas in the logistic map involving parametric force, a solution is chaotic as shown in Fig. 1 when the parameters $a_1$ and $a_2$ are set 3.0 and 3.83, respectively. Namely, chaotic solution can be observed in the combination of two parameters that generate two kinds of periodic solutions. In the following, the parameter values are fixed as $a_1 = 3.0$ and $a_2 = 3.83$.

Figure 1: Return map of parametrically forced logistic map for $a_1 = 3.0$ and $a_2 = 3.83$.

3. Synchronization

Synchronization phenomena generated in the coupled logistic map involving parametric force are investigated for one control parameter $\varepsilon$ which is coupling intensity when four and five maps are coupled. In the following computer calculations, the logistic map parameters are fixed as $a_1 = 3.0$ and $a_2 = 3.8$.

3.1. Four maps case

In this subsection, we consider the case of $N = 4$, namely four parametrically forced logistic maps are coupled. Figure 2 shows maximum Lyapunov exponents, which calculates how complex trajectory of the solution is, for one control parameter $\varepsilon$. Various synchronization phenomena are observed for $\varepsilon$. Examples of synchronization phenomena observed in the four coupled maps are shown in Figs. 3 and 4.

First, the Lyapunov exponent is positive and all the maps behave chaotic when $\varepsilon$ is zero, namely all the maps are not coupled. When $\varepsilon$ is small, the maps are not synchronized. Increasing $\varepsilon$, a self-switching phenomenon of synchronization on four coupled maps is observed as shown in Fig. 3 when $\varepsilon$ is set around 0.045. The phenomenon is that three among the four maps are synchronized and the combination of the synchronized pair changes with time. Figure 3 shows time series of differences between two maps. Areas where the amplitudes of the time series are small correspond to in-phase synchronization in the figure. In Fig. 3, firstly, map 1, map 3 and map 4 are synchronized. However, after a while, the synchronous state breaks up and map 1, map 2 and map 4 are synchronized. As seen above, the synchronous states switch with time. More increasing $\varepsilon$, the Lyapunov exponent becomes around zero or negative. In some parts of $\varepsilon$ between 0.05 and 0.08, there exist some lines of Lyapunov exponent. It means that the synchronization state is multi stable. Coexistence phenomena of various synchronization states are observed for the values of $\varepsilon$ as shown in Figs. 4(a-1) and (a-2). Two pairs of two chaotic maps are synchronized in Fig 4(a-1). While, three among the four chaotic maps are synchronized in Fig 4(a-2). These two synchronization state are observed for the same parameters, although initial values are different. The synchronization states depend on the initial values. In the region of the multi stable state, other coexistence phenomena of synchronization states are confirmed as;

- coexistence of three kinds of synchronization states that are synchronization of two among the four chaotic maps, synchronization of two pairs of two periodic maps and synchronization of three among the four periodic maps,
- coexistence of two kinds of synchronization states that are synchronization of two pairs of two periodic maps and two pairs which are synchronization of two periodic maps and quasi-synchronization of two maps,
- coexistence of two kinds of synchronization states that are synchronization of two among the four chaotic maps and synchronization of two pairs of two chaotic maps.

Figure 2: Lyapunov exponents in globally coupled parametrically forced logistic maps for $a_1 = 3.0$ and $a_2 = 3.83$. Horizontal axis: $\varepsilon$. Vertical axis: $\lambda$.

Figure 3: Time series of differences between two maps when a self-switching phenomenon of the synchronization are observed. $\varepsilon = 0.045$, $a_1 = 3.0$ and $a_2 = 3.83$. 
Next, more increasing $\epsilon$, the Lyapunov exponent becomes positive again. Synchronization of three among the four chaotic maps is observed when $\epsilon = 0.160$ as shown in Fig. 4(b). Finally, when $\epsilon$ is equal and over 0.200, all the maps are synchronized as shown in Fig. 4(c).

### 3.2. Five maps case

In this subsection, we consider the case of $N = 5$. Figure 5 shows maximum Lyapunov exponents for one-control parameter $\epsilon$. Various synchronization phenomena are observed for $\epsilon$. Examples of synchronization phenomena observed in the five coupled maps are shown in Figs. 6 and 7.

When $\epsilon$ is small, the maps are not synchronized. Increasing $\epsilon$, a self-switching phenomenon of synchronization on five coupled maps is observed as shown in Fig. 6 when $\epsilon$ is set around 0.045. The phenomenon is that three among the five maps are synchronized and another two maps are synchronized and the combination of the synchronized pair changes with time. Figure 6 shows time series of differences of $x(n)$ between two maps. More increasing $\epsilon$, the Lyapunov exponent becomes around zero or negative. In some parts of $\epsilon$ between 0.05 and 0.095, there exist some lines of Lyapunov exponent. Coexistence phenomena of various synchronization states are observed in the parts of $\epsilon$ as shown in Figs. 7(a-1) and (a-2). Three among the five periodic maps are synchronized in Fig 7(a-1). While, Three among the five periodic maps are synchronized and another two periodic maps are synchronized in Fig 7(a-2). In the region of the multi stable state, other coexistence phenomena of synchronization states are confirmed. It confirms the existence of some kinds of coexistence phenomena which are constructed as combinations of following synchronization states;

- quasi-synchronization of three among the five maps and quasi-synchronization of two other maps,
- quasi-synchronization of three among the five maps and synchronization of two other maps,
- synchronization of three among the five maps and synchronization of two other maps,
- two pairs of quasi-synchronization of two maps,
- two pairs of synchronization of two maps,
- synchronization of two maps and quasi-synchronization of two maps,
- two pairs of the synchronization of two maps and quasi-synchronization of one of the two pairs and another map,
- two pairs of the synchronization of two maps and quasi-synchronization of the two pairs,
- quasi-synchronization of three among the five maps.

Next, more increasing $\epsilon$, the Lyapunov exponent becomes positive again. Synchronization of four among the five chaotic maps is observed when $\epsilon = 0.190$ as shown in Fig. 7(b). Finally, when $\epsilon$ is equal and over 0.210, all the maps are synchronized as shown in Fig. 7(c).

### 4. Conclusion

In this study, we have investigated synchronization phenomena in globally coupled logistic maps whose parameters are forced into periodic varying when four and five maps are coupled. Various synchronization phenomena can be observed by choosing a coupling intensity in both the four coupled maps case and the five coupled maps case. The observed synchronization phenomena fall into five general categories, which are asynchronous, self-switching phenomenon of synchronization, coexistence phenomena
Figure 5: Lyapunov exponents in globally coupled parametrically forced logistic maps for $\alpha_1 = 3.0$ and $\alpha_2 = 3.83$. Horizontal axis: $\varepsilon$. Vertical axis: $\lambda$.

Figure 6: Time series of differences between two maps when a self-switching phenomenon of the synchronization are observed. $\varepsilon = 0.045$, $\alpha_1 = 3.0$ and $\alpha_2 = 3.83$.

Figure 7: Synchronization of five maps. $\alpha_1 = 3.0$ and $\alpha_2 = 3.83$. (a) $\varepsilon = 0.050$. (b) $\varepsilon = 0.190$. (c) $\varepsilon = 0.210$.

of synchronization states, synchronization of the total number of the coupled maps minus one map and synchronization of all the maps.

References

Synchronization in Coupled Maps with Triangular Networks

Yoko Uwate\(^\dagger\) and Yoshifumi Nishio\(^\dagger\)

\(^\dagger\)Dept. of Electrical and Electronic Engineering, Tokushima University
2-1 Minami-Josanjima, Tokushima, Japan
Email: \{uwate, nishio\}@ee.tokushima-u.ac.jp

Abstract—In this study, we investigate synchronization phenomena when the 2-dimensional maps based on neuron model are coupled with triangular network property. Furthermore, the difference of synchronization obtained from the coupled maps and the coupled oscillators is compared.

1. Introduction

Generally, complex dynamical phenomena can be observed in networks formed by many elements with nonlinearity. Coupled Map Lattice (CML) has proposed by Kaneko and Bunimovich [1]-[5], to use as general models for the complex high-dimensional dynamics, such as biological systems, networks in DNA, economic activities, neural networks, and evolutions. We can observe the spatio-temporal patterns in CML. Moreover, coupled oscillatory systems can also produce interesting phase patterns, including wave propagation, clustering, and complex phase patterns. It is very important to make clear this mechanism of these spatio-temporal patterns for understanding complex patterns observed in natural science. Usually, the chaotic maps are used for CML and many interesting spatio-temporal patterns were observed.

Recently, a discrete map for spiking-bursting neural behavior was proposed by Rulkov [6], [7]. Rulkov map (see. Fig. 1) in the form of a two-dimensional map can be useful for understanding the dynamical mechanism of oscillators in the large scale networks. And Rulkov map produce spiking-bursting behavior such as real neurons.

In this study, we investigate synchronization phenomena when the 2-dimensional maps based on neuron model are coupled with triangular network property. Furthermore, the difference of synchronization obtained from the coupled maps and the coupled oscillators is compared.

2. Coupled Maps with Triangular Networks

The several types of coupled maps with triangular networks are shown in Fig. 2.

We consider a chain of coupled maps:

\[
\begin{align*}
    x_{i,n+1} &= f(x_{i,n}x_{i,n-1}, y_{i,n}) \\
    &+ \frac{1}{2}g(x_{i+1,n} - 2x_{i,n} + x_{i-1,n}),
    \\
    y_{i,n+1} &= y_{i,n} - \mu(x_{i,n} + 1) + \mu \sigma_i
\end{align*}
\]

Figure 1: Rulkov map. The dashed line illustrates a super-stable cycle \(P_k\). The stable and unstable fixed points of the map are indicated by \(x_{ps}\) and \(x_{pu}\), respectively.

(a) \(N = 3\).  (b) \(N = 4\).

(c) \(N = 5\).  (d) \(N = 6\).

(e) General network model with triangular networks.

Figure 2: Several types of coupled maps with triangular network property.
\[ + \mu \frac{1}{2} g(x_{i+1,n} - 2x_{i,n} + x_{i-1,n}), \]
\[ i = 1, \ldots, N, \]

where \( x \) and \( y \) are the fast and slow dynamical variables, respectively. \( \mu = 10^{-3} \) and \( \sigma \) are the parameters of the individual map and \( g \) is the coupling. The function \( f() \) has the following form:

\[ f(x_n, y_n) = \begin{cases} 
\alpha / (1 - x_n) + y_n, & x_n \leq 0, \\
\alpha + y_n, & 0 < x_n < \alpha + y_n \\
-1, & x_n \geq \alpha + y_n \text{ or } x_{n-1} > 0, 
\end{cases} \]

In this simulations, we take \( \alpha = 3.5 \) and \( \sigma_i \) is set for randomly distributed in the interval [0.15:0.16].

2.1. Synchronization for \( N = 3 \)

First, we consider the simplest model as shown in Fig. 2(a). The three maps are coupled as ring topology. In this coupled maps model, three phase synchronization can be observed when the copuling strength is set to \( g = -0.029 \).

![Figure 3: Three-phase synchronization (g = -0.029).](image)

2.2. Synchronization for \( N = 4 \)

Next, the model of coupled maps with two triangular networks as shown in Fig. 2 (b) is considered. We observe two types of synchronization states dependence on the value of \( \sigma \). When \( \sigma \) is fixed with 0.24, two pair of three phase synchronization is obtained. The time wave forms of each map are shown in Fig. 4. From this figure, the first, the second and the third maps are synchronized with three-phase state. Also, the first, the second and the fourth maps synchronize at three-phase. Furthermore, we confirm that the third and the fourth maps are synchronized with in-phase state.

![Figure 4: Two pair of three-phase synchronization (\( \sigma = 0.24 \)).](image)

While, in the case of \( \sigma = 0.10 \), in/anti phase synchronization can be observed as shown in Fig. 5. The first and the second maps are synchronized at the in-phase state and the other combinations are synchronized at the anti-phase state.

![Figure 5: In/Anti-phase synchronization (\( \sigma = 0.10 \)).](image)

3. Comparison with Coupled Oscillatory System

In this section, we compare synchronization phenomena between the coupled maps and the coupled oscillators.
3.1. Circuit Model for \( N = 3 \)

The circuit model of three coupled van der Pol oscillator as ring topology is shown in Fig. 6. This circuit model corresponds to the three coupled maps shown in Fig. 2 (a).

The normalized circuit equations governing the circuit are expressed as

\[
\begin{align*}
\frac{dx_1}{dt} &= \epsilon \left( 1 - \frac{1}{3} x_1^2 \right) x_1 - (y_{a1} + y_{b1}) \\
\frac{dy_{a1}}{dt} &= \frac{1}{2} \left( x_1 - \eta y_{a1} - \beta y (y_{a1} + y_{b1}) \right) \\
\frac{dy_{b1}}{dt} &= \frac{1}{2} \left( x_1 - \eta y_{b1} - \gamma (y_{a3} + y_{b1}) \right)
\end{align*}
\]

[Second oscillator]

\[
\begin{align*}
\frac{dx_2}{dt} &= \epsilon \left( 1 - \frac{1}{3} x_2^2 \right) x_2 - (y_{a2} + y_{b2}) \\
\frac{dy_{a2}}{dt} &= \frac{1}{2} \left( x_2 - \eta y_{a2} - \beta y (y_{a2} + y_{b2}) \right) \\
\frac{dy_{b2}}{dt} &= \frac{1}{2} \left( x_2 - \eta y_{b2} - \gamma (y_{a3} + y_{b2}) \right)
\end{align*}
\]

[Third oscillator]

\[
\begin{align*}
\frac{dx_3}{dt} &= \epsilon \left( 1 - \frac{1}{3} x_3^2 \right) x_3 - (y_{a3} + y_{b3}) \\
\frac{dy_{a3}}{dt} &= \frac{1}{2} \left( x_3 - \eta y_{a3} - \beta y (y_{a3} + y_{b3}) \right) \\
\frac{dy_{b3}}{dt} &= \frac{1}{2} \left( x_3 - \eta y_{b3} - \gamma (y_{a2} + y_{b3}) \right)
\end{align*}
\]

where

\[
\begin{align*}
t &= \sqrt{LC} \tau, \quad v_k = \sqrt{\frac{g_1}{3g_3}} x_k, \\
i_{ak} &= \sqrt{\frac{g_1}{3g_3}} \sqrt{\frac{C}{L}} y_{ak}, \quad i_{bk} = \sqrt{\frac{g_1}{3g_3}} \sqrt{\frac{C}{L}} y_{bk}, \\
\epsilon &= g_1 \sqrt{\frac{L}{C}}, \quad \gamma = R \sqrt{\frac{C}{L}}, \quad \eta = r_m \sqrt{\frac{C}{L}}.
\end{align*}
\]

In this equations, \( \gamma \) is the coupling strength and \( \epsilon \) denotes the nonlinearity of the oscillators. For the computer simulations, \( \gamma \) and \( \epsilon \) are fixed with 0.1, 0.1, respectively. For the computer simulations, we calculates Eqs. (4)-(6) using a fourth-order Runge-Kutta method with the step size \( h = 0.005 \). Figure 7 shows the three time wave forms obtained by each oscillator. We can see that the three oscillators are synchronized at the three-phase state.

3.2. Circuit Model for \( N = 4 \)

Here, we consider the two coupled triangle oscillatory networks sharing a branch. The circuit model of two coupled triangle oscillatory networks sharing the branch is shown in Fig. 8.

The parameters of this circuit model are fixed as \( \epsilon = 0.1, \gamma = 0.1, \eta = 0.0001 \).
Figure 9 shows the time wave form of the voltage charged at the capacitance of each oscillator. From this figure, we can see that the first and the second oscillators are synchronized at in-phase (phase difference: 0 degree). While, the other combination oscillators synchronize with anti-phase (phase difference: 180 degree). Furthermore, the amplitude of between the first/second and the third/fourth oscillators has small difference. The phase plane of each combination oscillator is shown in Fig. 10.

5. Conclusions

In this study, we have investigated synchronization phenomena when the 2-dimensional maps based on neuron model are coupled with triangular network property. Furthermore, the difference of synchronization obtained from the coupled maps and the coupled oscillators was compared.

Acknowledgment

This work was partly supported by JSPS Grant-in-Aid for Scientific Research 22500203.

References

Analysis of Several Spatio-Temporal Phase Patterns in Coupled Chaotic Maps by Varying Coupling Strength

Masahiro Wada and Takuya Fukuda

Dept. of Intelligence and Informatics, Konan University
8–9–1, Okamoto, Higashinada-ku, Kobe 658-8501, Japan
Email: wada–m@konan–u.ac.jp, mn924009@center.konan–u.ac.jp

Abstract—Coupled chaotic map systems are attracted as a good model for representation of several phenomena in the real world. In the previous studies related with several coupled network models, the value of the coupling strength was almost a constant value and fixed. In this study, the value of the coupling strength is varying in dependence on state which is provided by a Gaussian function. Several spatio-temporal phase patterns by their complex dynamics could be confirmed. Two types of coupled network system were considered, and pattern dynamics was investigated.

1. Introduction

Pattern dynamics and mechanism of organization in several complex system attract many researchers’ attention as a good model which can realize the complicated phenomena in the real world. Coupled chaotic system and its dynamics can yield a wide variety of complex and strange phenomena. The coupled systems existing in nature exhibit great variety of phenomena such as complex mechanisms for all of the systems in the natural fields or in the universe. These phenomena can be found in a metabolic network, a human society, the process of a life, self-organization of neuron, a biological system, an ecological system and so many nonlinear systems. Among the studies on such coupled systems, many interesting researches relevant to the spatio-temporal chaos phenomena on the coupled chaotic systems have been studied until now, e.g. mathematical model in one- or two-dimensional network investigated earnestly by Kaneko[1]-[3], and found in physical circuit model[4]. Moreover, research of complicated phenomena and emergent property in the coupled cubic maps on 2-dimensional network system has been also reported[5]. The studies of coupled map lattice (CML), globally coupled maps (GCM) and so many studies related with such complex systems provided us tremendous interesting phenomena. We had also reported the research on spatio-temporal phase patterns in coupled maps using a fifth-power function[6]-[8], in which it has been carried out in the unique case. We had reported a research for complex network by non-uniform coupling strength as one of examples[9]. However, many coupled chaotic systems have wide variety of features and furthermore its dynamics is also expected to be applied much engineering applications, there are many problems which should be solved in large scale coupled network systems by their complexity.

In this study, analysis of several spatio-temporal chaotic behavior in coupled maps with varying coupling strength by state of neighbors will be presented. The chaotic map which has been governed by a third power polynomial function is properly selected as a chaotic cell. We consider the model which chaotic cells are mutually connected to neighbors as a ring or 2-dimensional network with an arbitrary coupling strength. In the almost previous studies related with CML and GCM, the value of the coupling strength was a unique value and fixed, and also using the same coupling strength. In this study, contrary to the previous them, we adopt a value of the coupling strength which is provided in dependence on each state by a Gaussian function as a non-uniform network. Several phase patterns made from complex dynamics will be shown. Then, we show some phenomena which spatio-temporal chaos, complex behavior and several phase patterns can be confirmed in the proposed coupled systems.

2. Chaotic Maps

Chaotic maps are generally used for several approaches to investigate complex dynamics and several phenomena on coupled network systems. Especially, the logistic map and other types of chaotic maps such as a cut map, a circle map, a tent map, a cubic map are well known and popular. Obviously, it is necessary to have a lot of equilibrium points with the complex phenomena that corresponds to the natural world. Let us consider an n-th order polynomial function. The n-th order polynomial function is normally written as follows.

$$ f(x) = \sum_{k=1}^{n} a_k x^k $$

(1)

where $a_k$ is the characteristic parameter which can determine for their chaotic feature. If it is needed to adopt the map with respect to the origin, odd-numbered coefficients $a_k$ are only set suitable values in (1). In other words, even-numbered coefficients are set as all zero. Then, we can easily confirm that it generates chaos in this function.

In this study, we consider a simple network using the chaotic map as a subsystem. We use the cubic map as the chaotic subcomponent in the following.
Figure 1: Diagram of a cubic map for $a = 2.6$

$$x(t + 1) = a_3 x(t)^3 + a_1 x(t)$$  (2)

where $a_k$ is a parameter which can determine the strength of nonlinear characteristic. We can easily confirm that chaos generates in this subsystem.

In order to simplify, consider the parameters $-a_3$ and $a_1$ is the same value $a$, then we hereafter use the cubic map in the following.

$$x(t + 1) = a x(t) \left( 1 - x(t)^2 \right)$$  (3)

It is well known that we can confirm a crisis of chaos, if the parameter $a$ is greater than around 2.6, which is obtained from Fig. 1. Hence the chaotic map can move both positive and negative area when the parameter $a$ is larger than a rigorous value $\frac{3 \sqrt{3}}{2} \approx 2.6$. Further, if $a > 3$, i.e. $\frac{2a}{3 \sqrt{3}} > \sqrt{\frac{1+a}{a}}$, we can also confirm that the system diverges to infinity.

In order to evaluate the function (1), Lyapunov exponent can be calculated by the following formula.

$$\lambda = \lim_{N \to \infty} \frac{1}{N} \sum_{k=1}^{N} \log \left| \frac{df(x)}{dx} \right|$$  (4)

Lyapunov exponent is a very important measurement often used to show the existence of chaos. Lyapunov exponents with bifurcation diagram by changing one parameter $a$ for Eq. (3) are shown in Fig. 2. These are typical results which can be obtained from computer calculation. In case of polynomial functions, period doubling and tangent bifurcation can be confirmed. Therefore chaotic maps possessing several equilibrium points can yield various wide interesting behavior.

3. Several Phase Patterns in Coupled Chaotic Maps

In this section, we consider two types of coupled chaotic network system as shown in Fig. 3, which each cell means a chaotic map as a subsystem of the network. It can be considered easily that coupled chaotic systems have wide variety of phase patterns or spatio-temporal features. The term “spatio-temporal” is extensively used for irregular dynamical behavior observed from large scale complex systems of the relevant to both time and space.

In order to confirm spatio-temporal phenomena or phase patterns, consider a coupled model of the chaotic maps which are connected to neighbors. Each chaotic cell is connected to neighbors by arbitrary coupling strength $\varepsilon$. The whole system of CML is represented as

$$x_k(t + 1) = (1-\varepsilon) f(x_k(t))$$

$$+ \frac{\varepsilon}{2} \left( f(x_{k-1}(t)) + f(x_{k+1}(t)) \right), \quad (k = 1, 2, \ldots, N)$$  (5)

where $t$ is an iteration, $k$ is an index number of the cell which follows the cyclic rule, and $N$ is a size of coupled cells, respectively.

On the other hand, the 2-dimensional network system is represented as

$$x_{ij}(t + 1) = (1-\varepsilon) f(x_{ij}(t))$$

$$+ \frac{\varepsilon}{4} \sum_{k \in \Xi} f(x_{ij}(t)) \quad (i, j)$$  (6)

where $[i, j]$ is an index number of cell, and $\Xi$ means four neighbor cells.

In the previous studies related with CML network, the coupling strength is used to be a constant and an unique value. In this study, we propose the coupled chaotic system model which the coupling strength will be used in the following Gaussian function.
Similarly, we consider a 2-dimensional network system which each cell is coupled to four neighbors. Figure 6 shows some simulation results obtained from 2-dimensional system (6) at time $t = 1000$. This figure indicates a grade of synchronous state for phase difference with an average of four neighbors, which is illustrated with gray scale monotone colors between white and black correspond to synchronous and asynchronous state, respectively. We can confirm that self-organizing formation advances gradually as the parameter grows. Although we cannot present all the simulation results, several phase patterns and spatio-temporal phenomena from its complex dynamics will be observed in such coupled systems.

4. Conclusions

In this study, we considered the coupled network using a cubic map as a chaotic cell, and investigated their dynamics. Some computer simulation results of spatio-temporal chaotic behavior several phase patterns in the coupled chaotic maps for ring and 2-dimensional network systems had been shown. Varying coupling strength provided by a Gaussian function is very important to solve several features in the real world. We consider that the coupled chaotic system is also a good model as a stochastic model such as natural patterns, self-organization, and so on. Furthermore, we would like to use these pattern dynamics as a stochastic model for several moving robots in near future.

References

Figure 5: Some simulation results of time-waveform with non-uniform coupling strength $\varepsilon(v)$ for $\sigma = 0.4$. Horizontal axis is a time $t$, and vertical axis corresponds to waveform of each cell $x_k$. (a) $a = 2.60$, $\gamma = 0.6$, (b) $a = 2.58$, $\gamma = 1.0$, (c) $a = 2.60$, $\gamma = 0.6$, and (d) $a = 2.88$, $\gamma = 0.6$

Figure 6: Snapshots of some simulation results of 2-dimensional $50 \times 50$ network at $t = 1000$. (a) $a = 2.60$, $\gamma = 0.60$, $\sigma = 0.20$, (b) $a = 2.59$, $\gamma = 1.0$, $\sigma = 0.40$, (c) $a = 2.69$, $\gamma = 0.80$, $\sigma = 0.40$, and (d) $a = 2.88$, $\gamma = 0.60$, $\sigma = 0.80$
Ant Colony Optimization with Intelligent and Dull Ants

Sho Shimomura†, Masaki Sugimoto†, Taku Haraguchi†, Haruna Matsushita‡ and Yoshifumi Nishio†

†Department of Electrical and Electronic Engineering, Tokushima University
2-1 Minami-Josanjima, Tokushima 770-8506, Japan
‡Department of Electrical and Electronics Engineering, Hosei University
3-7-2 Kajino-cho, Koganei-city, Tokyo 184-8584, Japan
Email: †{s-sho, sugimoto, nishio}@ee.tokushima-u.ac.jp, ‡haruna.matsushita.pd@k.hosei.ac.jp

Abstract—This study proposes a new Ant Colony Optimization (method; ACO) with Intelligent and Dull Ants (IDACO). In IDACO algorithm, two kinds of ants coexist: intelligent ants and dull ants. IDACO algorithm is nearer to the real ant colony than the standard ACO algorithm. We apply IDACO to Traveling Salesman Problems (TSPs) and confirm that IDACO obtains more effective results than the standard ACO which consists of only the intelligent ants.

1. Introduction

Ant Colony Optimization (ACO) [1] is a biologically inspired optimization algorithm with pheromone effect of ants and is effective to solve difficult combinatorial optimization problems, such as Traveling Salesman Problem (TSP) [2], graph coloring problem [3], Quadratic Assignment Problem (QAP) [4] and so on. TSP is a problem in combinatorial optimization studied in an operations research and a theoretical computer science. In TSP, given a list of cities and their pairwise distances, the task is to find the shortest possible tour that each city exactly visited once. In ACO algorithm, multiple solutions called “ants” coexist, and the ants drop pheromone on the path connecting the cities. Pheromone trails are updated depending on the behavior of the ants. The ants find a food source through paths having strong pheromone. By communicating with other ants according to the pheromone strength, the algorithm tries to find the optimal solution. However, ACO has a problem which is to fall into local solutions. Therefore, it is important to enhance the algorithm performances by improving its flexibility.

Meanwhile, it has been reported that about 20 percent of the ants are unnecessary ants called “dull ant” in the real ant’s world [5]. The dull ant keeps still around its colony whereas the other ants in the colony perform feeding behavior. In a computational experiment, the researchers performed the feeding behavior by using intelligent ants, which can trail the pheromone exactly, and dull ants which cannot trail the pheromone. From results, the ants group including the dull ants can obtain more foods than the group containing only the intelligent ants. It means that the coexistence of the intelligent and dull ant improves the effectiveness of the feeding behavior.

In this study, we propose a new type of ACO algorithm called Ant Colony Optimization with Intelligent and Dull Ants (IDACO). The important feature of IDACO is that two kinds of ants coexist. The one is an intelligent ant and the other is a dull ant. The intelligent ant can trail the pheromone and the dull ants cannot trail the pheromone. Because their features are essentially similar to the real ant’s world, we can say that IDACO algorithm is nearer to the real ant colony than the standard ACO algorithm.

2. Ant Colony Optimization with Intelligent and Dull Ants (IDACO)

We explain the proposed IDACO algorithm in detail. In the IDACO algorithm, the most important feature of IDACO is that two kinds of ants coexist; the intelligent ants and the dull ants. The intelligent ant is the same as the ant of the standard ACO, and it can exactly trail the pheromone. In constant, the dull ant cannot trail the pheromone.

\( N \)-city of TSP is denoted as

\[
S \equiv \{P_1, P_2, \cdots, P_N\}, \quad P_i \equiv (x_i, y_i),
\]

where a data area is normalized from 0 to 1, and \( P_i \) is an \( i \)-th city position \( (i = 1, 2, \cdots, N) \). Each ant (total \( M \)) is deposited on a city selected at random. \((1-d) \times M\) ants and \( d \times M \) ants are classified into a set of the intelligent ants \( S_{\text{int}} \) and of the dull ants \( S_{\text{dull}} \), respectively. \( d \) is a rate of dull ants in all the ants.

[Step1](Initialization): Let an iteration number \( t = 0 \). \( \tau_{ij}(t) \) is an amount of pheromone deposited on a path \((i, j)\) between the city \( i \) and the city \( j \) at time \( t \), and \( \tau_{ij}(t) \) is initially set to \( \tau_0 \).

[Step2](Find tour): For the intelligent ants and the dull ants, the visiting city is chosen by the probability \( p_{i,j}(t) \) and \( p_{i,j,d}(t) \), respectively, as shown in Fig. 4. The probability of \( k \)-th ant moving from the city \( i \) to the city \( j \) is decided by

\[
p_{k,i,j}(t) = \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in S_{\text{int}}} [\tau_{il}(t)]^\alpha [\eta_{il}]^\beta}, \quad \text{if } k \in S_{\text{dull}}
\]

\[
p_{k,i,j}(t) = \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in S_{\text{int}}} [\tau_{il}(t)]^\alpha [\eta_{il}]^\beta}, \quad \text{otherwise}
\]

- 504 -
where \( k = 1, 2, \cdots, M \), and \( 1/\tau_{ij} \) is a distance of the path \((i, j)\). The adjustable parameters \( \beta_I \) and \( \beta_D \) control the weight of the city information of the intelligent ant and of the dull ant, respectively. Therefore, the searching ability goes up and down by changing \( \alpha \) and \( \beta \). As Eq. (2) does not include the amount of deposited pheromone \( \tau_{ij}(t) \) and \( \tau_D(t) \), the dull ants cannot trail the pheromone. In contrast to the intelligent ants which judge next city by the pheromone and the distance from a present location, the dull ants judge next city depending on only the distance from a present location. \( N_k \) is a set of cities that \( k \)-th ant has never visited. The ants repeat choosing next city until all the cities are visited.

[Step3](Pheromone update): After all ants have completed their tours, the amount of deposited pheromone on each path is updated. We should note that the dull ants can deposit the pheromone on the path, though, they cannot trail the pheromone. Then, the tour length \( L_k(t) \) is computed for both the intelligent and dull ants, and the amount of pheromone \( \Delta \tau_{k_i}(t) \) deposited by \( k \)-th ant on the path \((i, j)\) is decided as

\[
\Delta \tau_{k_i}(t) = \begin{cases} 
1/L_k, & \text{if } (i, j) \in T_k(t) \\
0, & \text{otherwise}, 
\end{cases}
\]

where \( T_k(t) \) is a tour obtained by \( k \)-th ant, and \( L_k(t) \) is its length. Update \( \tau_{ij}(t) \) of each path \((i, j)\) depending on its \( \Delta \tau_{k_i}(t) \);

\[
\tau_{ij}(t + 1) = (1 - \rho)\tau_{ij}(t) + \sum_{k=1}^{M} \Delta \tau_{k_i}(t),
\]

where \( \rho \in [0, 1] \) is the rate of pheromone evaporation.

[Step4] Let \( t = t + 1 \). Go back to [Step2] and repeat until \( t = t_{\text{max}} \).

3. Numerical Experiments

In order to evaluate a performance of IDACO and to investigate its behavior, we apply IDACO to two TSPs and compare IDACO with different rates of dull ants with the standard ACO.

In the experiments, the number of ants \( M \) in the standard ACO and IDACO are set to the same as the number of cities. The standard ACO contains only the intelligent ants whose choice probability is decided by Eq. (3). IDACO includes \( d \times M \) dull ants, and \( (1 - d) \times M \) intelligent ants in each simulation. The rate of dull ants \( d \) is set to 0.2 and 0.5 in each simulation. We repeat the simulation 10 times for all the problems. The parameters of the standard ACO and IDACO were set as follows:

\[
\tau_0 = 10, \; \rho = 0.3, \; \alpha = 1, \; \beta = \beta_I = \beta_D = 5, \; t_{\text{max}} = 2000,
\]

where the evaporation rate \( \rho \), the weight of pheromone \( \alpha \), the weight of distance \( \beta \), \( \beta_I \) and \( \beta_D \) and the search limit \( t = t_{\text{max}} \) are fixed values.

In order to compare obtained solutions with the optimal solution, we use an error rate as follow:

\[
\text{Error rate}[^{\%}] = \frac{\text{(obtained solution) - (optimal solution)}}{\text{(optimal solution)}} \times 100. \tag{6}
\]

This equation shows how close to the optimal solution the ACOs obtain the tour length. Thus, the error rate nearer 0 is more desirable. Furthermore, in order to evaluate how well the solution of IDACO are improved from that of ACO, we use an improved rate as follow:

\[
\text{Improved rate}[^{\%}] = \frac{\text{(Avg. of Error of ACO)} - \text{(Avg. of Error of IDACO)}}{\text{(Avg. of Error of ACO)}} \times 100. \tag{7}
\]

3.1. Simulation 1: att48

![Figure 2: Benchmark problem att48 and its optimal tour.](image)
Figure 3: The best tours of the three ACOs for att48. (a) The standard ACO. (b) IDACO with \(d = 0.2\). (c) IDACO with \(d = 0.5\).

In Table 1, IDACO obtained better results than the standard ACO. Furthermore, as the result of att48, IDACO with \(d = 0.2\) obtained better results than IDACO with \(d = 0.5\). From the results of the Simulation 1 and 2, we can confirm that IDACO obtains the effective results when the rate of dull ants is same as that in the real ant world. Especially in case of TSP containing many cities, IDACO greatly improves the result of the standard ACO.

Table 1: Results of the standard ACO and IDACO for att48.

<table>
<thead>
<tr>
<th>Error rate</th>
<th>ACO (d = 0)</th>
<th>IDACO (d = 0)</th>
<th>IDACO (d = 0.2)</th>
<th>IDACO (d = 0.5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>2.87%</td>
<td>2.62%</td>
<td>3.84%</td>
<td></td>
</tr>
<tr>
<td>Minimum</td>
<td>2.43%</td>
<td>1.46%</td>
<td>2.09%</td>
<td></td>
</tr>
<tr>
<td>Improved rate from ACO</td>
<td>-</td>
<td>8.7%</td>
<td>-33.8%</td>
<td></td>
</tr>
</tbody>
</table>

3.2. Simulation 2: kroA100

Figure 4: Benchmark problem kroA100 and its optimal tour.

In Table 2, IDACO obtained better results than the standard ACO. Furthermore, as the result of kroA100, the IDACO with \(d = 0.2\) obtained better results than IDACO with \(d = 0.5\). From the results of the Simulation 1 and 2, we can confirm that IDACO obtains the effective results when the rate of dull ants is same as that in the real ant world. Especially in case of TSP containing many cities, IDACO greatly improves the result of the standard ACO.

Table 2: Results of the standard ACO and IDACO for kroA100.

<table>
<thead>
<tr>
<th>Error rate</th>
<th>ACO (d = 0)</th>
<th>IDACO (d = 0)</th>
<th>IDACO (d = 0.2)</th>
<th>IDACO (d = 0.5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>1.27%</td>
<td>1.12%</td>
<td>1.13%</td>
<td></td>
</tr>
<tr>
<td>Minimum</td>
<td>1.22%</td>
<td>1.12%</td>
<td>1.12%</td>
<td></td>
</tr>
<tr>
<td>Improved rate from ACO</td>
<td>-</td>
<td>11.81%</td>
<td>11.02%</td>
<td></td>
</tr>
</tbody>
</table>

4. Conclusions

In this study, we have proposed Ant Colony Optimization with Intelligent and Dull Ants (IDACO). We have investigated the performances by applying it to two TSPs. We have confirmed that IDACO including the dull ants obtained better results than ACO which containing only the intelligent ants because the dull ants help in getting out of the local optima. Furthermore, we have confirmed that IDACO with \(d = 0.2\) obtained better results than the standard ACO for all the cases. From these results, we can say that IDACO is nearer to the real ant colony than the standard ACO and is effective algorithm in solving TSP.

References


Figure 5: The best tours of the three ACOs for kroA100. (a) The standard ACO. (b) IDACO with $d = 0.2$. (c) IDACO with $d = 0.5$.


Comparison of the linear algebra approach and the evolutionary computing for magnetic field shaping in linear coils.

Bartłomiej Garda† and Zbigniew Galias†

†Department of Electrical Engineering
AGH University of Science and Technology
al. Mickiewicza 30, 30-059, Kraków, Poland
email: bgarda@agh.edu.pl, galias@agh.edu.pl

Abstract—In this work we consider the problem of shaping magnetic field excited by magnetic coils. An important special case is constructing coils that excite a homogeneous magnetic field in some specific regions. This problem is an important step in the design of superconducting magnetic coils for Magnetic Resonance Imaging devices, where in certain regions a strong static magnetic field with high homogeneity is needed. In this work, the authors investigate the problem of a linear coil design using two approaches. The first approach is based on linear algebra. The problem under study can be formulated as an over-determined set of linear equations. Since usually the matrix A describing the problem is ill-conditioned, solutions obtained using the standard method of least squares are very large in magnitude and hence are useless from the applications point of view. Therefore, a regularization method is employed to make the linear problem well-posed. The second method considered uses evolutionary computations. These two techniques are compared in terms of effectiveness and computational complexity. Performance of both methods is evaluated using several examples.

1. Introduction

Many electromagnetic devices are designed with respect to the required magnetic or electric field distribution in a given region.

![Figure 1. The magnet design problem. Cylindrical coordinates are applied](image)

The problem studied in this paper can be solved by using the idea of the “target field” approach applied to the problem of static magnetic field excited by the set of n coaxial coils [7]. Figure 1 presents that situation. The feasible coil space is filled by an array of candidate coils. The coils are assumed to be ideal current loops located at the center of the squares, see Fig. 1. The goal is to find the values of currents flowing through coils that excite the desired magnetic field at each target point. The shape of the target area is given (in Fig. 1 the target area is the circle located at the center of the coordinate system).

![Figure 2. Single coil loop cross-section caring current I, positioned at point (r,z) that excites magnetic field B at point (r,z).](image)

For the system of coaxial circular coils, only the B_r and B_z components of the magnetic field need to be considered. In [1] it was shown that B_r component is much smaller than B_z and it has negligible contribution on B. This effect is often called quadrature suppression. Therefore, it is sufficient to consider the B_z component only.

Each coil generates a magnetic field contribution at each target point. A contribution from a single coil can be calculated using the relation (compare Fig. 2):

\[
B_{ij} = \frac{\mu_0 I_i}{2\pi r_{ij}^2} \left( K(\kappa_{ij}) + 2 \frac{r_{ij}^2 - r_i^2 - (z_j - z_i)^2}{r_{ij}^2 + (z_j - z_i)^2} E(\kappa_{ij}) \right)
\]

(1)

where \(B_{ij}\) is the z component of the magnetic field generated at the j-th target point located at \((r_j, z_j)\) due to the current \(I_i\) of the i-th coil, and \(E(\kappa_{ij})\) and \(K(\kappa_{ij})\) are complete elliptic integrals of the first and second kind, respectively, and the factor \(\kappa_{ij}\) is given by:

\[
\kappa_{ij} = \frac{4 r_i r_j}{(r_i^2 + z_i^2) + (r_j^2 + z_j^2)}
\]

(2)

If the target point is positioned on the z axis, i.e. \(r_j=0\), the formula (1) can be simplified as

\[
B_{ij} = \frac{\mu_0 I_i}{2\pi} \frac{r_i^2}{(r_i^2 + z_i^2)^{3/2}}
\]

(3)

In the work, we investigate two approaches to solve the problem presented above. The first approach is based on
linear algebra, while the second one uses evolutionary computations.

2. The least-squares solution

Let us assume that there are N simple coils with currents \( i_1, i_2, \ldots, i_N \), and M target points with the desired value of the magnetic field \( b_1, b_2, \ldots, b_M \). Since the relation between the field at target points and the current at a given coil is linear, one can formulate the problem as an overdetermined set of linear equations.

\[
A i = b,
\]

where \( A \in \mathbb{R}^{M \times N} \) is a coefficient matrix, \( i \in \mathbb{R}^N \) is a vector of the currents to be found, and \( b \in \mathbb{R}^M \) is a vector describing the required field at the target points. Components of the matrix \( A \) can be computed by dividing the result obtained from the relation (1) or (3) by the current \( I_i \).

The least-squares solution is the one that minimizes the sum of squares of residual errors for all target points. This can be expressed by:

\[
\min_i \{ \| A i - b \|_2^2 \}.
\]

It is well known that the minimum (5) can be found by solving the set of normal equations:

\[
A^T A i = A^T b.
\]

Unfortunately, the matrix \( A \) is usually ill-conditioned (especially for large N and M) and the solution obtained in this way is useless.

In simulations, we consider two cases. In the first case target points are distributed on the circle of radius \( r_{\text{eq}} \) positioned at the center of coordinate system. In the second case target points are located on the z axis. The parameters are presented in the Table 1.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( r_{\text{eq}} ) - coil radius</td>
<td>2.5 cm</td>
</tr>
<tr>
<td>( r_{\text{eq}} ) - radius of target points circle</td>
<td>1.5 cm</td>
</tr>
<tr>
<td>( \varphi ) - coil length</td>
<td>20 cm</td>
</tr>
<tr>
<td>( \varphi ) - length of required field on the axis</td>
<td>25 cm</td>
</tr>
</tbody>
</table>

Table 1. Examined coils’ dimensions

In order to make the solution process numerically stable some sort of regularization scheme is needed.

3. Tikhonov regularization

In this work, we use the Tikhonov regularization [6]. To choose the desired solution, additional information (the so-called \( a \text{ priori} \) information) should be imposed on (5). Since we require that the solution is smooth, the criterion (5) can be reformulated as

\[
\min_i \{ \| A i - b \|_2^2 + \lambda^2 \| i \|_2^2 \}.
\]

The parameter \( \lambda \) controls the smoothness of the solution.

Let’s define the regularized matrix \( A_\lambda \) as

\[
A_\lambda = (A^T A + \lambda^2 I_N)^{-1} A^T,
\]

where \( I_N \in \mathbb{R}^{N \times N} \) is an identity matrix, \( \lambda \) is a regularization parameter. The regularized solution is:

\[
i = A_\lambda^T b.
\]

The goal is to achieve the uniform (homogenous) magnetic field equal to 1T. Figures 3 and 4 present solutions obtained for the case when the target points are located along the z-axis. Observe that the currents found have very large values (of order \( 10^{10} \) A) and in most cases currents in neighboring coils have opposite signs. The solutions found are useless.

This is caused by the fact that the inverse problem is ill-conditioned, i.e. very small changes in the vector \( b \) can produce huge errors in the solution \( i \).

In order to make the solution process numerically stable some sort of regularization scheme is needed.

\[
\text{Figure 3. Solution for small N and M (N=M=21).}
\]

\[
\text{Figure 4. Solution for N=M=99x199}
\]

The problem is to find the value of the parameter \( \lambda \) that provides a good compromise between the desired field shape and the smoothness of the solution. There are many techniques which estimate the optimal value of \( \lambda \). Here, we use the L-curve method [2].

The L-curve is a parameterized (\( \lambda \) is a parameter) plot of the norm of the solution \( \| i \|_2 \), versus the residual norm \( \| A i - b \|_2 \). An important feature of the L-curve is that its L-shaped corner appears for regularization parameters
close to the optimal value. The idea of the L-curve criterion for finding the regularization parameter \( \lambda \) is to choose a point on this curve as close as possible to the corner.

Fig. 5 presents the solution of equation (10) for the optimal value of the regularization parameter \( \lambda \). The corresponding L-curve is presented in Fig. 6. It can be seen that the Tikhonov regularization smoothes the distribution of currents. Observe that all currents flow in the same direction, and that the values are much smaller than for the least squares solution. All calculations were made using the Matlab environment and the regularization tools [3].

![Figure 6](image6.png)

**Figure 6.** The L-curve graph for the case presented on figure 5.

The results for the case when the target points are located on the circle are presented in Fig. 7. One can see that the solution is smooth and the values of currents are acceptable.

![Figure 7](image7.png)

**Figure 7.** Distribution of currents and the corresponding field distribution for the target points located on the half circle at the center of coordinate system. The angle \( \alpha \) is defined in Fig. 1.

4. Genetic calculations

For comparison, we made computation using the genetic algorithms. Probabilistic methods are frequently used as a tool for solving many scientific or engineering problems. Among them genetic algorithm (GA) approach is particularly well studied and is known to be very effective in solving certain problems [4].

For the case of target points located along the z-axis and a homogeneous field the problem studied can be formulated as minimizing the homogeneity factor

\[
\varepsilon = \left( \frac{1}{p} \sum_{j=1}^{p} \left( B_z - B_{eq} \right)^2 ight)^{1/2} / B_{eq},
\]

where \( B_z \) is the field value in the z axis and \( B_{eq} \) is the target field (in our case \( B_{eq}=1 \)T).

At the starting point the algorithm generates the initial population. In our case it is a set of coils with randomly assigned current values. We assume that each current must not exceed a certain threshold \( I_{max} \). Fig. 9 presents currents and the corresponding field distribution for one of members from the initial population. In each step of the genetic algorithm individuals are chosen for the genetic operations. Selection is done using the roulette method i.e. the probability of selecting an individual to further genetic operators is proportional to its adaptation (lower \( \varepsilon \) value).

In order to avoid premature convergence the mutation probability was decreased with the generation number \( k \) according to formula:

\[
p_m(k) = p_{m_{min}} + (p_{m_{max}} - p_{m_{min}}) \exp\left(-\left(\frac{k}{p_{max} - p_{min}}\right)^2\right) (10)
\]
At the beginning of GA the mutation probability starts from the value of $P_{m_{\text{max}}}$ and when the generation was growing the probability decreasing down to the fixed value of $P_{m_{\text{min}}}$.

The current distribution and its corresponding magnetic field distribution.

The parameters of the genetic algorithm are summarized in Table 2. The set of currents is binary coded using Grey code with 8 bits, which means that currents are selected from the set of current values with 256 elements. The coil is divided into 25 simple coil loops and there are 75 target points located on the z axis.

![Figure 9](image1.png) Exemplary individual from the beginning population. The current distribution and its corresponding magnetic field distribution.

![Figure 10](image2.png) The best individual from 2000th population and corresponding field distribution.

<table>
<thead>
<tr>
<th>Population size</th>
<th>75</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of generation (stop criterion) - MaxGen</td>
<td>2000</td>
</tr>
<tr>
<td>Crossover probability</td>
<td>0.8</td>
</tr>
<tr>
<td>Beginning mutation probability $P_{m_{\text{max}}}$</td>
<td>0.8</td>
</tr>
<tr>
<td>Fixed mutation probability $P_{m_{\text{min}}}$</td>
<td>0.08</td>
</tr>
</tbody>
</table>

Table 2. Genetic algorithm parameters

Fig. 10 presents the best individual (the one with the lowest ε value) from the 2000th population. One can see that the solution presented in Fig. 10 is not optimum in terms of the homogeneity factor ε. One can also see the lack of symmetry in the solution. There is a number of modifications for the GA technique which may improve its performance. This includes increasing the number of generations, the size of a population, using a different formula for mutation probability, variable population size, applying different coding system, or algorithm with subpopulations etc.

![Figure 11](image3.png) The mutation probability function described by the equation (10) for the parameters from Table 2

**Conclusion**

A linear algebra (deterministic) and evolutionary (probabilistic) methods for solving the coil design problem have been compared. Both techniques can provide solutions that might be acceptable from the applications point of view. It is interesting to note that for such problems the direct application of the least squares method does not produce a useful result. This is because such problems are ill-posed according to Hadamard [6]. Regularization methods may provide better solutions. With the Tikhonov method one modifies the cost function in such a way that impractical solutions have higher cost. A proper choice of the parameter controlling the smoothness of the solution is crucial.

On the other hand in genetic algorithms one can impose certain constraints on allowable solutions. In the examples considered these constraints may include the current values (also directions), symmetry of the solution, etc. Another advantage is that genetic algorithms are easy to implement. The main advantage of genetic algorithm lies in the fact that they can be directly used for solving nonlinear problems. It is worth to point out that for the problem considered genetic algorithm were much more time consuming.

**References**


Office Layout Support System using Genetic Algorithm
– Generation of Room Arrangement Plans for Polygonal Space –

Ryota TACHIKAWA† and Yuko OSANA†

†School of Computer Science, Tokyo University of Technology
1404-1 Katakuragi, Hachioji, Tokyo, 192-0982, Japan Email: osana@cs.teu.ac.jp

Abstract—In this paper, we propose an office layout support system using genetic algorithm. The proposed system has two phases; (1) generation of room arrangement plans and (2) generation of layout plans for workspace. In the proposed system, some conditions on rooms and furniture are given by a user, some room arrangement plans which satisfy the conditions are generated by genetic algorithm. After one of the generated room arrangement plans is selected by a user, and then some layout plans for workspace which satisfy the conditions are generated by genetic algorithm. In the proposed system, the combined genetic algorithm which is based on the adaptive genetic algorithm and the genetic algorithm with search area adaptation is employed in order to improve the convergence speed and the diversity of layout plans. We carried out a series of computer experiments and confirmed the effectiveness of the proposed system.

1. Introduction

When we consider how fixture and furniture such as desks and shelves are arranged to the limited space such as an office and a laboratory, we arrange various kinds of furniture virtually on a paper. Moreover, the software for an office layout is also put on the market, and we can also think arranging furniture virtually on a screen using it. However, it is difficult to consider the layout plans which satisfy various conditions such as a size of room, the number of the furniture to be arranged and so on.

As the system which can generate layout plans which satisfy the conditions given by users automatically, the interior design layout support system[1] has been proposed. However, in the system, each desk is arranged individually, so the desks are sometimes arranged in disorder. As a result, it is difficult to generate a practical layout plan. In ref.[2], the interior design layout support system using evaluation agents has been proposed, however, the system sometimes generate layout plans which do not satisfy all conditions given by users.

Recently, we have proposed some office layout support systems using genetic algorithm[3][4]. In these systems, some conditions such as size and form of room, size and the number of desks are given by users, some layout plans which satisfy the conditions are generated by genetic algorithm. We have proposed an office layout support system considering floor using genetic algorithm[5][6]. These systems have two phases; (1) generation of room arrangement plans and (2) generation of layout plans of workspace. However, these systems generate layout plans for only rectangular space. Although genetic algorithm has the impressive ability in the combination optimization problem, it has some problems in the local searching ability and appropriate parameter setting for crossover and mutation.

In this paper, we propose an office layout support system using genetic algorithm which can generate layout plans for polygonal space. In the proposed system, the combined genetic algorithm which is based on the adaptive genetic algorithm (AGA)[7] and the genetic algorithm with search area adaptation (GSA)[8] is employed in order to improve the convergence speed and the diversity of layout plans.

2. Office Layout Support System using Genetic Algorithm

The proposed office layout support system using genetic algorithm has two phases; (1) generation of room arrangement plans and (2) generation of layout plans of workspace. In the proposed system, some conditions on rooms and furniture are given by a user, some room arrangement plans which satisfy the conditions are generated by genetic algorithm. After one of the generated room arrangement plans is selected by a user, and then some layout plans for workspace which satisfy the conditions are generated by genetic algorithm. Figure 1 shows the outline of the proposed system.

![Figure 1: Outline of Proposed System.](image-url)
2.1. Generation of Room Arrangement Plans

2.1.1. Expression in Gene of Room Arrangement Plans.

In this research, the room arrangement plan is expressed in the form of a gene. The length of the gene corresponding to the room arrangement plan in the polygonal space with C angles which has R rooms is $2R + 3C - 1$. The gene consists of 4 parts: (1) rule for room size decision, (2) relation between rooms, (3) arrangement method in each wall and (4) angle of rooms in each wall. Figure 2 shows an example of the gene and the corresponding room arrangement plan.

2.1.2. Evaluations.

In genetic algorithm, in order to leave better genes to the next generation, the degree of fitness is calculated, and genes are chosen according to the degree of their fitness. In this system, the fitness is represented by 4 elements: (1) floor space for workspace ($FitW$), (2) dead space along walls ($FitD$), (3) space that is not along walls and (4) distance between reception room and entrance ($FitR$).

$FitW$ is determined by the percentage of the maximum space for the workspace (yellow area in Fig.3). $FitD$ is determined by the percentage of the dead space along walls (blue area in Fig.3). $FitI$ is determined by the percentage of the dead space not along walls (red area in Fig.3).

$FitR$ is calculated by

$$FitR = \frac{1}{R_c} \sum_{i=1}^{R_c} \frac{D_{R(i)}}{F_w + F_d}$$  \hspace{1cm} (1)

where $D_{R(i)}$ is the Manhattan distance between the center of the reception room $i$ and the entrance, $R_c$ is the number of reception rooms, $F_w$ is the floor width and $F_d$ is the floor depth.

2.1.3. Combined Genetic Algorithm

In the proposed system, the combined genetic algorithm which is based on the adaptive genetic algorithm[7] and the genetic algorithm with search area adaptation[8] is employed in order to improve the convergence speed and the diversity of layout plans. In the adaptive genetic algorithm, two kinds of crossover methods are used.

(1) Crossover for Non-Elite Genes

If the elite degree of the one parent gene (or both parent genes) is smaller than the threshold, two-point crossover is used. Here, the elite degree of the gene $p$ in the generation $T$ is given by

$$E_{deg}(T, p) = \sum_{j=0}^{l_{max}} \left[ Elite(p, j) \right] \cdot \beta^j$$

$$E_{deg}(T, p) = \sum_{j=0}^{l_{max}} \left[ Elite(p, j) \right] \cdot \beta^j$$  \hspace{1cm} (2)

where $l_{max}$ is the considered level depth, $|Elite(p, j)|$ is the number of the elite ancestors of the gene $p$ in the generation $T - j$, $|Anc(p, j)|$ is the number of ancestors of the gene $p$, and $\beta (0 < \beta \leq 1)$ is the damping factors. Figure 4 shows the example of the calculation of the elite degree.

(2) Crossover for Elite Genes

If the elite degree of the both parent genes is larger than the threshold, the proposed crossover method (Common Sequence Preserving Crossover : CSPX) is used. In the proposed CSPX, the common relation between rooms can be preserved.

(1) In the part for relation between rooms of genes, $S_{RR}$ (common relation between rooms) and $S_{CR}$ (common relation between wall and rooms) are obtained.

(2) The part corresponding to $S_{RR}$ are copied to the child genes.
2.2.2. Evaluation.

When all shelves are arranged along walls, the fitness is represented by (1) position of printers (FitP) and (2) relation between group and own shelves (Fit5). When shelves are used as partitions, the fitness is represented by (1) position of printers (FitP) and (2) access between groups (FitG).

3. Computer Experiments

In this section, we show the computer experiment result to demonstrate the effectiveness of the proposed system.

3.1. Generation of Room Arrangement Plans

In this experiment, room arrangement plans in the hexagonal space are generated by the proposed system. Figure 6 shows an example of generated room arrangement plans.

3.2. Comparison with Other Algorithms

3.2.1. Transition of Fitness

Figure 7 shows the transition of maximum and average fitness. In this figure, the transition of fitness in the system which uses normal genetic algorithm/AGA/GSA for reference. As shown in this figure, the fitness in the proposed system (which uses combined genetic algorithm) is larger than that in the systems based on the normal genetic algorithm and the AGA.

3.2.2. Variation of Generated Plans & Execution Time

Figure 8 shows the variation of generated plans. As shown in this figure, the proposed system can generate various plans as similar as the systems based on the normal genetic algorithm and the AGA. In contrast, only few plans were generated in the system based on the GSA although the fitness in the same trials was high (See Fig.7).

Table 1 shows the average number of the generated plans and the execution time in each system.

![Figure 6: Generated Room Arrangement Plans.](image)
3.3. Generation of Layout Plans for Workspace

In this experiment, the layout plans for workspace were generated in the proposed system. Figures 9 and 10 show examples of the generated layout plans.

4. Conclusions

In this paper, we have proposed the office layout support system using genetic algorithm. In the proposed system, the combined genetic algorithm which is based on the adaptive genetic algorithm and the genetic algorithm with search area adaption is employed in order to improve the convergence speed and the diversity of layout plans. We carried out a series of computer experiments and confirmed the effectiveness of the proposed system.

References


Table 1: Average Number of Generated Plans and Execution Time.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Execution Time (min : sec)</th>
<th>Average Number of Generated Plans</th>
</tr>
</thead>
<tbody>
<tr>
<td>Combined</td>
<td>7:29.00</td>
<td>89.4</td>
</tr>
<tr>
<td>Normal</td>
<td>4:21.00</td>
<td>108.5</td>
</tr>
<tr>
<td>AGA</td>
<td>4:17.20</td>
<td>106.8</td>
</tr>
<tr>
<td>GSA</td>
<td>19:29.00</td>
<td>9.8</td>
</tr>
</tbody>
</table>

Figure 7: Transition of Fitness (Comparison with Other Algorithms).

Figure 8: Center Position of Room 1.

Figure 9: Generated Layout Plans for Workspace.

Figure 10: Generated Layout Plans (3D).
Design of Class E Amplifier Using Particle Swarm Optimization

Yuichi Tanji† and Hiroo Sekiya‡

†Dept. of RISE, Kagawa University
‡Graduate School of Advanced Integration Science, Chiba University, Japan
Email: †tanji@eng.kagawa-u.ac.jp, ‡sekiya@faculty.chiba-u.jp

Abstract—Class E amplifier is known as a switching type power amplifier which has a high efficiency and output power. However, the design is very difficult, since the constraints on the steady-state must be satisfied. In this paper, an optimization procedure of class E amplifier is proposed, where the particle swarm optimization is used. In the proposed approach, the design parameters or the optimal solution is evaluated by a standard commercial circuit simulator. Therefore, we can determine the parameters considering the nonlinear characteristics of MOSFET included in the circuit fully. We define a new cost function for determining the design parameters of class E amplifier. By the simulation, it is confirmed that the adjusted circuit behaves as a class E amplifier certainly.

1. Introduction

There are many applications of drivers with sinusoidal waveform for power systems. A class E amplifier is the best choice as a circuit configuration of the drivers, since it is capable of combining a high efficiency (> 50%) with a resonant output power (30 dBm) [1]. On the other hand, the design of class E amplifier is very difficult. To make the circuit behave as a class E amplifier which has switching constraints on the steady-state, the designers have to adjust the passive elements in the circuit and the device parameters of MOSFET in order to minimize the switching losses.

To overcome its difficulty, an optimization procedure based on the shooting Newton method is proposed in [2]. In this method, the class E amplifier is idealized by two linear circuits on the on/off state of the nMOS switch, and the passive elements are determined simultaneously so that the class E conditions are satisfied. This idea is extended to the design considering the nMOS model fully, using circuit simulator [3], [4]. These methods determine the design parameters and the steady-state responses simultaneously, which degrades the robustness of the algorithms.

In this paper, we propose an optimization procedure of class E amplifier using Particle Swarm Optimization (PSO). PSO is known as a global optimization procedure. Therefore, the proposed method is robust to find the design parameters of class E amplifier. In our approach, the behavior of the class E amplifier is analyzed by HSPICERF [5], which is a standard commercial steady-state circuit analysis tool. Therefore, we expect to include the characteristics of MOSFET included in the class E amplifier fully. We define a cost function for determining the design parameters of class E amplifier. Some passive elements of the circuit are determined by the PSO. By the simulation, it is confirmed that the adjusted circuit behaves as a class E amplifier certainly. The proposed method is very simple, which would make the design of class E amplifiers easy.

2. Class E Amplifier

A class E amplifier is basically configured as shown in Fig. 1. The circuit consists of an input voltage \( V_D \), a dc-feed inductor \( L_C \), an nMOS switch \( S \), a shunt capacitor \( C_S \) to the nMOS switch, a series resonant circuit composed of the inductor \( L_0 \) and capacitor \( C_0 \), and the output resistor \( R \). To achieve the high-efficiency, all the losses occur during switching must be minimized, which demands that the drain-source voltage becomes zero when the switch closes. Furthermore, it is necessary that the time derivative of the switch voltage, which is equal to the current flowing...
through the capacitor $C_0$, is also to be zero at the switching moment [1]. As a result, the conditions as a class E amplifier are obtained by

$$v_s(0) = 0,$$

$$\frac{dv_s}{dt} \bigg|_{t=0} = 0.$$

(1)

(2)

Figure 2 shows a typical waveform of the switch voltage $v_s$ which smoothly lands into the ground at $T$ and $2T$ without switching losses. In order to fulfill the conditions (1) and (2), the design parameters such as values of the passive elements and device parameters of the MOSFET $S$ should be adjusted optimally. Moreover, the conditions must be satisfied on the steady-state, which make the design of class E amplifier difficult.

Class E amplifier has a high $Q$ value, which means that long transition continues until it reaches the steady-state. Therefore, we need an expensive computational cost of the analysis of class E amplifier to confirm whether the class E conditions (1) and (2) are satisfied or not. This is prohibited from using the transient simulation based on a numerical integration formula, thus, a method for finding the steady-state solution would be used. The methods of steady-state analysis of nonlinear circuits are categorized into time and frequency-domain methods. In the frequency-domain method, all the waveforms are assumed by sum of kernel functions such as Fourier series. The input of class E amplifier is a pulse waveform. Hence, we should not use the frequency-domain methods to analysis of class E amplifier. The shooting Newton method, which is a time-domain method, is suitable for this analysis. In the proposed optimization procedure, HSPICE RF is used [5]. HSPICE RF includes not only the shooting Newton analysis (SN) of HSPICE RF is carried out. Otherwise the update is skipped.

3. Determining Design Parameters

3.1. PSO

PSO is a method for optimization without explicit knowledge of the gradient of problem to be optimized. Since the class E amplifier is analyzed by a circuit simulator in our approach, a cost function for the optimization can not be explicitly written [2]. Since the gradient is not easily obtained, PSO becomes a good tool for finding the optimum design parameters of the class E amplifier.

The update rule of PSO used for determining the design parameters of class E amplifier is described by

$$\mathbf{x} \leftarrow \mathbf{x} + \mathbf{v},$$

$$\mathbf{v} \leftarrow \mathbf{v} + c_1 r_1 (\mathbf{x} - \mathbf{x}) + c_2 r_2 (\mathbf{x} - \mathbf{x}),$$

(3)

(4)

where $\mathbf{x}$ and $\mathbf{v}$ are respectively the position and velocity of particle. $\omega$ is an inertia. $c_1$ and $c_2$ imply the ratio of particles in a group which turn to a good position. $r_1$ and $r_2$ are random numbers in $[0, 1]$. $\mathbf{x}$ is the best position for all the particles. $\mathbf{x}$ is the current best position of the particle.

3.2. Implementation

To apply the PSO to determining the design parameters, it is necessary to define the cost function. The conditions as a class E amplifier are related with $v_s(0)$ and $dv_s/dt|_{t=0}$ as (1) and (2). However, $dv_s/dt|_{t=0}$ is more sensitive to the parameter changes than $v_s(0)$. Therefore, the cost function including $[dv_s/dt|_{t=0}]$ can not be defined to determine the design parameters to be optimal. We use $v_s$ only and define the cost function.

The cost function is defined by

$$cost = \frac{1}{N} \sqrt{v_s(0)^2 + \ldots v_s(T_1)^2},$$

(5)

where $T_1$ is the pulse width of the input voltage $D_s$ of Fig. 2 and $N$ is the number of points in $[v_s(0), v_s(T_1)]$. The PSO algorithm minimizes (5) changing the design parameters. The way to evaluate (5) is summarized below.

1. A new position is selected by (3) and (4).

2. If components of the position of particle correspond the passive elements of the class E amplifier and are all non-negative, the shooting Newton analysis (SN) of HSPICE RF is carried out. Otherwise the update is skipped.

3. If the shooting Newton method converges, the root mean square value (5) which defines how the position is optimal, is automatically calculated using a HSPICE RF command (Measure). Otherwise the update is skipped.

4. Results

To design the class E amplifier, we defined the following parameters [2]:

1. $\omega = 2\pi f$.

2. $\omega_0 = 2\pi f_0 = 1/\sqrt{L_0 C_0}$.

3. $Q = \omega L_0 / R$.

4. $A = f_0 / f = \omega_0 / \omega$.

5. $B = C_0 / C_S$.

6. $H = L_0 / L_C$.

As a specification, $f = 1.0 [\text{MHz}]$, $V_D = 5.0 [\text{V}]$, $R = 5.0 [\Omega]$, $Q = 10.0$, $H = 0.001$, $L_C = 7.90 [\text{mH}]$, and $L_0 = 7.90 [\mu\text{H}]$ were given. As a result, $C_S$ and $C_O$ have to be determined only.
In (3) and (4), $c_1 = c_2 = 1$ were used and $w_1$ was a random number in $[0, 1]$. Using 25 particles, the PSO was carried out until 30 iterations, where all the particles were updated at 1 iteration, that is, a particle was updated 30 times maximum. Figures 3(a)-3(c) show the positions of particles of the PSO algorithm. The particles concentrate in a position with increase of the iterations. After 30 iterations, we obtained the best position (solution); $C_S = 3.64[nF]$ and $C_O = 5.09[nF]$. Figure 4 shows the simulation results of the class E amplifier using HSPICERF. The switching voltage $v_s$ dumps smoothly around $t = 0$ so that the switching losses are almost zero. The output voltage is almost sinusoidal waveform. The circuit adjusted by the PSO certainly behaves as a class E amplifier.

To design the class E amplifier, the cost function (5) was used. However, the cost function is related with one of the class E conditions only, that is, (1). On the other hand, in order to investigate the effect of (2) for the optimization, we defined the cost function:

$$cost = \frac{1}{N} \sqrt{\sum s_i(t_0)^2 + \ldots + s_i(T_i)^2},$$

and run the PSO algorithm. In (6), $i_s$ is the current which flows through the capacitor $C_S$ in Fig. 1. Therefore, the cost function is associated with (2). The figures 4(a)-4(c) show the positions of the particles. We can see that the PSO fails to capture the optimal position. PSO is known as a global optimization method. Hence, a problem which PSO fails is difficult or the cost function of which is not suitable. This means that the condition (2) may not be necessary for optimization algorithms to design a class E amplifier.

The simulation of class E amplifier needs a large cost, even if HSPICERF is used. The PSO algorithm needed 4,477 [sec.] on Intel Pentium 4 CPU 2.40 [GHz] with 2 [GB] memory, where CentOS 5.4 was used. In this example, we provided suitable initial positions as shown in Fig. 3(a), thus, the particles concentrate at the 30th iteration. However, if suitable values were not given, many iterations would be necessary to get a good solution. Therefore, we should improve efficiency of the PSO algorithm for designing the class E amplifier.

5. Conclusions

The optimization procedure of class E amplifier has been proposed, where the PSO algorithm is used. To adjust the design parameters of class E amplifier, the cost function for the optimization is defined. In our approach, a standard
Figure 5: Positions of particles in the PSO using the cost function (6). (a) Positions at the 5th iteration. (b) Positions at the 10th iteration. (c) Positions at the 30th iteration.

commercial steady-state analysis tool is used. Since the device model is realistic, we can determine the design parameters suitably. HSPICERF includes various functions for example, including scattering parameters. Therefore, our approach can include the physical effects of the circuit in detail. However, when the circuit is analyzed considered such effects, the simulation needs a lot of CPU times. Therefore, we must improve efficiency of the PSO algorithm. This is our future work.

References


Improvement of Tug-of-war Model for Two-armed Bandit Problem: Biologically Inspired Computing Method for Nonlocally-correlated Parallel Searches

Song-Ju Kim†, Masashi Aono†, and Masahiko Hara†

†Flucto-Order Functions Research Team, RIKEN-HYU Collaboration Research Center, Advanced Science Institute, RIKEN Fusion Technology Center 5F, Hanyang University, 17 Haengdang-dong, Seongdong-gu, Seoul 133-791, Korea
2-1 Hirosawa, Wako-shi, Saitama 351-0198, Japan
Email: songju@riken.jp

Abstract—The “tug-of-war (TOW) model” proposed in our previous studies [1, 2, 3] is a unique method for parallel searches inspired by the photoavoidance behavior of the single-celled amoeba, the true slime mold Physarum. In the TOW model, many branches of the amoeba act as search agents to collect information on light stimulations while conserving the total sum of their resources (volume). We showed that the nonlocal correlation via resource conservation can be advantageous to manage the “exploration–exploitation dilemma” for solving the multi-armed bandit problem.

In this study, we investigate the effect of the information from the other branch on the TOW model’s performance, for the purpose of improving the model. We improve the TOW model so that it can exhibit better performances regardless of the reward probabilities.

1. Introduction

We consider that there must be some crucial differences between biological organisms and digital computers with respect to their information processing. We expect that biological organisms are good at dealing with some kind of problems. In the amoeba’s body (the true slime mold Physarum (Fig. 1A)), a constant amount of intracellular protoplasmic sol shuttles through tubular channels, while its extracellular gel layer (ectoplasm), like a sponge, rhythmically oscillates the contraction tension to squeeze and absorb the sol (Fig. 1B). While the amoeba oscillates its branches to collect environmental information, the volume of the sol flowing through its body remains constant, unless nutrients are provided. We are interested in how this physical conservation law affects the information processing of the amoeba [1, 2, 3, 4]. To elucidate this issue, we considered the “multi-armed bandit problem” because it is related to the difficulties of biological organisms faced while adapting to uncertain environments.

In this study, we focused on the two-armed bandit problem stated as follows. Consider a slot machine that has 2 arms. Both arms have individual reward probabilities \( P_A \) and \( P_B \). At each trial, a player pulls one of the arms and obtains some reward, for example, a coin, with the corresponding probability\(^1\). The player wants to maximize the total reward sum obtained after a certain number of selections. However, it is supposed that the player does not know these probabilities. How can the player gain maximal rewards? The problem is to determine the optimal strategy for selecting the arm which yield maximum rewards by referring to past experiences. In the original form of the problem, the player was allowed to pull only one arm at each trial. However, to explore the advantages of parallel computing, we allowed the player to simultaneously pull both the arms. With this modification, the situation becomes more realistic, as it were a “two-bandit problem.” The new form of the problem considers 2 slot machines \( A \) and \( B \), each having only 1 arm. Machines \( A \) and \( B \) have reward probabilities \( P_A \) and \( P_B \), respectively.

The player has to “explore” many unknown machines to gather much information to determine the best machine. However, these explorations are risky because the player

---

\(^1\)In this study, we assume that each pull results in a reward of fixed size with the given probability. We are dealing with the simplified variant of the general two-armed bandit problem.
may lose considerable rewards that could have been “exploited” from the already-known best machine. Thus, there is a trade-off between “exploration” and “exploitation.” Living organisms generally encounter this “exploration–exploitation dilemma” because they have to survive in an unknown world. In order to survive, organisms need to adapt to the unknown situations by overcoming this dilemma. We speculate that organisms would have developed some efficient methods to overcome this dilemma.

In our previous studies [1, 2, 3], we proposed the “tug-of-war (TOW) model” which is a unique method for parallel searches inspired by the photoavoidance behavior of the true slime mold amoeba. The TOW model is a bio-inspired computing method capable of effectively solving problems without necessarily being a biological model for reproducing an amoeba’s behavior. In our previous reports, we showed that the nonlocal correlation via resource conservation can be advantageous to manage the “exploration–exploitation dilemma” for solving the multi-armed bandit problem. We showed that the average accuracy rate of the TOW model is higher than those of well-known algorithms such as the modified ε-greedy algorithm and modified softmax algorithm. We also showed that the TOW model effectively adapts to a changing environment in which the reward probabilities dynamically switch.

In this study, we investigate the performances of the extended version of the TOW model for the two-armed bandit problem, and show that the optimized weight parameters depend on reward probabilities. This fact suggests that the performance of the TOW model can be further enhanced. We propose an improved TOW model which can exhibit better performances regardless of the reward probabilities.

2. Models

2.1. Tug-of-war Model

On the basis of the photoavoidance behavior of an amoeba, we proposed the tug-of-war (TOW) model in our previous studies [1, 2, 3]. Consider that the shape of an amoeba is like a slug, as shown in Fig. 2. Variables \( x_A \) and \( x_B \) correspond the volume increments in branch A and B, respectively. If \( x_A \) (\( x_B \)) is greater than 0, we consider that the amoeba selects A (B). Subsequently, light stimuli are applied to the branch A (B) with the probability \( 1 - P_A \) \( (1 - P_B) \) as a “punishment,” i.e., an effect opposite to a “reward.” In this model, there can be 4 types of selections: A, B, A and B, and no selection at each time step.

The volume increments \( x_A \) and \( x_B \) are determined by the following difference equations:

\[
x_A(t + 1) = x_A(t) + v_A(t), \quad (1)
\]

\[
x_B(t + 1) = x_B(t) + v_B(t), \quad (2)
\]

\[
v_A(t) = v_A(t - 1) + a_A(t), \quad (3)
\]

\[
v_B(t) = v_B(t - 1) + a_B(t). \quad (4)
\]

Here, \( v_A(t) \) and \( v_B(t) \) denote velocities of the corresponding volume increment, and \( a_A(t) \) and \( a_B(t) \) denote accelerations of the corresponding volume increment.

The internal resource deviation from the constant amount of the total resource \( V \), \( S(t) \), is determined by the following equation:

\[
S(t + 1) = S(t) - (v_A(t) + v_B(t)). \quad (5)
\]

If the initial conditions \( x_A(0), x_B(0), v_A(0), v_B(0) \), and \( S(0) \) are set to zero, the value \( x_A(t) + x_B(t) + S(t) \) will always be zero. This implies that \( S(t) = -(x_A(t) + x_B(t)) \), ensuring the conservation of the total resource \( V \).

In order to incorporate the learning mechanism into this model, we introduced local biases of internal resource \( Q_A \) and \( Q_B \) for the resource on branches A and B, respectively (see the bottom figure in Fig. 2). For every time \( t \), the number of selections and number of stimulations are accumulated in \( Q_X(t) \) such that

\[
Q_X(t) = \mu (N_X - 2 L_X). \quad (6)
\]

where \( N_X \) is the number of selection \( X \) (A or B) until time \( t \), and \( L_X \) is the number of light stimulations on \( X \) (A or B) side until time \( t \). Here, \( \mu \) is the learning parameter.

By referring to the information on the number of selections and number of light stimulations, we assumed that a local bias of the internal resource is formed on each branch A or B. Thus, the local resource deviations \( S_A(t) \) and \( S_B(t) \) are given by

\[
S_A(t) = S(t) + Q_A(t - 1) - Q_B(t - 1), \quad (7)
\]

\[
S_B(t) = S(t) + Q_B(t - 1) - Q_A(t - 1). \quad (8)
\]

This implies that the communication between branch A and branch B is realized via resource conservation.

In the model, accelerations are essential variables (driving force). The acceleration \( a_X(t) \) (\( X = A \) or B) is determined from Table 1; it depends on the local resource deviation \( S_X(t) \) and light ON-OFF condition. The intrinsic
3. Results

3.1. Optimization of Weight Parameters

In order to investigate the effect of $L_X$ in Eq. (6) on the TOW model’s performance, we adopt the following form, instead of Eq. (6):

$$Q_X(t) = \mu \left( N_X - (1 + w) L_X \right).$$

(9)

Here, $w$ is the weight parameter. In the original TOW model, the weight parameter $w$ is always 1.

The above form, Eq. (9), is equivalent to the following form:

$$Q_X(t) = \mu \left( N_X - L_X + w L_Y \right).$$

(10)

because of the fact that $Q_A - Q_B = Q'_A - Q'_B$ in Eqs. (7) and (8). Here, $X = A$ if $X = B$, or $B$ if $X = B$. In the form (10), the first two terms denote the information of success (no light stimulation) in a branch, while the third term denotes the information of failures (light stimulation) of the other branch. The weight parameter $w$ can be interpreted as the contribution weight from the other branch.

How does this parameter $w$ affect the model’s performance? The performance of the model is evaluated in terms of the “accuracy rate”; accuracy rate is defined as the rate of correct (higher probability) selections made until $t$. Figure 3 shows the average accuracy rates for the models with $P_A = 0.4$ and $P_B = 0.6$ (circle), $P_A = 0.4$ and $P_B = 0.7$ (square), $P_A = 0.4$ and $P_B = 0.8$ (triangle up), $P_A = 0.3$ and $P_B = 0.6$ (diamond), and $P_A = 0.45$ and $P_B = 0.6$ (triangle down), respectively. The horizontal axis denotes the weight parameter $w$, and the vertical axis denotes the average accuracy rate at the number of selections = 500 for 1,000 samples of the TOW model. At each weight parameter $w$, the learning parameter $\mu$ was optimized in order to obtain the highest average accuracy rate. The elliptic curve on each line denotes its peak. The optimal $w$ (peaks in Fig. 3) depend on reward probabilities. This fact suggests that the performance of the TOW model can be further enhanced. We can summarize the dependence as follows: (I) The optimal $w$ is 1.0 if reward probabilities have a symmetry (the mean value of reward probabilities is 0.5.). (II) If the mean value is larger (smaller) than 0.5, the optimal $w$ is also larger (smaller) than 1.0. (III) The shift of the optimal $w$ from 1.0 is proportional to the deviation of the mean value of reward probabilities from 0.5.

From Fig. 3, the parameter $w = 1.0$ is the best choice except for the cases in which the problem is difficult ($|P_A - P_B|$ is small) and does not have the symmetry ($P_A + P_B \neq 1$). We call these cases “non-symmetric difficult problems.” Figure 4 shows the average accuracy rate for such case, namely the model with $P_A = 0.6$ and $P_B = 0.65$ (square).
In the same way, the elliptic curve on each line denotes its peak. In this case ($P_A = 0.6$ and $P_B = 0.65$), the peak is about 0.732 at $w = 1.4$. This value is not a little larger than 0.690 at $w = 1.0$ (original TOW). Therefore, we have to improve the TOW model so that the model can exhibit better performance even for such cases. It is easy to develop the model which can exhibit the best performances if we know the reward probabilities $P_A$ and $P_B$. However, we can use only estimates for those probabilities, such as $q_{X} = \frac{N_X - L_X}{N_X}$, ($X = A$ or $B$).

3.2. Improved TOW Model

We investigated performances of several improved models which were found by using heuristic method, and eventually found the two best forms. If we substitute $w=1.0+\gamma D$ to Eq. (9), we can obtain the following equation:

$$Q_X(t) = \mu \left( N_X - 2 L_X - \gamma D L_X \right).$$

(11)

Here, $\gamma$ is a parameter, and $D$ is the deviation from the symmetry defined as follows:

$$D = \frac{1}{2} \left( q_A + q_B - 1 \right),$$

(12)

$$= \frac{1}{2} \left( 1 - \frac{L_A}{N_A} - \frac{L_B}{N_B} \right).$$

(13)

Computer simulation studies showed that $\gamma = 1.0$ is the best for its performance.

If we substitute $\gamma=2$ and $D = (1/2 - L_X/N_X)$ to Eq. (11), instead of Eq. (13), we can obtain the following simplified form:

$$Q_X(t) = \mu \left( N_X - 2 L_X \left( 3/2 - L_X/N_X \right) \right).$$

(14)

In fact, these two improved models, Eqs. (11, 13) and (14), can exhibit better performances. For example, in the case of $P_A = 0.6$ and $P_B = 0.65$, the average accuracy rate at the number of selections $= 500$ is 0.708, which is larger than 0.690 in the original TOW model although the value is still smaller than 0.732 at $w = 1.4$. In the other cases except for “non-symmetric difficult problems,” these models exhibit almost the same performances as the original TOW model.

4. Conclusions and Discussions

We improved the “tug-of-war (TOW) model” which conducts unique parallel searches using many nonlocally correlated search agents. The conservation law entails a “nonlocal correlation” among the branches, i.e., volume increment in one branch is immediately compensated by volume decrement(s) in the other branch(es). This nonlocal correlation was shown to be useful for decision making in the case of a dilemma. In our previous reports [1, 2, 3], the average accuracy rate of the model is higher than those of well-known algorithms such as the modified $\varepsilon$-greedy algorithm and modified softmax algorithm. Moreover, the model flexibly adapts to changing environments, a property essential for living organisms surviving in uncertain environments.

In this study, we investigated performances of the extended version of the TOW model for the two-armed bandit problem. We added the weight parameter $w$ to the original TOW model, and show that the optimized weight parameters depend on reward probabilities. This implied that the TOW model can be improved for better performance. Using heuristic method, we developed improved TOW models which can exhibit better performances regardless of the reward probabilities. The improved models were the best improvements among those we have ever examined. However, whether a better improvement will be possible is still open problem.

This TOW model is applicable to the Monte-Carlo tree search which is used in algorithms for the “game of GO” [5, 6]. We believe that the variant of the TOW model will become one of the best promising approaches to develop the effective algorithm due to its parallelism and non-local correlation between search agents.

References


Asymptotic behaviour of blinking (stochastically switched) dynamical systems

Martin Hasler

School of Computer and Communication Sciences,
Ecole Polytechnique Federale de Lausanne, Switzerland
Email: martin.hasler@epfl.ch

Abstract—We discuss the behavior of continuous-time dynamical systems that have external input signals that are constant in small time intervals and that can only take the values 0 and 1. They can be interpreted as switching between 2m different dynamical systems if there are m such input signals. Switching is supposed to be fast with respect to the time constants of the (non-switched) different systems. Therefore, one expects that the switched (blinking) system behaves like the time-averaged system. More precisely, we suppose the switching to be stochastic such that the value of external signal at a certain time interval is a random variable and that all these random variables are independent.

In general, the solutions of the blinking and the averaged system starting from the same initial state, stay close together if the switching is fast, but this property holds only for finite time. However, if the solution of the averaged system converges to an attractor, this is also true for the corresponding rapidly switched blinking system under some weak hypotheses. In general, the solutions do not stay close together forever, but they converge to the same attractor.

Strictly speaking, this is only possible if the attractor of the averaged system is an invariant set to the blinking system. However, if this is not the case, the solution of the blinking system will still come close to attractor of the averaged system and stay close in a probabilistic sense. Furthermore, if the averaged system has more than one attractor (multi-stability), there is a small, but positive probability that the solution of the blinking system converges to another attractor than the solution of the averaged system.

Hence, there are 4 cases to distinguish, according to whether or not the averaged system has more than one attractor and whether or not the attractor(s) of the averaged system is (are) invariant under the blinking system. We give for each case an example and we prove a theorem that characterizes the relation between the asymptotic behavior of the averaged and the blinking system. Depending on the case, convergence of the blinking system to the attractor of the averaged system is either in the strong sense (for almost all switching sequences) or in the weak sense (with probability converging to one) if switching is fast.
Entropy-Based Measures of Causality and Application to Epilepsy
Dimitris Kugiumtzis

Department of Mathematical, Physical and Computational Sciences,
Faculty of Engineering, Aristotle University of Thessaloniki, Greece
URL: http://users.auth.gr/dkugiu, E-mail: dkugiu@gen.auth.gr

Abstract—Among different measures of coupling and causality, measures based on entropies have gained much attention recently, such as the transfer entropy (TE) and its extensions based on permutation entropies, i.e. the so-called symbolic transfer entropy (STE) and the transfer entropy on rank vectors (TERV). All these measures make use of univariate embedding of each of the two time series. Very recently, we proposed a measure for coupling and causality that is derived from mixed embedding, which relies on information criteria regarding past, current and future states. The components of the mixed embedding vector indicate the presence of information transfer, and a measure is formed to quantify it, called mutual information from mixed embedding (MIME). We compare the measures from univariate embedding, TE, STE and TERV, and the measure from multivariate embedding, MIME. For this, we make simulations on a number of known nonlinear dynamical systems. Further, we apply the four measures to EEG records containing preictal and ictal states. It turns out that MIME is rather robust and conservative in detecting causal effects while the other three measures are positively biased indicating often false causal effects.

keywords: multivariate time series, Granger causality, information flow, mixed embedding, epileptic EEG

1. Introduction

In the study of complex systems, such as climatic processes, financial markets and brain dynamics, it is important to identify and estimate the strength and direction of inter-dependence among the interacting components, measured as multivariate time series. Among different measures of uni- or bi-directed dependence (phase synchronization, Granger causality using prediction models or coherence measures, and local geometric properties in reconstructed state spaces for the driving and driven systems, e.g. see [1, 2]), we concentrate on the class of information-based causality measures [3, 4, 5]. The most popular of this type of causality measures is the transfer entropy (TE) [3]. We also consider the variant of TE using rank vectors instead of sample vectors [4] and the correction of this [6]. The main objective of this work is to compare the transfer entropy measures with a measure we proposed very recently, derived from mutual information conditioned on the components of the driving system present in a mixed embedding [7].

We conduct a simulation study to assess the ability of the measures to detect correctly the direction and strength of coupling, using some known chaotic systems. Then we assess the measures on a real-world application, the investigation of the information flow in brain areas before and after epileptic seizures.

2. Information-based Causality Measures

Let \{x_t\} and \{y_t\}, \ t = 1, \ldots, n, denote two simultaneously observed time series derived from the dynamical systems X and Y, respectively. Using the method of delays, the reconstructed points from the two time series are \(x = [x_\tau, x_{\tau+\tau_1}, \ldots, x_{\tau+(m-1)\tau_1}]\) and \(y = [y_\tau, y_{\tau+\tau}, \ldots, y_{\tau+(m-1)\tau}],\) allowing different delay parameters \(\tau_1, \tau\) and embedding dimensions \(m_1, m_2\) for the systems X and Y, respectively.

Transfer entropy (TE) quantifies the information flow from X to Y by the amount of information explained in Y at T steps ahead by the state of X, accounting for the concurrent state of Y [3]. In terms of the Shannon entropy \(H(x) = \sum p(x) \log p(x)\), TE for the causal effect of system X on system Y is defined as

\[
\text{TE}_{X \rightarrow Y} = H(x_t, y_t) - H(y_t, y_t^\tau) - H(\hat{y}_t^\tau) - H(y_t) - H(\hat{y}_t^\tau) - H(y_t). \tag{1}
\]

For the estimation of the entropy terms, we use the approach of nearest neighbors [8]. To account for the effect of X on the evolution of Y over a time horizon \(T\), extend the single one-step ahead mapping to the future vector \(y_t^\tau = [y_{\tau+1}, \ldots, y_{\tau+T}]\) in the definition of TE in eq.(1).

In [4], the symbolic transfer entropy (STE) is defined as TE but on rank-points formed by the reconstructed vectors of X and Y. Each sample reconstructed vector, say \(w_r\), in eq.(1) is replaced by the rank-point \(\hat{w}_r = [r_1, r_2, \ldots, r_m]\), where \(r_j \in [1, 2, \ldots, m]\) are the ranks of the vector components \(j = 1, \ldots, m\). Following this sample-point to rank-point conversion, \(y_t^\tau\) is replaced by the rank point at time \(t + T, \hat{y}_t^\tau\), and STE is defined as

\[
\text{STE}_{X \rightarrow Y} = H(\hat{x}_t, \hat{y}_t) - H(\hat{y}_t^\tau, \hat{x}_t, \hat{y}_t) + H(\hat{y}_t^\tau, \hat{y}_t) - H(\hat{y}_t). \tag{2}
\]

where the entropies are computed from the estimated probability mass functions of the rank-points.

In [6], it was shown that instead of replacing \(y_t^\tau\) with \(\hat{y}_t^\tau\), as done in STE, it is more appropriate to use \(\hat{y}_t^\tau = \hat{y}_t^\tau\)
the ranks of \( y_{t} \) in the augmented vector \( [y_{t}, y_{t+1}, \ldots, y_{t+T}] \). The proposed measure of transfer entropy on rank vectors (TERV) is

\[
\text{TERV}_{X \rightarrow Y} = H(\hat{x}, \hat{y}_{t}) - H(\hat{y}_{t}, \hat{x}) + H(\hat{y}_{t+1}, \hat{x}) - H(\hat{x}),
\]

In the following, we present briefly the causality measure of mutual information from mixed embedding (MIME), which relies on a non-uniform embedding scheme for the bivariate time series \( \{x_{t}, y_{t}\}, t = 1, \ldots, n \) with the purpose to explain best the future of \( Y \), as given by the future vector \( y_{t+1} \) [7]. The rationale of the proposed embedding scheme is that the components of the derived embedding vector, denoted \( z_{t} \), must be least dependent to each other and able to explain best \( y_{t+1} \).

The components of the embedding vector \( z_{t} \) are to be selected from the set of delayed components \( Z_{n} = \{x_{t}, x_{t-1}, \ldots, x_{t-L_{x}}, y_{t-L_{y}}, \ldots, y_{t} \} \), where \( L_{x}, L_{y} \) are the maximum lags for \( X \) and \( Y \). The progressive embedding scheme starts with an empty embedding vector \( \tilde{z}_{0}^{j} \). Then at a step \( j \), the component \( z_{j}^{j} \in Z_{n} \setminus \tilde{z}_{j-1}^{j-1} \) to be added to \( \tilde{z}_{j-1}^{j-1} \) is the one that maximizes the mutual information (conditioning) for the current components in \( \tilde{z}_{j-1}^{j-1} \). The criterion for the selection of \( z_{j}^{j} \) reads

\[
\max_{z_{j}^{j}} \left( I(y_{t+1}; z_{j}^{j} | \tilde{z}_{j-1}^{j-1}) \right) = I(\tilde{z}_{j}; z_{j}^{j}) - I(\tilde{z}_{j}^{j}; z_{j}^{j}) > A,
\]

for a threshold \( A \leq 1 \). The closer \( A \) is to 1 the more relaxed is the stopping criterion allowing for more components to enter in the form of \( z_{t} \). On the other hand, a smaller \( A \) results in lower dimensions of the mixed embedding. Here, we use \( A = 0.95 \), which was found in [7] to be a good trade-off value.

The embedding vector from the mixed embedding scheme may contain components from both \( X \) and \( Y \), and can be represented in terms of these two sets of components as

\[
z_{t} = [z_{t}^{X}, z_{t}^{Y}] = [x_{t-L_{x}}, x_{t-L_{x}-1}, \ldots, x_{t-L_{x}-L_{y}}, y_{t-L_{y}}, \ldots, y_{t-L_{y}}].
\]

Then the measure MIME is defined as

\[
\text{MIME}_{X \rightarrow Y} = 1 - \frac{I(y_{t+1}; z_{t}^{X})}{I(y_{t+1}; z_{t})} = \frac{I(y_{t+1}; z_{t}^{X} | z_{t}^{Y})}{I(y_{t+1}; z_{t})}.
\]

MIME\(_{X \rightarrow Y}\) measures the information of \( Y \) explained only by components of \( X \) in the embedding vector, normalized by the total mutual information (in order to give a value between 0 and 1). If \( z_{t} \) contains no components from \( X \), then MIME\(_{X \rightarrow Y} = 0 \) and \( X \) has no effect on the future of \( Y \).

In the computation of MIME, first the optimal representation of the driving system \( X \) and response system \( Y \) in the mixed embedding is found, giving the \( m_{s} \) and \( m_{r} \) components in \( z_{t} \), respectively. This is a main difference of MIME from the other information-based causality measures, for which \( m_{s} \) and \( m_{r} \) are decided a priori as the embedding dimensions of separate fixed delay embeddings for \( X \) and \( Y \), commonly setting \( m_{s} = m_{r} \) [3, 4, 5]. Indeed we have found that the outcome of TE, STE and TERV depends strongly on the choice of \( m_{s} \) and \( m_{r} \) [2, 6], and MIME overcomes this problem using the progressive embedding scheme at the cost of significantly larger computation time.

### 3. Simulations on Chaotic Coupled Systems

In the simulations below we always set \( \tau_{x} = \tau_{y} = 1 \) and use 10 neighboring points for the estimation of entropies in TE and MIME.

We first evaluate the measures on the driver-response Henon system given by

\[
x_{n+1} = 1.4 - x_{n}^{2} + 0.3x_{n-1},
\]

\[
y_{n+1} = 1.4 - (Cy_{n} + (1 - C)y_{n}^{2}) + 0.3y_{n-1},
\]

for coupling strength \( C \) taken the values 0, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6. For each \( C \) we generate 100 realizations and compute the measures for \( T = 1 \). For MIME we set \( L_{x} = L_{y} = 5 \). The results on TE, STE and TERV vary significantly with the choice of \( m_{s} \) and \( m_{r} \), and we even observe larger values in the wrong coupling direction when \( m_{s} \) is much smaller than \( m_{r} \). We further show results on TE, STE and TERV only for \( m_{s} = m_{r} \), which is the choice typically met in the works on these measures. In Fig. 1, we show results for \( m_{s} = m_{r} = 3 \) and for noise-free and noisy time series of small and large length \( n \). We note that regardless of \( n \) and the presence of noise all the measures detect well the causal effect \( X \rightarrow Y \) obtaining positive values, but STE and TERV are also positive for \( C = 0 \) and in the wrong direction of coupling, whereas TE gets positive only for noisy data and large \( C \), in particular when \( n = 4096 \) (see Fig. 1d). On the other hand, MIME\(_{x \rightarrow y}\) is always zero and actually this holds for all 100 realizations, as shown in Fig. 1b for the standard deviation (SD) of the measures. MIME has also the smallest SD for the direction \( X \rightarrow Y \) giving a rather stable and consistent estimation of the causal effect. It should be noted that for very weak coupling (\( C = 0.05 \)), MIME does not detect the driving of \( X \) as no components enter the form of the constructed embedding vector, while TERV in particular is larger in the correct direction. As argued in [7], MIME can be more sensitive to weak coupling by increasing the threshold \( A \), but then there is higher probability of having irrelevant components in the form of the embedding vector by chance, which then generates positive MIME\(_{y \rightarrow x}\) as well. Thus using \( A = 0.95 \), MIME turns out to be a more strict but stable measure of causality.
Figure 1: (a) The mean of the measures, as shown in the legend, from 100 realizations of the unidirectionally coupled Henon map for the correct direction $X \rightarrow Y$ (black line) and the opposite direction (gray line, cyan online). The time series length is $n = 1024$ and the data are noise-free. (b) The standard deviation (SD) of the measures in (a). (c) As in (a) but when adding normal white noise to both time series with SD being 20% of the data SD. (d) As in (a) but for 20% additive normal white noise.

The same results on MIME are found from the simulations on the next system, the coupled R"ossler–Lorenz system given by

$$
x_1(t) = 6(-y_1(t) - z(t_1))
$$

$$
y_1(t) = 6(x_1(t) + 0.2y_1(t))
$$

$$
z_1(t) = 6(0.2 + x_1(t)z_1(t) - 5.7z_1(t))
$$

$$
x_2(t) = 10(y_2(t) - x_2(t))
$$

$$
y_2(t) = 28x_2(t) - y_2(t) - x_2(t)z_2(t) + C_1y_1(t)^2
$$

$$
z_2(t) = -8/3z_2(t) + x_2(t)y_2(t),
$$

where the driving time series regards $y_1$ and the response $y_2$, for $C$ being 0, 0.5, 1, 1.5, 2, 3, 4. We compute the measures for $T = 3$ to account for driving effects over several steps ahead. This complicates the computation of TE, STE and TERV because the entropy terms take as arguments larger vectors, introducing more bias in the estimation of the entropies and subsequently the estimation of the measures. The bias is particularly large for small time series and increases with the addition of noise as shown in Fig. 2 for $m_x = m_y = 3$ and $n = 1024$. It also increases with the embedding dimensions, particularly for STE and TERV. In the computation of MIME we let $L_x = L_y = 15$ to account for all significant delays. Again MIME$_{Y \rightarrow X}$ lies at the zero level and increases slightly only for noisy data and large $C$ (see Fig. 2b). This zero level of MIME$_{Y \rightarrow X}$ combined with the significantly positive MIME$_{X \rightarrow Y}$ for all $C > 0$ suggests a reliable detection of the direction of the causal effect and estimation of its strength. The other measures fail to provide reliable estimation due to the presence of biased positive measure values for the direction $Y \rightarrow X$. In particular, STE tends to give larger values for the wrong direction and this is corrected by TERV. The variance of TE, STE and TERV is again at the same (significant) level in both directions, blurring the observed difference of the mean values of TE and TERV in the two directions in Fig. 2, whereas MIME$_{Y \rightarrow X}$ is again zero for all 100 realizations except for the noisy data and large $C$, where it has also large variance.

4. Application of the Causality Measures to epileptic EEG

The application regards human scalp electroencephalographic (EEG) recordings from several hours before epileptic seizure onset to many minutes after the seizure end. We consider anti-hemispheric channels in pairs from the left and right frontal (F3 and F4), central (C3 and C4), temporal (T7 and T8) and parietal lobe (P3 and P4). We used 6 records of generalized tonic-clonic seizures of different patients. Each EEG record was split in segments of 30 sec (sampling time 0.01 sec) and the measures TE, STE, TERV and MIME were computed on each EEG segment of channel pairs ($T = 1, L_x = L_y = 20$). For TE, STE, TERV, $m_x = m_y$ were set to 3, 6 and 10, giving varying results: the STE and TERV profiles over the whole recording increased a lot with the embedding dimension, while the TE profile was always at the same level but varied in shape (see Fig. 3a and b for the TE profiles from one EEG record). The STE and TERV measures produced the same profile with small changes for $m_x = m_y = 3$ and being almost identical for larger embedding dimensions. Therefore only the TERV profile is shown for the same episode in Fig. 3c. The profiles of TE, STE and TERV are positive for all segments and channel pairs, and for both directions, so that one would conclude that information flows constantly from left to right and vice versa at all brain areas and regardless of the epileptic state (preictal, ictal, postictal). Given the presence of positive bias in all these measures, as observed in the simulations, this conclusion may not be correct and then one should look at differences in the level of these measures in the two directions in order...
to find specific patterns of information flows. For the specific profiles in Fig. 3, it seems that there is larger causal effect (information flow) from left to right central, parietal and temporal lobes than in the opposite direction. This is exactly what we observe clearly with the MIME measure as MIME is positive in the left to right direction at these lobes and zero in the opposite direction in almost all segments. The same characteristics were observed in the other 5 epileptic records. For the last preictal period in Fig. 3, MIME stays at zero for all directions and channel pairs, whereas the TE measure gets large as the embedding dimension increases in both directions and all channel pairs (and STE and TERV become even larger, not shown here), suggesting that there are other effects than driving, causing this large increase of TE, STE and TERV but not affecting MIME. This pattern at the late preictal period was absent in the other 5 epileptic records. No significant differences were observed after the seizure onset with TE, STE and TERV, whereas MIME often turned to zero.

5. Discussion

It has been already pointed in the literature that the causality measures, including the measures TE, STE and TERV, contain positive bias that can be attributed to effects other than the driving effect, mainly the dynamics of the individual systems, the state space reconstruction, the time series length and noise. We observed the positive bias of the three measures both in the simulations with the coupled Henon maps and the Rössler–Lorenz system, as well as in the application to epileptic EEG records. On the other hand, the causality measure MIME, which is a normalized conditional mutual information derived from a non-uniform mixed embedding, turns out to be a less biased estimate of causal effect. MIME has the nice property of being exactly zero when no causal effect is found, i.e. no components of the driving time series are present in the vector of mixed embedding. This property is particularly useful in real-world applications, as it detects only significant driving effects, whereas TE, STE and TERV are always positive at a varying level due to bias, so that it cannot be concluded whether the driving effect is true. Therefore opposite to MIME, these measures, as any other causality measure, cannot be applied without including a bias correction or a significance hypothesis test.

Acknowledgments

I thank Pål G. Larsson for providing the EEG data.

References


Nonlinear and Nonparametric Models for Forecasting the US Gross National Product

Siddharth Arora¹,², Max A. Little²,³ and Patrick E. McSharry¹,²

¹Smith School of Enterprise and the Environment, University of Oxford, UK
²Oxford Centre for Industrial and Applied Mathematics (OCIAM), Mathematical Institute, University of Oxford, Oxford, UK
³Oxford Centre for Integrative Systems Biology, Department of Physics, University of Oxford, Oxford, UK

Email: (S. Arora) arora@maths.ox.ac.uk, (M.A. Little) littlem@physics.ox.ac.uk, (P.E. McSharry) patrick.mcsharry@smithschool.ox.ac.uk

Abstract– Nonlinear models, like the self-exciting threshold autoregressive (SETAR) model and Markov-switching autoregressive (MS-AR) model, have been proposed for modeling the Gross National Product (GNP) time series. Both SETAR and MS-AR, however, require estimation of a large number of parameters relative to the small amount of GNP observations. While modeling the training data reasonably well, these models tend to overfit and perform poorly in terms of out-of-sample forecasting. The aim here is to investigate the efficacy of a novel parsimonious nonparametric and nonlinear model, which can outperform SETAR and MS-AR in terms of out-of-sample GNP forecasting accuracy. As it is important to quantify forecast uncertainty, leading to well-informed policy-making, we generate both point and density forecasts. We evaluate point forecasts using the root mean square error (RMSE) and mean absolute error (MAE), while density forecasts are evaluated using the continuous ranked probability score (CRPS).

1. Introduction

The GNP provides a measure of the economic wealth of a country. It is one of the most commonly used macroeconomic indicators and is reported quarterly. The need for accurate and timely forecasts of the GNP time series stems from the requirements of efficient decision and policy-making. The complex dynamics underlying the GNP time series, along with limited number of post-war observations, however, make the task of generating reliable forecasts quite challenging [1].

The last two decades have witnessed a surge in nonlinear modeling techniques for characterizing economic time series [2, 3, 4, 5]. Deviations from the Gauss-Markov assumptions (linearity, homogeneity and independence) are indications of the presence of nonlinearity in the data generating process. The GNP time series is shown to follow an asymmetric cyclical process, giving rise to different regimes (growth and recession), with growth periods being much longer than recessionary periods [2]. Nonlinear models like the SETAR model [4, 5], and MS-AR models [3] have been employed for forecasting the GNP time series. The rationale behind these nonlinear models (SETAR and MS-AR) is to characterize regimes of growth and recession separately rather than treat them as one, as opposed to the linear models. Hence, SETAR and MS-AR are also known as regime-switching models.

Due to the complex model structure, the parameter estimation procedure for regime-switching models is not straightforward. Also, it has been shown that the forecast performance of SETAR is highly sensitive to the uncertainty in parameter estimates [6]. The major limitation of regime-switching models, however, lies in the fact that there is no clear consensus if these models (SETAR and MS-AR) are better than their linear counterpart (AR models), from the perspective of out-of-sample forecasting [7].

Motivated by Occam’s razor and the impact of accurate GNP forecasts on policy-making and in turn on our financial markets, we seek parsimonious nonlinear models which are based on simple assumptions and can generate accurate point and density forecasts of US GNP. In this paper, we propose a nonlinear model that can be viewed as a hybrid between a nearest neighbour method [8] and the random analogue prediction (RAP) model [9]. The proposed model does not make prior assumptions regarding the true functional form generating the data, hence the model is nonparametric. We compare the competitiveness of the proposed model with classical linear and nonlinear models (AR, SETAR and MS-AR) in generating out-of-sample point and density forecasts.

The paper is arranged as follows. Section 2 gives details of the employed dataset, and discusses the classical models used previously for forecasting US GNP. This section also proposes a novel nonlinear and nonparametric model. Section 3 provides out-of-sample forecast comparison between different models and the random walk benchmark, while section 4 offers conclusion.
2. Linear and Nonlinear Modeling Approaches

2.1. Data

The GNP time series comprises seasonally adjusted quarterly prices of real US GNP (in billions of dollars). The time series contains US GNP recordings from 1947Q1 to 2008Q3 at 2000 prices. The recordings were obtained from the Federal Reserve Economic Data II (FRED II) affiliated with the Federal Reserve Bank of St. Louis. The original data was transformed to give quarterly percentage growth rate, and all the models were estimated using the transformed time series.

2.2. Autoregressive Model

An autoregressive (AR) model is a linear recursive model that belongs to the Box and Jenkins [10] class of models. The AR model characterizes the time series assuming linear relationship between observations. The number of parameters incorporated in the model corresponds to the order of the model. An AR model of order \( p \) is denoted as AR \((p)\) and can be represented using the following form:

\[
x_t = \alpha_0 + \sum_{i=1}^{p} \alpha_i x_{t-i} + \epsilon_t
\]

where \( x_t \) is an observation in the time series at time \( t \), \( \alpha_i \) is a constant, \( \alpha_i \) denotes AR model parameters for \( i = 1, 2, \ldots, p \), while \( \epsilon_t \) is an independent and identically distributed (IID) process with mean zero and variance \( \sigma^2 \). In this paper, we use the Akaike Information Criterion (AIC) proposed in [11] for selecting the AR model order. The AR model parameters are estimated using ordinary least squares (OLS).

2.3. Self-exciting Threshold Autoregressive Model

Threshold Autoregressive (TAR) models were proposed in [12]. The TAR model divides the original time series into a number of distinct non-overlapping regimes, and models every regime as a separate linear process. SETAR models are a subclass of TAR models whereby the threshold variable is forced to be endogenous, i.e. the threshold is chosen from an observation in the time series.

A SETAR model composed of \( N \) number of distinct regimes, for a time series \( x_t \), is denoted as SETAR \((N_1, p_1, p_2, \ldots, p_N)\), and is represented as:

\[
x_t = \alpha_{0j} + \sum_{i=1}^{p_j} \alpha_{ij} x_{t-i} + \epsilon_{tj}
\]

where \( \alpha_{ij} \) are the autoregressive coefficients for a given regime index \( j \) that obeys \( r_{1j} \leq x_t < r_{pj} \), while \( r_{1j} \) for \( j = 1, 2, \ldots, N \), are the thresholds that divide the time series into different regimes, \( p_j \) is model order for the corresponding regime, whereas \( \epsilon_{tj} \) is an IID process with mean zero and variance \( \sigma^2 \). We estimate a bi-regime SETAR model for US GNP as employed previously in [4, 5]. To estimate the model parameters, we varied \( r_1 \) and \( d \) over a grid, and select a particular set of parameter values that minimizes the overall residual sum of squares (RSS). For a given threshold and delay order, the AR model order and model coefficients were estimated by applying OLS in each regime separately.

2.4. Markov-switching Autoregressive Model

The bi-regime Markov-switching model (as proposed in [3]) with AR order \( p \) can be denoted as MS (2)-AR \((p)\), and is defined as:

\[
x_t = \mu(s_t) + \sum_{i=1}^{p} \alpha_i (x_{t-i} - \mu(s_{t-i})) + \epsilon_t
\]

where \( x_t \) is an observation in the time series, \( \alpha_i \) are the AR coefficients for \( i = 1, 2, \ldots, p \), \( s_t \) is a regime variable such that, \( s_t = 1 \) corresponds to growth and \( s_t = 2 \) corresponds to recession. The mean of the process \( \mu(s_t) \) switches between the two regimes, such that \( \mu(1) \) is positive if \( s_t = 1 \), and negative otherwise.

The transition between the two regimes depends on the variable \( s_t \), governed by a first order Markov process with transition probabilities \( P_{ij} = P(s_t = j \mid s_{t-1} = i) \). The parameter vector that needs to be estimated for MS-AR model is \( \theta = \{ \mu(1), \mu(2), \alpha_1, \alpha_2, \ldots, \alpha_p, \sigma, P_{11}, P_{22} \} \). The parameter vector \( \theta \) is chosen so as to maximize the likelihood of the observations. The estimation of this parameter vector was based on maximizing the likelihood using expectation maximization (EM), for details see [13].

2.5. Fraction-Nearest Neighbor Model

The nearest neighbor method [8] is a nonparametric nonlinear approach which relies on the assumption that neighboring states have similar future outcomes (see for example [9, 14]). The fraction-Nearest Neighbor \((f\text{-NN})\) is an adaptive nearest neighbor model that defines a neighborhood size based on the fraction of total points in the training set. Denoting neighborhood in terms of a fraction has the advantage that the neighborhood size adapts to changes in local data density and total number of observations, as opposed to the case when neighborhood size is defined in terms of a fixed radius, or number of neighbors.

The rationale behind the \( f\text{-NN} \) model lies in estimating an optimum neighborhood fraction, and using the collective behavior of the selected neighbors in generating a forecast. This model requires estimation of parameter \( f \),
which denotes the size of the optimum neighborhood. Given an optimum neighborhood size, one can quantify the collective behavior of neighboring states using for example, the mean, or build a local linear model, and use the estimated local model for forecasting the time series.

To estimate the optimum neighborhood size, we first create a set of delay vectors using the time series for GNP quarterly growth rate. The dimension of the delay vector can be viewed as being similar to the order of the model. Having created a set of delay vectors, we compute the distance (Euclidean distance) of the current delay vector, with previous delay vectors.

For estimating the model parameter, we divide the in-sample data into two parts, such that the first half is employed for training and the second-half for validation. The value of \( f \) is varied from a minimum, so as to include only the closest delay vector to the current state, to a maximum, so as to include all states within the training set. For a given \( f \), a set of delay vectors that lie within the neighborhood are selected from the training set. The future observations of these delay vectors are randomly sampled and issued as a density forecast. The value of \( f \) that minimizes the density forecast error (quantified using CRPS) on the validation set is chosen for generating out-of-sample forecasts.

3. Results and Observations

3.1. Performance Scores

The point forecast performance for a particular forecast horizon \( h \) is evaluated using the root mean square error (RMSE) and mean absolute error (MAE), given by:-

\[
RMSE_h = \sqrt{\frac{1}{N - T - h + 1} \sum_{i=T}^{N} (x_i - \hat{x}_{i+h})^2}
\]

\[
MAE_h = \frac{1}{N - T - h + 1} \sum_{i=T}^{N} |x_i - \hat{x}_{i+h}|
\]

where \( RMSE_h \) (MAE\(_h\)) is the RMSE (MAE) at horizon \( h \), \( x_i \) is the actual observation, \( \hat{x}_{i+h} \) is the \( h \)-step ahead forecast, \( T \) is the forecast origin and \( N \) is the length of the time series. In order to evaluate the density forecast performance, we use the empirical form of continuous ranked probability score (CRPS). The CRPS is defined as:-

\[
CRPS = E_F | X - x | - \frac{1}{2} E_F | X - X'| \quad (6)
\]

where \( X \) and \( X' \) are independent samples being drawn from the forecasts density function, each having the same distribution \( F \), \( E_F \) is the expectation with respect to the distribution \( F \), while \( x \) is the actual observation. For details regarding CRPS, please see [15].

3.2. Forecasting Scheme

A series of forecasts for horizons ranging from one to four quarters (one year) ahead is generated based on a rolling forecast scheme, as previously employed for GNP time series by [6]. The time series for GNP quarterly growth rate dating from 1947Q2-1996Q4 is divided into in-sample data from 1947Q2-1996Q4 for training and out-of-sample data 1997Q1-2008Q3 for testing. Using the training set, the specifications of the optimum models were estimated as: AR (4), SETAR (2; 1, 1), MS (2)-AR (5), and \( f = 0.13 \).

All models were estimated using only the training set, and the estimated parameters were then held ‘fixed’ while making out-of-sample forecasts. Hence, each model had access to precisely the same amount of information as every other model during the forecasting stage. We generated density forecasts using the proposed and classical based on Monte Carlo simulations (size 10,000). In order to generate point forecasts, we issued the mean of the forecast distribution at each forecast horizon as the point forecast.

3.3. Forecast Comparison

We evaluate AR, SETAR and MS-AR and the proposed model (f-NN) on out-of-sample GNP data (1997Q1-2008Q3), using three different performance scores (RMSE, MAE and CRPS). We employ the random walk (RW) benchmark, whereby the current value in the time series is issued as a forecast for the next step \( (h=1) \). For multistep ahead forecast \( (h>1) \), the current observation is issued as a \( h \)-step n. Note that we can generate only point forecast using this benchmark, hence there are no RW statistics for CRPS. We generate and evaluate forecasts from one quarter \( (h=1) \) up to four quarters \( (h=4) \) ahead.

The evaluation of point forecasts is presented in Table I and II. When evaluation is based on RMSE (Table I), we find that all models outperform the RW benchmark at all forecast horizons. This signifies that with respect to the benchmark, all models have some skill in forecasting the GNP time series. On comparing different models, we find that f-NN is one of the best models, producing most accurate forecasts for all horizons, except for \( h=2 \), where SETAR is better. When forecast evaluation is based on MAE (Table II), we find that f-NN consistently outperforms all the models on all forecast horizons. SETAR generates more accurate point forecasts than AR and MS-AR.

The superior performance of the proposed model (f-NN) is further highlighted when forecast evaluation is based on CRPS. It is evident from Table III, that density
forecasts from f-NN are superior to any of the classical models. For point forecasts, SETAR is the most accurate model compared to AR and MS-AR, while MS-AR is found to be superior to both SETAR and AR in generating density forecasts on the out-of-sample data.

Table I
Out-of-sample point forecast evaluation using the RMSE, for forecast horizon (\(h\)) ranging from 1 to 4 quarters ahead. Least RMSE (most accurate) value at each horizon is depicted in bold.

<table>
<thead>
<tr>
<th>(H)</th>
<th>RW</th>
<th>AR</th>
<th>SETAR</th>
<th>MS-AR</th>
<th>f-NN</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.780</td>
<td>0.588</td>
<td>0.579</td>
<td>0.592</td>
<td><strong>0.550</strong></td>
</tr>
<tr>
<td>2</td>
<td>0.648</td>
<td>0.572</td>
<td>0.552</td>
<td>0.586</td>
<td>0.560</td>
</tr>
<tr>
<td>3</td>
<td>0.791</td>
<td>0.602</td>
<td>0.598</td>
<td>0.628</td>
<td><strong>0.568</strong></td>
</tr>
<tr>
<td>4</td>
<td>0.712</td>
<td>0.606</td>
<td>0.607</td>
<td>0.632</td>
<td><strong>0.577</strong></td>
</tr>
</tbody>
</table>

Table II
Out-of-sample point forecast evaluation using the MAE. Least MAE (most accurate) value at each horizon is depicted in bold.

<table>
<thead>
<tr>
<th>(H)</th>
<th>RW</th>
<th>AR</th>
<th>SETAR</th>
<th>MS-AR</th>
<th>f-NN</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.670</td>
<td>0.493</td>
<td>0.484</td>
<td>0.498</td>
<td><strong>0.446</strong></td>
</tr>
<tr>
<td>2</td>
<td>0.499</td>
<td>0.471</td>
<td>0.448</td>
<td>0.481</td>
<td><strong>0.446</strong></td>
</tr>
<tr>
<td>3</td>
<td>0.637</td>
<td>0.487</td>
<td>0.472</td>
<td>0.509</td>
<td><strong>0.455</strong></td>
</tr>
<tr>
<td>4</td>
<td>0.569</td>
<td>0.489</td>
<td>0.480</td>
<td>0.511</td>
<td><strong>0.461</strong></td>
</tr>
</tbody>
</table>

Table III
Out-of-sample density forecast evaluation using the CRPS. Least CRPS (most accurate) value at each horizon is depicted in bold.

<table>
<thead>
<tr>
<th>(h)</th>
<th>AR</th>
<th>SETAR</th>
<th>MS-AR</th>
<th>f-NN</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.361</td>
<td>0.357</td>
<td>0.348</td>
<td><strong>0.320</strong></td>
</tr>
<tr>
<td>2</td>
<td>0.357</td>
<td>0.352</td>
<td>0.345</td>
<td><strong>0.325</strong></td>
</tr>
<tr>
<td>3</td>
<td>0.373</td>
<td>0.370</td>
<td>0.369</td>
<td><strong>0.330</strong></td>
</tr>
<tr>
<td>4</td>
<td>0.375</td>
<td>0.370</td>
<td>0.369</td>
<td><strong>0.333</strong></td>
</tr>
</tbody>
</table>

4. Conclusion

We proposed a simple nonlinear and nonparametric model that convincingly outperformed AR, SETAR and MS-AR models on multiple forecast horizons, when evaluated using different performance scores. These results point towards classical parametric models (linear and nonlinear) over-fitting the in-sample data, due to which they fail to generalize on the out-of-sample dataset. Given the need for quantifying uncertainty in forecasts for informed decision and policy-making, we emphasize the need for evaluating models based on their ability to generate accurate density forecasts, as quantified by CRPS.

References


Abstract—In recent years the tools of permutation complexity have found interesting applications in time series analysis. These tools make use of any quantity or functional based on the order relations (permutations) appearing between consecutive elements of a time series. We propose to extend the use of permutation complexity tools to the analysis of complex spatiotemporal dynamics. In this paper we illustrate how interesting properties of a paradigmatic class of models of spatiotemporal dynamics, the Cellular Automata, in particular its topological entropy, can be estimated using permutation complexity tools. We discuss later the implications of this result and how similar ideas can be applied to more general types of spatiotemporal data.

1. Introduction

If the state space of a dynamical system is equipped with a total ordering, this additional structure can be taken into account when analyzing its behavior. The result is what we call permutation analysis, an approach to dynamics and dynamical complexity characterized by conceptual simplicity, an algebraic flavor and computational speed. The tools of permutation analysis include ordinal patterns, order-isomorphy, metric and topological permutation entropy, discrete entropy, and regularity parameters. Permutation analysis has been successfully applied to the estimation of entropies [1, 2], measure of complexity in time series [3], recovery of control parameters of unimodal maps from symbolic sequences [4], characterization of synchronization [5], detection of determinism in time series [6, 7], etc. The next challenge is to extend these applications to physical systems, and more specifically to space-time systems.

In a recent work [8] we proposed a way to apply the tools of permutation analysis to the study of complex spatiotemporal systems. The aim of this paper is to illustrate some of the aspects of these ideas making use of Cellular Automata (CA) as simple models of spatially extended physical systems. Cellular automata were introduced by Ulam [9] and von Neumann [10], and are currently the object of intensive study in mathematical physics, computer science, biology, etc. [11, 12, 13]. For a readable account on cellular automata and their remarkable performance in physical modeling (including turbulence, space-time chaos, symmetry-breaking, and ordering), see, e.g. [14]. The ideas exposed here can later be adapted to more general spatiotemporal data sets by discretizing them both in space and in time, for a detailed description we refer to the reader to Refs. [8, 15].

2. Cellular automata as dynamical systems

For our purposes, it suffices to consider only one-dimensional CA. In this case, the configuration space is the two-sided sequence space

$$S^Z = \{ (s_n)_{n \in \mathbb{Z}} = (\ldots, s_{-k}, \ldots, s_{-1}, s_0, s_1, \ldots, s_k, \ldots) : s_n \in S \}.$$ (1)

The state of cell \(i\) at time \(t \geq 0\) will be denoted \(s_i(t)\). At each time step \(t + 1\), the previous state at each cell \(i\), \(s_i(t) \in S\), is updated according to the local rule \(f : S^Z \to S\) of the form

$$s_{i+1}(t) = f(s_i(t - l), s_i(t - l + 1), \ldots, s_i(t + l)).$$ (2)

The local rule \(f\) leads to a global transition map of the configuration space, \(F : S^Z \to S^Z\) defined in the obvious way:

$$F(\ldots, s_i(t), \ldots) = (\ldots, f(s_i(t - l), s_i(t - l + 1), \ldots, s_i(t + l)), \ldots).$$ (3)

We are going to deal here with finite size CA. The state vector of the system at time \(t\) will be denoted as \(x_t^i\), so

$$x_t^i = (s_i(1), s_i(2), \ldots, s_i(N)),$$ (5)

where \(N\) is the length of the CA.

The topological entropy of a dynamical system provides a good estimation of its complexity. For cellular automata, one can use the following procedure to estimate it [16]. Let \(R(l, t)\) be the number of distinct rectangles of width \(w\) and height (temporal extent) \(t\) occurring in a space-time evolution diagram of \((S^Z, F)\). Fig.1. Then

$$h_{\text{top}}(F) = \lim_{w \to \infty} \lim_{t \to \infty} \frac{1}{t} \log R(w, t).$$ (6)
Therefore, the complexity of the CA can be measured by the number of distinct words or patterns per time unit generated by the global transition map $F$ as time evolves. It follows that

$$h_{\text{top}}(F) \leq 2l \log |\mathcal{S}|,$$

(7)

where $l$ is the neighborhood size of the automaton and $|\mathcal{S}|$ is the cardinality of $\mathcal{S}$.

In next section we are going to describe some tools of permutation analysis and in particular the notion of ordinal pattern. After this, we are going to discuss how they can be used to estimate the topological entropy of a CA.

3. Ordinal patterns and topological entropy

Let $x_0^n = (x_n)_{n \in \mathbb{N}_0}$ be a sequence generated by a source $X$ whose elements $x_n$ belong to a space endowed with a total ordering $\prec$. We say that a length-$L$ block (segment, word, ...) $x_n^{a+L-1}$ is an ordinal pattern if it follows from

$$x_{a+\pi_0} < x_{a+\pi_1} < ... < x_{a+\pi_{L-1}},$$

(8)

where in case $x_i = x_j$ and $i < j$, we set $x_i < x_j$ for definiteness. Note that $\pi_0, ..., \pi_{L-1}$ is a permutation of the numbers $0, 1, ..., L-1$; for this reason, ordinal patterns are sometimes called permutations too. The set of ordinal $L$-patterns will be denoted by $\mathcal{S}_L$.

Topological permutation entropy (otherwise called the capacity of the source $X$), is defined as

$$h^*_\text{top}(X) = \lim_{L \to \infty} h^*_\text{top}(x_0^{L-1}) = - \lim_{L \to \infty} \frac{1}{L} \log N(L),$$

(9)

where $N(L)$ is the number of allowed ordinal $L$-patterns in the ‘messages’ output by $X$. Thus, the estimation of $h^*_\text{top}(X)$ boils down to counting the number of distinct patterns in sliding windows of size $L$. For a wide class of dynamical systems (maps), the topological permutation entropy is equal to the topological entropy [15].

For a map $f : I \to I$ we say that an ordinal $L$-pattern $\pi$ is allowed or admissible for $T$ if there exists $x \in I$ of type $\pi$, otherwise the ordinal pattern is forbidden for $T$. Since $|\{\pi \in \mathcal{S}_L\}| = L!$ and $\lim_{L \to \infty}(\log L!/L) = \infty$, it follows from the result cited above that that orbits of quite general maps have necessarily forbidden patterns [17, 18]. The forbidden ordinal patterns of the shift and signed shift transformations on sequence spaces have been studied in [19] and [20], respectively.

In order to see this, consider sequences generated using the logistic map $x_{n+1} = g(x_n) = 4x_n(1-x_n)$. In Fig. 3 we can see how for this system there are forbidden ordinal patterns, in particular the pattern $(2, 1, 0)$ cannot be observed on a time series of this system. On the other hand, unconstrained random sequences have no forbidden patterns with probability one. This being the case, the existence of forbidden patterns (together with the robustness of ordinal patterns to additive noise) can be exploited to discriminate deterministic noisy time series from white noise (an independent and identically distributed random process), with a remarkable success [7].

4. Topological entropy of one-dimensional CA

Let $F : S^Z \to S^Z$ be the global transition map of an elementary CA. We consider for simplicity here CA where $S = \{0, 1\}$. As mentioned in Sec. 2, its dynamical complexity can be measured by means of the topological entropy (6).

Another possibility consists in using the topological permutation entropy $h^*_\text{top}(F)$ instead, that can be computed as follows.

The topological permutation entropy of the automaton defined by the local rule (2), can be estimated via the ordinal patterns of its global map $F : \{0, 1\}^Z \to \{0, 1\}^Z$ us-

- 534 -
ing two equivalent procedures. The first one would be to estimate the topological permutation entropy of the “sequence” of spatiotemporal data \( \{x^t_i\}_{i=1}^M \), using for this purpose lexicographical order, that is, we say that \( x^t_n < x^t_m \) if \( s_n(i) = s_m(i) \) for \( 1 \leq i \leq j-1 \) and \( s_n(j) < s_m(j) \). An equivalent way to achieve the same goal would be to consider the sequence of real numbers \( \{\phi^+_i\}_{i=1}^M \), that for \( S = \{0, 1\} \) has the form

\[
\phi^+_i = \phi(x^t_i) = \sum_{j=1}^{N} s(j) \in [0, 1).
\]  

(10)

Remarkably, and similarly to what happened with maps, the topological permutation entropy of the CA is equal to its topological entropy [15]. This result provides a first insight on the validity of the permutation complexity tools for the study of spatiotemporal dynamics. We provide now some numerical evidences validating this result.

5. Numerical results

In order to illustrate our ideas we consider the CA with local rule

\[
f(p, q, r) = p + r \mod 2,
\]  

(11)

which is an instance of a positively expansive CA, thus with complicated dynamics. The topological entropy for this CA is \( h_{top}(F) = 2 \log 2 = 2 \) bit/iteration [16].

Figure 3 shows different aspects of the cellular automaton, fixing the size \( N = 250 \). We can see in Fig. 3 (a) the time evolution of cells \( 1 \leq i \leq 250 \), which clearly displays a complex spatiotemporal dynamics. In Fig. 3 (b) we can see a plot of the sequence \( \{\phi^+_i\}_{i=1}^M \) for this CA, it also has a corresponding complex appearance. Notice that simple behaviour on the CA would typically yield to a simple appearance in the corresponding sequence of \( \phi^+_i \). The existence of certain structure in this seemingly complex sequence is revealed in Fig. 3 (c) where the “return map” of the sequence, i.e. the plot of \( \phi^+(x^t_i) \) vs \( \phi^+(x^t_{i+1}) \) is shown. This graph has seemingly a fractal structure; if the sequence \( \{\phi^+_i\}_{i=1}^M \) were purely random one would expect to see a random cloud of points.

Figure 3 (d) illustrates that our result is verified for this CA, as claimed. There we show the convergence of the topological permutation entropy rates of order \( L \) computed using the ideas provided above to the value of \( h_{top}(F) = 2 \) bit/symbol. This is a simple example of how the tools of permutation analysis can be used in order to quantify the complexity of a CA.

6. Conclusion and outlook

In this paper we have provided a simple example on how the tools of permutation analysis can be used to estimate the complexity of a system with complex spatiotemporal dynamics, using as an example a CA. We have shown that for a simple class of CA it is possible to estimate its topological
permutation entropy and that it converges to its topological entropy.
This result illustrates some of the basic features of the application of permutation analysis to spatiotemporal dynamics. In a recent work we have shown how these ideas can be extended to the study of spatiotemporal data of Coupled Map Lattices (CMLs) and even to real spatiotemporal data from magnetoencephalograms (MEGs) [8]. Note that the state of each site of these two type of data are not (in principle) discrete. However, in the examples studied we have notice that it suﬃces to discretize the state of each site using two simbols (for example ﬁxing \( s_i(t) = 0 \) if the state of the site at time \( t \) is below its mean value and \( s_i(t) = 1 \) if it is above its mean value). Numerical evidence show that a permutation analysis of the resulting discretized spatiotemporal data provides a way to quantify the complexity of the systems considered. Furthermore, it can be shown that a combination of the permutation analysis in time, using the above ideas, and in space, considering the state vector as a sequence, allows one to distinguish quite neatly between diﬀerent types of complex spatiotemporal data. All these evidences, together with the computational speed of the permutation complexity methods and its robustness against noise (due to the fact that they rely on inequalities) makes the permutation complexity analysis a promising tool for the analysis of complex spatiotemporal data.

Acknowledgments

The authors would like to thank NOLTA 2010 organizing committee members for their fruitful suggestions and comments. Financial support from the Ministry of Education and Science (Spain) under Project No. FIS2006-08525, and from the Ministry of Science and Innovation (Spain) under Projects No. MTM2009-11820 and No. FIS2009-09898 is acknowledged.

References


Robust and Non-Robust $\omega$-limit Orbits in 1D Cellular Automata

Giovanni E. Pazienza†

†Cellular Sensory and Wave Computing Laboratory, MTA - SZTAKI, Budapest, Hungary
and Pazmany University, Budapest, Hungary
Email: gpazienza@sztaki.hu

Abstract—The two most popular classifications of 1D elementary Cellular Automata rules are based on the dynamics of the so-called robust $\omega$-limit orbits, which can be observed when a long random bit string is used as initial state. In this paper, we introduce a classification that takes into account also the dynamics of the non-robust $\omega$-limit orbits, which can be observed only for very specific initial states.

1. Introduction

Cellular automata (CA) are a classical example of how interdependent elementary structures can generate complex dynamics. In the last few years, this topic has been successfully addressed by using the theory of Nonlinear Dynamics [1], which considers CA as a special case of Cellular Nonlinear Networks (CNNs). One of the main results of such approach has been the introduction of a new classification of CA rules that is based on the notion of $\omega$-limit orbit, which we will introduce later in this paper. Such classification is based on quantitative criteria, and it differs from the popular one proposed by Wolfram [2] which rather considers qualitative criteria. Still, both Chua’s and Wolfram’s classifications are based on the so-called robust $\omega$-limit orbits while they ignore the dynamics of so-called non-robust $\omega$-limit orbits. In this paper, we propose yet another classification of the CA local rules, but in this case based on the properties of both the robust and the non-robust $\omega$-limit orbits.

The paper is structured as follows: in Sec 2, we introduce some fundamental concepts of Cellular Automata; in Sec. 3, we discuss both Chua’s and Wolfram’s classifications; in Secs. 4 and 5, we present several examples concerning the robust and the non-robust $\omega$-limit orbits, respectively; in Sec. 6, we draw the conclusions.

2. Brief notes on Cellular Automata

Cellular Automata consist of regular uniform lattice of cells assuming a finite number of states; here, we consider one-dimensional CA in which cells are arranged in an array of length $L = I + 1$ and can take only two states: 0 and 1. For instance, a bit string $x$ at the generic time step $n$ is

$$x^n = (x^n_0, x^n_1, \ldots, x^n_{I-1}, x^n_I)$$  \hspace{1cm} (1)$$

where the subscript indicates the position of the cell in the array. Hereafter, letters in bold indicate bits strings, and letters in italics are used for the single bits.

Cells are updated synchronously and the time evolution of a bit string can be effectively summarized by the notation

$$x^{n+1} = f(x^n)$$  \hspace{1cm} (2)$$

in which the superscript indicates the iteration. The state of each cell at iteration $n + 1$ depends on the states of its neighbors (here we consider only the nearest neighbors) at iteration $n$:

$$x^n_i = f(x^n_{i-1}, x^n_i, x^n_{i+1})$$  \hspace{1cm} (3)$$

In the following, we use periodic boundary conditions, which means that

$$x^{n+1}_0 = f(x^n_1, x^n_0, x^n_I)$$ and $$x^{n+1}_I = f(x^n_{I-1}, x^n_I, x^n_0)$$  \hspace{1cm} (4)$$

Under the restrictions detailed above, there are only 256 possible functions $f$, called rules, which we can be denoted by $f_{110}$ up to $f_{255}$. For instance, the notation:

$$x^{n+2} = f_{110}(f_{110}(x^n))$$  \hspace{1cm} (5)$$

indicates the application of rule 110 to the bit string $x^n$ two times to obtain the bit string $x^{n+2}$. If the functions $f_i$ are deterministic and the length $L$ of the bit string is finite, then the evolution of an arbitrary initial state under an arbitrary rule $f_i$ will end up in a periodic orbit, in the sense that there exist $p$ and $T$ such that

$$x^p = x^{p+T}$$  \hspace{1cm} (6)$$

Obviously, $x^{p'} = x^{p'+T}$, for all $p' > p$. The bit strings from $x^0$ to $x^{p-1}$ are said to belong to the transient, which has length $p$, while the bit strings from $x^p$ on are said to belong to the periodic orbit, which has length $T$. In this paper, we have opted for the terminology $\omega$-limit orbits borrowed from classical nonlinear dynamical systems [3]. Even though the classical usage of the word orbit allows it to be the entire trajectory (both transient and steady state), the definition of $\omega$-limit orbit requires it to be periodic for finite $L$. For this reason, in our terminology the $\omega$-limit orbit excludes the transient part of the orbit and, for finite $L$, it coincides with the periodic orbit.

For some rules, it may happen that the there exists a $\tau$ for which:

$$x^p = S^\tau(x^{p+\tau})$$  \hspace{1cm} (7)$$
where $S^\sigma$ indicates a shift, left or right, by $\sigma$ positions, where conventionally $\sigma$ is positive for left shifts and negative otherwise. Therefore, an $\omega$-limit orbit can be characterized by its parameters $\tau > 0$ and $|\sigma| \geq 0$.

Since it is unfeasible to analyze the evolution of CA rules starting from all $2^L$ possible initial states when $L$ is big, researchers use long (e.g., $L$ greater than 100) random initial bit strings to characterize the behavior of a rule, as explained in detail in Sec. 3. The $\omega$-limit orbits found via this procedure are said to be robust because they can be observed starting from a generic initial state. Rules can also have some $\omega$-limit orbits, called non-robust, that can be reached only from some very specific initial states.

3. Classification of CA rules by Wolfram and Chua

3.1. Wolfram’s classification

Cellular Automata local rules can be grouped according to many criteria, and several different classifications have been presented so far [4] [5] [6] [7] [8] [9] [10] [11]. Probably, the most famous is due to Wolfram [2], who proposed to classify CA local rules into four classes (here labeled from ‘W1’ to ‘W4’), depending on the evolution of the system from a random initial state:

- W1: evolution leads to a homogeneous state;
- W2: evolution leads to a set of separated simple stable or periodic structures;
- W3: evolution leads to a chaotic pattern;
- W4: evolution leads to complex localized structures, sometimes long-lived.

On the one hand, this classification can be applied to any Cellular Automaton model, regardless the number of states, spatial arrangement, neighborhood etc.; on the other hand, it has received criticism for being based on empirical criteria (e.g., see [12]).

3.2. Chua’s classification

An alternative classification scheme composed by six different groups (labeled from ‘C1’ to ‘C6’) was proposed by Chua in [1]. In this case, the feature used to discriminate the rules is the robust behavior of the $\omega$-limit orbits found by using random bit strings:

- C1: rules exhibiting robust period-1 $\omega$-limit orbits;
- C2: rules exhibiting robust period-2 $\omega$-limit orbits;
- C3: rules exhibiting robust period-3 or period-6 $\omega$-limit orbits;
- C4: rules exhibiting robust $\sigma_\tau$-shift $\omega$-limit orbits, where $\sigma$ and $\tau$ do not depend on the initial bit string or the length $L$;
- C5: bilateral local rules exhibiting a robust $\sigma_\tau$-shift $\omega$-limit orbits, where $\sigma$ and $\tau$ depend on the initial bit string and/or the length $L$;
- C6: non-bilateral local rules exhibiting a robust $\sigma_\tau$-shift $\omega$-limit orbits, where $\sigma$ and $\tau$ depend on the initial bit string and/or the length $L$.

Some examples of spatial-temporal patterns generated by the rules belonging to the different groups are displayed in Table 1.

Table 1: Examples of spatial-temporal patterns obtained from a single black pixel for six different rules and the classifications according to Wolfram (W) and Chua (C), as described in Sec. 3: (a) Rule 0, W1 and C1; (b) Rule 51, W2 and C2; (c) Rule 62, W2 and C3; (d) Rule 170, W2 and C4; (e) Rule 90, W3 and C5; (f) Rule 110, W4 and C6.

3.3. Relationship between Wolfram’s and Chua’s classifications

These two classifications are related to each other: rules belonging to W1 are a proper subset of those of C1; rules belonging to W2 can be in any of the Chua’s groups from C1 to C4; finally, rules belonging to W3 and W4, can be either in C5 or in C6, depending on their characteristics. Therefore, we can summarize these results as follows:

$$W1 \subset C1$$

$$W2 \equiv ((C1 \setminus W1) \cup C2 \cup C3 \cup C4)$$

$$W3 \cup W4 \equiv (C5 \cup C6)$$

We ought to emphasize that Chua’s classification has been expressly developed for 1D elementary CA, and hence it could be different for other models (e.g., more states, different neighborhood), while Wolfram’s classification can be applied without changes independently of the particular
CA model under consideration. Obviously, the presence of non-robust \( \omega \)-limit orbits is peculiar to rules of \( W_1 \) and \( W_2 \), in Wolfram’s classification, and of \( C_1, C_2, C_3, \) and \( C_4 \), in Chua’s classification, since the classes \( W_3, W_4, C_5, \) and \( C_6 \) are characterized by the fact of not having a dominant, i.e., robust, kind of orbit.

4. Results on the robust \( \omega \)-limit orbits

An exhaustive analysis of all CA rules has led to the conclusion that all globally-independent local rules have at most two robust \( \omega \)-limit orbits. In particular, the rules of \( C_5 \) and \( C_6 \) do not have any robust \( \omega \)-limit orbit, by definition; rules 14, 43, 57, 142, 184 (all of them belonging to \( C_4 \)) have two robust \( \omega \)-limit orbits; all other rules have only one robust \( \omega \)-limit orbit. The rules of the first four groups in Chua’s classification can be grouped according to their values of the Bernoulli parameters \( \sigma \) and \( \tau \), as shown in Table 2. In particular, the rules of \( C_1, C_2, \) and \( C_3 \) are in the first column, because they are periodic in time but not in space and hence \( \sigma = 0 \), while the rules of \( C_4 \) (so-called Bernoulli rules) have \( \sigma \neq 0 \) since they are periodic in space and time.

Table 2: Distribution of the rules of the first four classes in Chua’s classification according to the values of \( \sigma \) and \( \tau \) of their robust \( \omega \)-limit orbits. All these rules belong either to \( W_1 \) or to \( W_2 \) in Wolfram’s classification.

| \( \tau \) | \( \sigma = 0 \) | \( |\sigma| = 1 \) | \( |\sigma| = 2 \) |
|---|---|---|---|
| 1 | C1 | C4 | |
| 2 | C2 | C4 | C4 |
| 3 and \( \tau = 6 \) | C3 | |

Observe that no Bernoulli rule has robust \( \omega \)-limit orbits with \( \tau > 2 \). Also, no rule can have robust \( \omega \)-limit orbits with \( |\sigma| > \tau \) because this would imply that the information travels in space faster than one bit per iteration, which is not possible in our nearest-neighbors model. Nevertheless, in the examples displayed in Table 3 (a), (b), and (d) the information apparently travels faster than one bit per iteration, but this effect is due to particular spatial configurations that can occur only in non-robust periodic orbits.

5. Results on the non-robust \( \omega \)-limit orbits

Both Wolfram’s and Chua’s classification are based on the results obtained by using one or more long random bit strings as initial state, but this method allows us observe only the robust \( \omega \)-limit orbits, as explained in Sec. 2. However, as already pointed out in [13, 14] there exist rules whose robust \( \omega \)-limit orbits are ‘dull’, while their non-robust \( \omega \)-limit orbits exhibit a variety of interesting dynamics. For example, Rule 164 is \( W_2 \) and \( C_1 \), and hence they are not considered ‘interesting’ in such classifications. A similar situation happens for Rule 37 which is \( W_2 \) and \( C_2 \). However, both these rules have a non-robust \( \omega \)-limit orbit with \( \tau = 3 \) and \( \sigma = 1 \) for \( L=14 \), as it can be observed in Table 3. Other two examples are given by Rules 9 and 25, both of them \( W_2 \) and \( C_4 \), which have non-robust \( \omega \)-limit orbit with \( \sigma = 2 \) and \( \sigma = 3 \), respectively, and \( \tau = 3 \).

Table 3: Examples of non-robust \( \omega \)-limit orbits with \( \tau = 3 \):
(a) Rule 164, \( W_2 \) and \( C_1 \) \((\sigma = 1)\); (b) Rule 37, \( W_2 \) and \( C_2 \) \((\sigma = 1)\); (c) Rule 9, \( W_2 \) and \( C_4 \) \((\sigma = 2)\); (d) Rule 25, \( W_2 \) and \( C_4 \) \((\sigma = 3)\).

Many of the 70 globally-independent rules belonging to the first two Wolfram’s groups (or, equivalently, to the first four Chua’s group) have only one kind of orbit. For example, all \( \omega \)-limit orbits of Rule 76 are period-1, and all \( \omega \)-limit orbits of Rule 162 have \( \tau = 1 \) and \( \sigma = 1 \). However, this is not always the case: some rules have orbits with both \( \tau = 1 \) and \( \tau = 2 \). Remarkably, there are a few rules – namely, 9, 25, 37, 74, 94, 164 – having non-robust orbits with \( \tau = 3 \). The reason why such a feature is so interesting is that a classical work [15] in Nonlinear Dynamics proofs that in a continuous function defined over an interval, the presence of a period-3 orbit implies the presence of orbits with any other period. Obviously, this result does not apply directly to Cellular Automata because, in general, they describe discontinuous functions. Nevertheless, we noticed that all rules in the last two groups of Chua’s classification have at least one periodic orbit with \( \tau = 3 \) and, at the same time, the five rules with \( \tau = 3 \) mentioned above tend to have a more complex behavior than the remaining of the first four groups of Chua’s classification. For instance, in Table 4 orbits with \( \tau > 3 \) for two of the five rules aforementioned are displayed. Therefore, somehow the parameter \( \tau = 3 \) may give an indication of richer dynamics, as also confirmed by the analysis made by using other methodologies [16]. For this reason, besides the classification of rules according to their robust orbits, as done by Wolfram and Chua, we suggest a classification based on their non-robust or-
Table 4: Examples of non-robust $\omega$-limit orbits with $\tau$ greater than 3: (a) Rule 9, with $\tau = 5$ and $\sigma = 0$; (b) Rule 164, with $\tau = 7$ and $\sigma = 7$.

bits. In particular, we can distinguish three classes: i) rules with only robust $\omega$-limit orbits; ii) rules whose non-robust orbits have $\tau = 1$ and/or $\tau = 2$; iii) rules whose non-robust orbits have $\tau \geq 3$. This last group contains at least one rule from each of the first four Chua’s groups.

6. Conclusion

In this paper, we propose a classification of the CA local rules based on the properties of both robust and non-robust $\omega$-limit orbits. Thanks to this new approach, we found that some rules belonging to C1, C2, C3, and C4 – or, equivalently, to W1 and W2 – have at least some $\omega$-limit orbits with characteristics similar to those of the rules belonging to C5 and C6 – or, equivalently, to W3 and W4. This result opens a new scenario to analyze the computational properties of Cellular Automata: some rules that have been considered “uninteresting” so far become suddenly rich in dynamic behaviors, still to be studied.

References

Uncertainty Profiles for Predicting Complex Nonlinear Dynamics in Cellular Automata: The Case of Five Cells Neighborhood

Radu Dogaru¹ and Ioana Dogaru¹
†Dept. of Applied Electronics and Information Engineering, “Politehnica” University of Bucharest
Bvd. Iuliu Maniu 1-3, Bucharest, Romania
Email: radu_d@ieee.org, ioana_d70@yahoo.com

Abstract— Uncertainty profiles are behavioral indexes that can be easily determined for a given CA cell structure and neighborhood, as a nonlinear relationship between the probability of a cell to be in state "1" when the probabilities of the neighbor cells in the previous iterations are known. They allow to predict the nature of CA dynamics without the need to effectively simulate the dynamics. This paper reviews the method of uncertainty profile as a method for “designing for emergence” and exemplifies it first time for CAs with 5 cells neighborhoods. The predictive power of this method is demonstrated in conjunction with other recent methods for nonlinear analysis and classification of CA dynamics.

1. Introduction

Cellular automata (CA) are a subclass of nonlinear dynamic systems endowed with certain characteristics that makes them attractive as naturally inspired, massively parallel computing architectures. Their applications range from modeling to cryptography and signal processing. Unlike traditional computers, where sets of instructions are used to build algorithms tailored to certain applications, in “programming” the CA the goal is to establish the values of a relatively small set of parameters called a cell’s gene [1] such that a certain kind of dynamic behavior, with applicable potential, will emerge in an array of identical cells defined by the unique gene. Such dynamics is often called emergent although a precise definition of emergence is lacking for the moment. In the case of Boolean CA the gene is a vector of binary elements \( \mathbf{Y} = [y_{x_{-1}}, y_{x_{-2}}, \ldots, y_0] \) defining the local Boolean function of a cell with \( n = \log_2(N) \) inputs. The decimal representation of the above string will be called an ID. It is assumed that the desired dynamics is often vaguely specified (e.g. “chaotic behavior” or “must have a Class IV behavior in the sense of Wolfram”, etc.). Such problem is rather general and applies for any kind of nonlinear dynamic system. Although for some dynamic behaviors (typically the equilibrium behaviors) analytic approaches are possible, such as those given by various stability theorems, the “design for emergence” is still an open issue. In some sense, defining the “programming” of nonlinear dynamic systems to perform useful functions may have the same relevance as the defining of various programming techniques which sprung the explosion of the information technology using “classic” computational structures such as the microprocessor. Advanced nano and molecular technologies are ideal for the realization of CA computational mediums that will need adequate “programming” techniques to spread a new wave of computing applications. Only recently, starting with pioneering works in the area of local activity [2][3] the importance of such design for emergence techniques was recognized. A review of the such techniques is given in [4]. In the same work the uncertainty profile method (UPM) was first exposed. It allows to establish a mathematical relationship between the gene space and a behavior space without a need to simulate the dynamical system. Its effectiveness in predicting various emergent behaviors was demonstrated for elementary CA [5][9]. Certain novel relevant dynamic behaviors with interesting applications, such as binary synchronization of chaos [6][7][8] were also put in a direct relationship to the UPM method. In [9] we investigated the relationship between our UPM method and a set of nonlinear dynamic methods for classifying elementary CA. They are assigned precisely into 6 classes as described in [10][11] from a series of monumental works dedicated to a nonlinear dynamics approach to CA. It turned out that UPM may predict quite well if a CA with a given gene is in one ore another of the 6 categories although its probabilistic nature gives also some little misclassification errors [9]. In this paper we extend the application of this method to one-dimensional CA with 5 cells neighborhoods. In Section 2 we will briefly review main concepts in designing for emergence when applied to cellular automata. Section 3 gives the definition of the uncertainty profile and formulate for its calculation in the case of our interest. Although the search space in this case is rather big (there are \( 2^{32} \), i.e. about 4 billion possible genes) in Section 4 we demonstrate how easily one can pick genes leading to desired dynamic behaviors from the 6 classes, while UPM defines an inverse relationship between the desired behavior space and the solution gene space.

2. Cellular Automata and Design for Emergence

The nonlinear dynamic systems investigated herein are CA with \( M \) cells given by the following discrete time state equation which applies synchronously to all cells (a cell is identified by a spatial index \( i = 1, M \) ) and with a specified initial condition for all CA cells:

\[
x_i(t+1) = \text{Cell}(x_{i-2}(t), x_{i-1}(t), x_i(t), x_{i+1}(t), x_{i+2}(t)) \text{ID} \quad (1)
\]
where \( y_j = Cell(u_1, u_2, u_3, u_4, u_5, ID) \) is the local rule or transition function of a cell given a neighborhood of \( n=5 \) cells. A periodic boundary condition is assumed, such that when \( i=1 \) (leftmost cell) its neighbors to the left are given by indexes \( i=M, M-1 \). The local function is completely specified by the gene vector mentioned in Introduction, or its associated ID. In other words, the \( j \) index in the above is the decimal representation of the binary word \( [u_5, u_4, u_3, u_2, u_1] \).

In [10][11] a precise classification and some expected behavioral properties of CA was given for all 88 basic elementary CA rules. Briefly speaking, the main feature to discriminate among classes was the length (period) of the most likely attractor (given a random initial state) and its dependency on the number \( M \) of CA cells. Classes 1, 2, 3 (with 26, 13 and 1 members) are similar in that for all of them there is a constant \((1, 2, 3)\) period of the main attractor. Class 4, (or Bernoulli-shift), with 30 members, includes all CA rules leading, as for the previous ones, to a predictable period (given the gene and without simulating the CA) that depends linearly by the number of cells \( M \). Finally, classes 5 and 6 (with 10 and 8 members respectively) include the most complex CA rules, where a prediction of the attractor length and its dependence on \( M \) is not possible without effectively running the CA. Rules in Class 5 are symmetric (bilateral) while those in Class 6 are not. Quite notably, CA rules found so far by other researchers as possessing interesting properties fall in Chua’s Class 6.

As seen above, while various behavioral descriptors (e.g. the attractor length in the above, or Lyapunov exponents, or various entropies etc.) may be defined and associated with a behavioral space \( B \), its association with the gene space \( G \) can be usually done only via an algorithm to simulate the nonlinear dynamics and compute the descriptors. Thus, associating a point \( g_i \) from the gene space to a behavior point \( b_i = A(g_i) \) in \( B \) is straightforward. Such an approach makes extremely difficult to design for emergence (i.e. solve the inverse problem of finding \( g_i \) when \( b_i \) is given). The difficulty stands in the lack of mathematical instruments to show the existence and to compute the inverse \( A^{-1} \) of the algorithm simulating the nonlinear dynamics. Yet, as seen in the case of stability theory, local activity theory, or as applied in the case of Classes 1-4 above, using nonlinear theory tools makes possible to directly locate genes for given behaviors. Still, remains of interest to locate genes associated with complex behaviors such as those in Classes 5-6 that may be also subdivided into even finer categories (e.g. among chaotic behaviors one may identify synchronizable behaviors etc.). To achieve this goal we introduce next uncertainty profile as a vector of behavioral indexes. Associations between behaviors described previously in various classes and concrete values of uncertainty profiles were established in [9] and may be used to predict the behavior for arbitrary neighborhoods and sizes. In the next we will exemplify for the case of \( n=5 \) cells in a neighborhood (or, using a taxonomy introduced in [4] CA belonging to ‘1a5’ family).

3. Uncertainty Profiles as Behavioral Descriptors

Uncertainty profiles are behavioral indexes that can be easily determined for a given CA cell structure and neighborhood. The idea is to consider that all initial state \((i=0)\) cells are in a quiescent state \( q \in [0,1] \) with probability \( p_q = 1 \), except a group of \( n \) neighbor cells assigned state 1 with probability 0.5 or maximum uncertainty \( u_1 = 1 \). In our theory the uncertainty of a cell “\( k \)” is computed as \( u_k = 1 - |2p_q - 1| \) and consequently it is 0 in either cases when a cell is an “sure” state (0 or 1). As seen in Fig.1 the effective value of \( u_k \) is less important than its spatial spread after one CA iteration \((i=1)\), which relates to various types of CA behaviors as shown in [4].

For a given local Boolean function and probability of its inputs, it is possible to compute the output probability using an information theoretic approach. Details are given in [4], resulting that the output probability is a degree \( n \) polynomial with respect to the input probabilities and with \( N = 2^n \) coefficients given by weighted summations of the gene’s \( Y \) bits. When the neighborhood arrangement is given, as shown in Fig.1, the polynomial output probability formulae allows to compute the uncertainty for all \( 2n-I=9 \) positions that are affected in the next state, i.e. resulting in an uncertainty vector \( U = [U_1, U_2, ..., U_N] \).

Without entering into details (explained elsewhere [12]) it turns out that for input uncertainties equal to 1 in the \( n=5 \) additional cells (initial state) two representative probability matrices \( R_0 \) and \( R_1 \) can be precisely defined, each having \( 2n-I \) lines and \( N = 2^n \) columns. \( R_0 \) is given below while \( R_1 \) is its mirrored version (last column of \( R_0 \) becomes the first of \( R_1 \), etc.):
Consequently the relationship connecting the gene space (represented by vectors $Y$ or their associated ID) with the behavior space (represented by $U$ vectors) is given by the next equations:

$$P_0 = R_s Y \text{ and } P_1 = R_s Y$$

$$U_0 = 1 - 2P_0 + I \text{ and } U_1 = 1 - 2P_1 + I$$

$$U = 0.5[(2 - y_s - y_n, y_n)U_0 + (y_s + y_n, y_n)U_1]$$

The above relationships hold for any neighborhood provided that their corresponding $R$ matrices are predetermined. It is clear that an inverse relationship between a given behavior and the resulting gene can now be established. A specialized software described in [12] allows the user to change the $Y$ bits looking for the $U$ vector to change so that it corresponds to a desired behavior. Moreover, conditions for the existence of a certain behavior may be easily translated into conditions imposed to the gene bits, as shown in an example in [4] for the case of semi-totalistic CA. In [9] three types of dynamic behaviors (i.e. imploding or I, exploding or E, and preserving or P) were associated to a condition imposed to the uncertainty vector $U$. They may be summarized as follows:

**E:** A profile is “exploding” (i.e. uncertainty spreads within the array or in other words local connectivity gives rise to global computation like random number generation or computation with gliders etc.) if there are at least two members (elements) of the profile with maximum value($U_s = U_i = 1$) and they are distant at more than $n$ cells (i.e. $|k - l| \geq n$). Also, a profile is “exploding” if the above condition is not fulfilled but if the sum of uncertainties is larger than $n$.

**P:** A preserving profile is a non-exploding one with at least $n$ non-zero elements. Preserving behaviors imply that a computation takes place but dominated by the local connectivity (less complex, i.e. filtering).

**I:** An imploding profile is a non-exploding one but with less than $n$ non-zero elements. Such profiles are associated with a dynamics of the CA such that after a few (usually less than $n$) iterations, all cells are in the same sure state (all in 1 or all in 0 or oscillating). They correspond to period 1 and 2 behaviors.

In addition, one can analyze the $U$ vector from the symmetry point of view. A gene may be declared as $S$ (symmetric, when left side elements of $U$ are equal to their right side elements – with $U_s$ in the middle) or $A$ (asymmetric) and in this later case a degree of symmetry may be also computed [9]. It is noted that symmetric behaviors are less complex than asymmetric ones. For the 256 elementary CA a comparison between the classification in [10][11] and the one induced by the $U$ profile gave a relatively consistent match following the rules given in the next table:

<table>
<thead>
<tr>
<th>Class [10]:</th>
<th>1 and 2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>U-based behaviors</td>
<td>I</td>
<td>PS</td>
<td>PA</td>
<td>ES</td>
<td>EA</td>
</tr>
</tbody>
</table>

In addition to Chua’s classification, the UPT method reveals finer subdivisions within each behavioral domains and relationships to other interesting dynamic behaviors. For instance, most asymmetric (A) rules with “exploding” and “preserving” $U$ profiles do have attractors that synchronize binary (the state of only one cell per iteration is sent to the receiver). Particularly interesting are those in the “exploding” case since they are usually chaotic. It was also found that binary synchronization is unlikely for symmetric (S) profiles. Also gliders are favored by a sum of uncertainties in the $U$ profile that is close to $n$.

4. Selection of Desired Behaviors for the “1a5” CA

Let us consider the following problem: find a gene $Y$ (or its associated ID) such that its corresponding “1a5” CA behavior is “exploding” and “asymmetric” (i.e. one that is the most complex).

A first solution is given next using a software that evaluates equations (2) (3) (4) interactively. We know that we are looking for a vector $U$ which is asymmetric (A) and fulfills the condition $E$. One simple way to look for a gene is to enter an arbitrary decimal ID i.e. 22...2 until it is seen that the corresponding $U$ profile fulfills the desired conditions. Such a situation is presented in Fig.2 from the panel of the software. The resulting profile is $U= [088141514168161]_32$ with a sum of uncertainties equal to 6.1875 (i.e. larger than 5.5). It is also clear that it is an asymmetric profile.

![Fig.2. A particularly computed uncertainty profile for a given ID= 222222222.](image)

To test if the desired behavior is indeed in that class, one may employ the key “simulate” producing the evolution shown in Fig. 3 (time is on the vertical axis):

![Fig.3. Dynamic evolution of CA with ID=2222222222 exhibiting a complex behavior (interacting gliders)](image)

The same software allows to test for the binary synchronization property in a CA with the same ID and a given number of cells (time is now on the horizontal axis).
Three traces (upper is the Tx CA, middle the Rx CA and lower trace the difference of states between Tx and Rx) show that indeed such a phenomena occurs as predicted and expected for EA behaviors (Fig. 4).

Another solution to pick genes with desired behaviors consists in selecting a random pool of IDs (here 10,000) and use the UPT method to calculate the U profile for each of them. Note the very good speed of our method, where all profiles for 10,000 IDs were calculated in less than one second on a personal computer. A behavior space may be drawn as shown in Fig. 5 after selecting two important parameters (cumulated uncertainty, i.e. the sum of all elements in U, and a symmetry index – detailed in [9]). Each point is assigned one of the behavioral classes as discussed before.

In addition to these classes it was observed that the most “interesting” behaviors hold for a low (but non zero) asymmetry index and in the edge between P and E behaviors. Since for each point in the behavioral space a list of IDs is given, looking for a desired behavior in such sub-regions (e.g. the black dots in Fig. 5. located on the lower edge of the behavioral triangle domain) allow to rapidly locate IDs leading to desired CA dynamics. Simulations of such genes confirm the prediction. It is interesting to answer how many rules are assigned to each behavioral category. Using 10000 randomly selected IDs the conclusion is given in the next table, where the results for elementary CA (or “1a3” according to taxonomy in [4]) from [9] are given for comparison.

<table>
<thead>
<tr>
<th>Behavioral class</th>
<th>I</th>
<th>PS</th>
<th>PA</th>
<th>ES</th>
<th>EA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1a5 CA</td>
<td>0.1%</td>
<td>0.3%</td>
<td>17.3%</td>
<td>1.6%</td>
<td>80.7%</td>
</tr>
<tr>
<td>1a3 CA</td>
<td>9.1%</td>
<td>20.4%</td>
<td>39.7%</td>
<td>15.9%</td>
<td>14.9%</td>
</tr>
</tbody>
</table>

Concluding, the UPT method may be conveniently applied to “1a5” type CA to identify desired behaviors. From the above table is clear that now most (more than 80%) of the CA cells are associated with complex dynamics instead of only 15% in the case of elementary (“1a3") CA.

Acknowledgement: This work was supported by the research grant ADDBIOSNAR (No. 12079/2008, with CNMP, Romania).

References

Border Collision Bifurcations in a Simple Switching Circuit

Danièle Fournier-Prunaret†, Laura Gardini‡ and Pascal Charge†

†LATTIS-INSA, LAAS-CNRS, Université de Toulouse
Toulouse, France
‡Department of Economics, University Carlo Bo
Urbino, Italy
Email: pascal.charge@insa-toulouse.fr, danielle.fournier@insa-toulouse.fr, gardini@econ.uniurb.it

Abstract—A two-dimensional piecewise smooth continuous model describing a circuit proposed as chaos generator is analyzed. The parameter space is investigated in order to classify regions of existence of stable cycles, and regions associated with chaotic behaviors. Border collision bifurcation curves and degenerate flip bifurcation curves are analytically detected. Moreover, the homoclinic bifurcations occurring in cyclical chaotic regions leading to chaos in one-piece are also detected.

Keywords: Two-dimensional chaos generator, switched dynamical systems, piecewise map.

1. Introduction

Chaotic signals appear to be interesting signals for many applications, particularly in telecommunications and transmissions. For some kind of applications, it is necessary to consider robust chaos [1], which can endure, even if parameters values are slightly changed. A way to obtain robust chaotic signals is to consider systems where border collision bifurcations appear [2][3][7][8]. We have previously proposed a chaos generator, which is a switching circuit where a latch is used to modify the behaviour of an analogical circuit, where such kind of bifurcations occur [4][5][6]. In this paper, we continue to analyze our circuit by considering border collision bifurcations and degenerate flip bifurcations and to put in evidence robust chaos. Our paper is organised as follows: in section 2, we recall the circuit and its modeling; in section 3, we analyze the bifurcations and the route to chaos in a peculiar case.

2. Description of the circuit

The circuit is shown in Figure 1 and its model has been introduced in a more detailed way in [4][5][6]. Just recall that the state variables of the system are the two voltage capacitor $v_x(t)$ and $v_y(t)$. At every clock period $T$, the flip-flop is set and then the switches position is ‘1’. When one of the capacitance voltages reaches the reference value $V_{ref}$, the two switches are turned toward their position ‘0’. So, according to the switches position, the two capacitors are simultaneously charging or discharging. Thus, using classical models of circuits, we easily obtain the equations of the system. We consider the case when $V_x = V_y$. Let us recall the parameter normalization:

$$\alpha = \frac{V_x}{V_{ref}} > 1, \quad \mu = \frac{R_y C_y}{R_x C_x} > 0, \quad \delta = e^{-\pi e^{V_{ref}} C_x} < 1$$

The normalized state variables are given by:

$$x_n = \frac{v_x(nT)}{V_{ref}} \in [0, 1], \quad y_n = \frac{v_y(nT)}{V_{ref}} \in [0, 1]$$

The following switching curves in $Q = [0, 1] \times [0, 1]$:

$$x = x_b, \quad y = y_b, \quad \Delta(x, y) = \alpha - \alpha y - \frac{\alpha - 1}{\alpha}$$

assuming $x_b \geq 0$ and $y_b \geq 0$, which occurs for $\delta \leq \delta < 1$, where $\delta = \max \left\{ \frac{\alpha - 1}{\alpha}, \frac{\alpha - 1}{\alpha^2} \right\}$

define three different domains in $Q$ (see Figure 2):

$$D_1 = \{(x, y) | 0 \leq x \leq x_b \text{ and } 0 \leq y \leq y_b\}$$
$$D_2 = \{(x, y) | x_b \leq x \leq 1 \text{ and } \Delta(x, y) \geq 0\}$$
$$D_3 = \{(x, y) | y_b \leq y \leq 1 \text{ and } \Delta(x, y) \leq 0\}$$

Figure 1: The circuit
in which the system is defined by different functions. In fact, the circuit is modeled by the map \((x_{n+1}, y_{n+1}) = T(x_n, y_n)\) as follows:

if \((x_n, y_n) \in D_1:\n\)

\[
T(x_n, y_n) = T_1(x_n, y_n) = \begin{cases} 
\alpha + (x_n - \alpha)\delta \\
\alpha + (y_n - \alpha)\delta^{1/\mu}
\end{cases}
\]

if \((x_n, y_n) \in D_2:\n\)

\[
T(x_n, y_n) = T_2(x_n, y_n) = \begin{cases} 
\alpha + x_n\delta \\
(\alpha \frac{(\delta - 1)}{\alpha - 1})^{1/\mu} - \alpha + y_n\delta^{1/\mu}
\end{cases}
\]

if \((x_n, y_n) \in D_3:\n\)

\[
T(x_n, y_n) = T_3(x_n, y_n) = \begin{cases} 
\alpha + x_n\delta \\
(\alpha \frac{(\delta - 1)}{\alpha - 1})^{1/\mu}
\end{cases}
\]

(6)

It is easy to see that the map is well defined as \(T\) is continuous and maps the square \(Q\) (the phase space of interest) into itself.

**Figure 2:** Phase space \(Q\) and three different regions \(D_i\).

**3. Analysis of bifurcations and route to chaos**

In this section we analyze the bifurcations occurring in the circuit modeled by \(T\) given in (6). The model is described by a continuous piecewise smooth map which depends on three parameters \(\alpha, \mu, \delta\), under the constraints given in (1) and (4). The function \(T_1\) is affine and its fixed point, say \(X^*_1 = (x^*_1, y^*_1) = (\alpha, \alpha)\), is outside the square \(Q\) (as \(\alpha > 1\)) and thus it is a so-called virtual fixed point [2]. Moreover both eigenvalues of \(T_1\) are positive and less than 1, so the virtual fixed point is a stable node. This implies that initial conditions inside the region \(D_1\) are mapped towards the virtual attractor and are forced to enter in a different region, \(D_2\) or \(D_3\), from which the iterated points are kept inside \(Q\).

**Figure 3:** Bifurcation diagram in the parameter plane \((\mu, \delta)\) at \(\alpha = 1.1\). Colored regions denote the existence of stable \(k\)-cycles. The white region corresponds to the existence of robust chaos.

**Figure 4:** Some bifurcation curves obtained from Proposition 2 and corresponding to limit of periodicity regions in Figure 3.

**Proposition 1.** The two cases \(\mu > 1\) and \(\mu < 1\) are topologically conjugated.

The proof follows immediately due to the following
property: 

\[ T_1(x_n, y_n, \delta, \mu) = T_1(y_n, x_n, \delta^{1/\mu}, 1/\mu) \]  
\[ T_2(x_n, y_n, \delta, \mu) = T_3(y_n, x_n, \delta^{1/\mu}, 1/\mu) \]  

(7)

The case \( \mu > 1 \) has been extensively studied in [6]. In this paper, we present some results for the case \( \mu < 1 \). As it has been proved in [6] for \( \mu > 1 \), by using the following properties of \( T \) when \( \mu < 1 \):

- The dynamics of \( T \) are described by interactions between \( T_1 \) and \( T_3 \).
- \( T_1 \) is an affine map.
- \( T_3 \) is a triangular map.
- we can do a change of variables in order to get a piecewise 2-dimensional smooth map, for which we know very well the dynamics and the bifurcations; it is then possible by reversing the variables to get the bifurcation curves for our system,

we obtain the following proposition:

**Proposition 2.**

1. Let \( \mu < 1 \) and \( \alpha > \alpha^* = 1 + \delta^{3/\mu} \). Then the fixed point \( X_5^k \) of \( T_3 \) is globally attracting in the state space \( Q \). At \( \alpha = \alpha^* \), a degenerate flip bifurcation occurs and an arc of invariant curve in the region \( D_3 \) is filled with stable 2-cycles. (cf. Figure 5)

Let now \( 1 < \alpha < \alpha^* = 1 + \delta^{1/\mu} \). Then:

2. The stable 2-cycle of \( T \) undergoes a degenerate flip bifurcation, at the bifurcation curve given by:

\[ \text{DFB}_2 : \quad \alpha = 1 + \delta^{2/\mu} \]  

(8)

which may lead to \( m \)-cyclical chaotic sets of any even period \( m \), which undergo bifurcations, merging in pair, up to a one-piece chaotic set.

3. For any \( k \geq 3 \) pairs of \( k \)-cycles, one of which may be locally stable and one unstable, appear via border collision bifurcation crossing the bifurcation curve \( \text{BCB}_k \) given by:

\[ \text{BCB}_k : \quad \alpha = 1 + \frac{(1 - \delta)\delta^{(k-1)/\mu}}{1 - \delta^{(k-1)/\mu}} \]  

(9)

which are maximal cycles, the stable one has one periodic point in \( D_3 \) and \((k-1)\) points in \( D_1 \); the unstable one has two periodic points in \( D_3 \) and \((k-2)\) points in \( D_1 \).

4. For any \( k \geq 3 \) the stable \( k \)-cycle undergoes a degenerate flip bifurcation at the bifurcation curve given by:

\[ \text{DFB}_k : \quad \alpha = 1 + \delta^{k/\mu} \]  

(10)

so that the stability region of the \( k \)-cycle (colored regions in Figure 3) is given by \((\alpha, \delta) \in \Pi_k = (\alpha, \delta) \mid 1 + \delta^{k/\mu} < \alpha < 1 + \frac{(1 - \delta)\delta^{(k-1)/\mu}}{1 - \delta^{(k-1)/\mu}}, \delta < \delta < 1 \)  

(11)

5. Crossing the degenerate flip bifurcation \( \text{DFB}_k \) there is the appearance of \( 2k \)-cyclical chaotic sets, which merge into \( k \)-cyclical chaotic sets at the homoclinic bifurcation occurring at the bifurcation curve \( H_k \) given by:

\[ H_k : \quad \alpha - \delta^{2k/\mu} = 0 \]  

(12)

which in turn merge into one piece chaotic set at the homoclinic bifurcation occurring at the bifurcation curve \( H'_k \) given by:

\[ H'_k : \quad \alpha - \frac{\delta^{k/\mu}}{(\alpha - 1)} = 0. \]  

(13)

In fact the dynamic behaviors of the 2-dimensional map obtained after the change of variables is related to the dynamics of a one-dimensional skew tent map ([10] and references therein) which are completely known. This is the reason why we can obtain all the bifurcation curves analytically.

Results of Proposition 2 are illustrated in Figure 6 and Figure 7 via bifurcation diagrams. We can see the degenerate flip bifurcations for fixed point \( X_5^k \), 2 and 3-cycles, border collision bifurcations for 3-cycles and homoclinic bifurcations for 6-pieces and 3-pieces chaotic attractors. A robust chaotic attractor is shown on Figure 8.

![Figure 5: Degenerate flip bifurcation of fixed point \( X_5^3 \). Infinitely many 2-cycles exist on an invariant arc in \( D_3 \), the stable set of which is the horizontal line through the cycles.](image-url)
References


Analysis of Spike-Trains from Simple Resonate-and-Fire Chaotic Circuit

Satoshi Imai † and Toshimichi Saito †

†EE Dept., Hosei Univ., Tokyo 184-0002, Japan; tsaito AT hosei.ac.jp

Abstract—This paper studies characteristics of inter-spike intervals of simple chaotic spiking neurons having piecewise constant vector field. The circuit repeats vibrate-and-fire dynamics and can output various spike-trains. Using the piecewise exact solution, we can derive the return map and can analyze the dynamics precisely. Using the histogram and recurrence plot of the spike-trains, we classify the characteristics for some key parameters.

1. Introduction

Integrate-and-fire model (IFM) is a simple spiking neuron model. The IFM repeats integrate-and-fire dynamics and can output various spike-trains. The IFM can exhibit synchronous and bifurcation phenomena [1]-[4]. The spiking neuron can be a building-block of pulse-coupled neural networks having rich applications including image processing [5]-[8]. Analysis of spiking neuron models is important to develop bifurcation theory and engineering applications. This paper studies the resonate-and-fire circuit (RFC): our original spiking neuron model having piecewise constant characteristics [4]. Below the threshold, the state variable vibrates divergently around the origin and draws a rect-spiral. If the state variable reaches the threshold, it is reset to the base and the circuit outputs a spike. Repeating the vibrate-and-fire dynamics, the circuit can output various spike-trains. Using the piecewise exact solution, we can derive the return map that enables us to analyze chaos and rich bifurcation phenomena precisely [4]. In this paper, we especially consider the characteristics of spike-trains using three methods: histogram of inter-spike-interval (ISI), recurrence plot of the return map orbit (RP1) and RP of the ISI sequence (RP2). The RP is a visualization method of the time series signals [9]-[10]. Applying them to typical chaotic spike-trains, we can suggest the following:

(1) Using the histogram, we have classified the ISI characteristics into some groups based on wide-band spectrum, line spectrum and their mixture.

(2) Using the RP1 and RP2, we have classified the complex spike-trains and simple spike-trains.

(3) If the burst spike-trains are dominant, the image of the RP1 is different from that of RP2, otherwise the RP1 and RP2 can provide similar images.

2. Resonate-and-Fire Circuit

Fig. 1 shows the RFC. Below the threshold $V_T$, the dynamics is described by

$$C_1 \frac{dv_1}{dt} = I_2 \text{sgn}(v_1 + v_2) \quad \text{for} \: v_1(t) < V_T$$

$$C_2 \frac{dv_2}{dt} = I_1 \text{sgn}(-v_1) \quad \text{(1)}$$

$$\text{sgn}(x) = \begin{cases} 1 & \text{for} \: x > 0 \\ -1 & \text{for} \: x < 0 \end{cases} \quad \text{(2)}$$

As shown in Fig. 2, if $v_1$ reaches $V_T$, the switch $S$ is closed and $v_1$ is reset to the base $E_a$ instantaneously holding $v_2 = \text{constant}$.

$$[v_1(t^*), v_2(t^*)]^T = [E_a, v_2(t)]^T \quad \text{for} \: v_1(t) = V_T \quad \text{(3)}$$

Using the dimensionless variables and parameters

$$x = \frac{v_1}{aV_T}, \quad y = \frac{v_2}{V_T}, \quad \tau = \frac{I_1}{C_2 V_T}, \quad a = \frac{C_2 I_2}{C_1 I_1}, \quad q = \frac{E_a}{V_T} \quad \text{(4)}$$

eqs. (1) and (3) are transformed into

$$\frac{dx}{d\tau} = \text{sgn}(y + ax) \quad \text{for} \: x(\tau) < 1 \quad \text{(5)}$$

$$\frac{dy}{d\tau} = \text{sgn}(-x) \quad \text{(6)}$$

$$[x(t^*), y(t^*)]^T = [q, y(t)]^T \quad \text{for} \: x(\tau) = 1$$

![Figure 1: Resonate-and-Fire Circuit](image)

![Figure 2: Dynamics of the resonate-and-fire neuron](image)
This system has piecewise constant vector field and characterized by two parameters: damping \( a \) and base \( q \). Fig. 2 illustrates a typical waveform. \( x \) vibrates divergently below the threshold \( x = 1 \). If \( x \) reaches threshold \( x = 1 \), \( x \) is reset to the base \( q \). Repeating vibrate-and-fire dynamics, the RFC outputs the spike-trains.

Let \( \tau_n \) be the \( n \)-th spiking position and let \( \Delta \tau_n = \tau_n - \tau_{n-1} \) be the \( n \)-th ISI. Fig. 3 shows typical trajectories. As parameters vary, the RFC can exhibit various spike-trains and typical examples of the ISI histograms are shown in Fig. 4. In Fig. 4(a), the histogram has wide band spectrum. In Fig. 4(b) and (c), the histogram has narrow band spectrum. In Fig. 4(d), the histogram consists of line and continuous spectrums. The RFC can output various spike-trains, however, the histogram-based analysis is not sufficient to classify the ISI characteristic. Hence, we consider the ISI by both histogram and RP in Sec. 4.

3. Return Map

Here, we derive the return map. Fig. 5 shows the key objects. Let \( L_q = \{(x,y) | x = q\} \) be the domain of the return map. We consider the trajectory started from point \( y_0 \) on \( L_q \). If \( x \) reaches the threshold \( x = 1 \), \( x \) jumps the base \( q \). Then trajectory returns to \( L_q \) and let \( y_1 \) be the return point. Since \( y_1 \) is determined by \( y_0 \), we can define the return map \( f \) from \( L_q \) to itself:

\[
y_{n+1} = f(y_n)
\]  

(7)

Fig. 6 shows typical return maps corresponding to Fig. 3. In the figure, invariant interval \( I \) is shown: an orbit eventually enters into \( I \). In Fig. 6(a), the map has infinite branches in \( I \) and exhibits complex chaotic behavior. In Fig. 6(b), the invariant interval \( I \) exists near the fixed point. As \( q \) increases, the graph moves upward and size of \( I \) increases. In Fig. 6(c), we have confirmed that the chaotic orbit can move in the thin period-2 islands.

Figure 5: Definition of the return map

Figure 6: Return map for \( a = 0.2 \) ((a) chaos for \( q = 0 \), (b) island for \( q = 0.48 \), (c) island for \( q = 0.65 \), (d) chaos for \( q = 0.8 \))
4. Recurrence Plot

The RP is known as an analyzing method of chaotic dynamics. This method transforms the time series data to two-dimensional graphics. Using the RP, we can visualize time correlation, periodic and chaotic behavior, stationarity and nonstationarity. Using the RP, we consider the classification of the spike-trains of the RFC.

Let us consider the RP for time series data $v(t)$ ($t = 1, 2, \ldots, N$). Let $S$ be a two-dimensional plane of $N \times N$. Calculate the distance $D(i, j)$ between $i$-th data $v(i)$ and $j$-th data $v(j)$:

$$D(i, j) = |v(i) - v(j)|$$

If $D(i, j) < \theta$, we plot the $(i, j)$ cell of $S$, where $\theta$ is threshold. Repeating this process for all $D(i, j)$ ($i = 1, 2, \ldots, N; j = 1, 2, \ldots, N$), we can make the RP.

We construct the RP for two kinds of data: return map orbits $\{y_0, y_1, \ldots, y_{N-1}\}$ and ISI sequence $\{\Delta \tau_1, \Delta \tau_2, \ldots, \Delta \tau_N\}$. Let $D_y$ and $D_{ISI}$ be the distance calculated for the $y$ data and the ISI data, respectively. Let $\theta_y$ and $\theta_{ISI}$ be the threshold for RP of the $y$ data and the ISI data, respectively. We plot the $(i, j)$ picture cell, as the following:

For $y$ data

$$D_y(i, j) = |y_i - y_j| < \theta_y$$

For ISI data

$$D_{ISI}(i, j) = |\Delta \tau_i - \Delta \tau_j| < \theta_{ISI}$$

The RPs for $y$ data and ISI data are abbreviated by RP1 and RP2, respectively. Figs. 7 and 8 show the RP1 and RP2 corresponding to the Fig. 3. Figs. 7(a) and 8(a) are complex image, because the RFC outputs complex spike-trains. The $y$ and the ISI data have wide band spectrum. For $q = 0.48$ and 0.65, the chaotic attractors are islands and are similar to periodic attractor, hence the RP becomes monotone as shown in Figs. 7(b), 7(c), 8(b) and 8(c). The RP for period-2-like islands (Figs. 7(c) and 8(c)) has lighter tone than that for period-1-like islands (Figs. 7(b) and 8(b)). Figs. 7(d) and 8(d) exhibit complex image, and we can see RP2 is darker than RP1. In order to consider the difference between RP1 and RP2, we have calculated the plot rate for $q$ as shown in Fig. 9. The rate has the peak around $q = 0.48$ that corresponds to the period-1 islands. The second and third peaks correspond to higher-period islands. Note that the difference between RP1 and RP2 increases as $q$ increases from $q = 0.48$ of the first peak. As one reason of the difference, we can say the following: as $q$ increases, the equidistant ISI component and the line spectrum increases, whereas reset points on the base $x = q$ have different $y$ component. $y$ data can have larger variation than the ISI data.

Figure 7: Recurrence plot using $y$ data for $\theta_y = 0.1$, $N = 500$ and $a = 0.2$ (a) $q = 0$, (b) $q = 0.48$, (c) $q = 0.65$, (d) $q = 0.8$
Figure 8: Recurrence plot using the ISI data for $\theta_{ISI} = 0.5$, $N = 500$ and $a = 0.2$ (a) $q = 0$, (b) $q = 0.48$, (c) $q = 0.65$, (d) $q = 0.8$.

5. Conclusions

We have analyzed the RFC having various chaotic dynamics. The spike-trains were analyzed using histogram, RP1 and RP2. Using the histogram, we have classified the value of the ISI. Using the RP1 and RP2, we have visualized dynamical property of spike-trains. Especially, the difference between RP1 and RP2 has been considered. Future problems include more detailed analysis of the characteristics of spike-trains for some key parameters and extracting some key measure(s) from the RP.

References

Switched Systems and Applications to Mutual Synchronization

Andrea Espinel‡ and Ina Taralova†
†IRCCyN UMR CNRS 6597, Ecole Centrale de Nantes
1 Rue de la Noë, 44321 Nantes Cedex, France
Email: andrea.espinel@irccyn.ec-nantes.fr, ina.taralova@irccyn.ec-nantes.fr
‡IRCCyN UMR CNRS 6597, Ecole Centrale de Nantes
1 Rue de la Noë, 44321 Nantes Cedex, France

Abstract—This paper deals with the challenging problem of mutual synchronization of discrete-time chaotic systems. This particular type of synchronization allows to model successfully the complex dynamics of diverse systems, such as ecological, economical, physical, chemical etc. Unlike the usual master-slave synchronization, here we consider bidirectional connected sub-systems with identical priority. Switched Luenberger observers are designed to carry out the mutual synchronization, and their performances are compared in the cases with, and without gaussian noise.

1. Introduction

Mutual synchronization describes a large variety of phenomena ranging from physics and biology to engineering and social sciences. Unidirectional and bidirectional synchronization of continuous time chaotic systems has been investigated in [1,2], fewer papers have been devoted to discrete time systems [3,4].

In [9], the problem of the classical unidirectional master-slave synchronization has been reformulated from control theory point of view, in terms of (non) linear observer design. The novelty in the present paper is that unlike the above master-slave configuration, here the synchronization is to be achieved bidirectionaly, using two symmetrical observers (one for each subsystem). The idea is to exploit the richness of the nonlinear dynamics: for identical parameters, the system can exhibit qualitatively different behaviors (multistability), such as periodic, chaotic etc according to the initial conditions. This provides the challenging opportunity to analyze the mutual synchronization between two subsystems, which have been tuned to different orbits, but also between subsystems with slightly different parameters. The latter assumption can be used to model physical systems subject to ageing, temperature variations etc.

The function that has been chosen as an application example in this paper has already shown its excellent properties as an Efficient Chaotic Random Number Generators (CPRNG). It uses chaotic sampling and ultra weak coupling which beats most of the classical random number generators [5].

2. System Definition

The system under consideration has been introduced for the first time by Lozy in [6]. It can be written as

\[ x(n + 1) = f(x(n)) = A \cdot x(n) \]  

For a second order system, the matrix \( A \) is defined by:

\[
A = \begin{pmatrix}
1 - \epsilon_1 & \epsilon_1 \\
\epsilon_2 & 1 - \epsilon_2
\end{pmatrix}
\]

and \( A \) is the tent function evaluated by the components of the vector \( x \in [-1,1] \):

\[
A(x) = sx + 1 , s = \begin{cases} 
2, & x < 0 \\
-2, & \text{else}
\end{cases}
\]

However, state-space representation of the system is more convenient for control theory analysis [5]. Taking into consideration that the system is also a switched piece-wise affine system, it can be expressed as

\[ x(n + 1) = A_n x(n) + B \]
\[ y(n) = C x(n) \]

where \( s_{10} \) is associated to \( x_1 \) and \( s_{20} \) to \( x_2 \) (see eq.2):

\[
A_n = \begin{pmatrix}
1 - \epsilon_1 s_{10} & \epsilon_1 s_{20} \\
\epsilon_2 s_{10} & 1 - \epsilon_2 s_{20}
\end{pmatrix}, B = \begin{pmatrix} 1 \\ 1 \end{pmatrix}
\]

It comes that the phase plane is divided into four different regions with locally linear behavior.

2.1. Analysis of the System

The chaoticity of the system depends on the choice of \( \epsilon_1 \) and \( \epsilon_2 \). The fixed points determination has been completed by the parameter plane analysis to establish the chaotic regions in the parameter plane. Fixed points are defined by:

\[ x_1(n) = (1 - \epsilon_1) s_{10} x_1(n) + \epsilon_1 s_{20} x_2(n) + 1 \]
\[ x_2(n) = \epsilon_2 s_{20} x_2(n) + (1 - \epsilon_2) s_{20} x_1(n) + 1 \]

Then, each component can be defined in function of the system parameters.

As the values of \( s \) depend on the two components \( (x_1, x_2) \), the fixed points determination have been studied independently in the four regions of the phase plane. The analysis gives two fixed points. In the region \( x_1 \in [-1;0], x_2 \in [-1;0] \), the coordinates of the
fixed point are \((-1,-1)\). In the region \(x_1 \in [0;1], x_2 \in [0;1]\), the fixed point is located at \((1/3,1/3)\).

Hereafter we are interested in the second fixed point whose Jacobean eigenvalues are \((-2, 2(-1 + \epsilon_1 + \epsilon_2))\). It is a saddle between \(0.5 \leq \epsilon_1 + \epsilon_2 \leq 1.5\) and an unstable node outside.

\[ A_n^{-1} = \begin{pmatrix} \frac{\epsilon_2 - 1}{s_{10}} & \frac{1}{s_{10}} - \frac{\epsilon_2 - 1}{s_{10}} \\ \frac{\epsilon_1 + \epsilon_2 - 1}{s_{20}} & \frac{\epsilon_1 + \epsilon_2 - 1}{s_{20}} - \frac{\epsilon_1 - 1}{s_{20}} \end{pmatrix} \]

Furthermore, the line \(x_1(n) = x_2(n)\) is associated with the stable manifold of the saddle fixed point \((1/3,1/3)\). The diagonal is invariant for the recurrence, and its preimage is the anti-diagonal \(x_1(n) = -x_2(n)\). For both of them we obtained the one dimensional Lozi map:

\[ x_1(n+1) = x_2(n+1) = s_{10} x_1(n) + 1 \]  

3. Mutual Synchronization

Unlike [7], here we do not deal with the classical master slave synchronization of the Lozi system, but we are interested in the bidirectional coupling of two connected Lozi subsystems, and their synchronization towards different type of orbits (fixed point or chaotic behavior). To do this, a Luenberger observer has been used for the estimation of \(x\).

\[ \dot{x}(n+1) = A \hat{x}(n) + B (y(n) - y(n)) \]  

Now, to achieve the mutual synchronization, an observer has been designed for each subsystem, called \(x^M(n+1)\) and \(x^S(n+1)\). It should be noted that the super script M and S are chosen for convenience, but they do not indicate here any hierarchy. Thus, the two subsystems are now modeled by:

\[ x^M(n+1) = A x^M(n) + K (y^M(n) - y^S(n)) \]
\[ x^S(n+1) = A x^S(n) + K (y^S(n) - y^M(n)) \]

The gain \(K\) is the same for both subsystems and determines the dynamics of the error defined by:

\[ e(n) = x^M(n) - x^S(n) \]

that is equivalent in the opposite direction \(x^M(n) = x^S(n)\). Then, the matrix that governs the error dynamics is calculated as follows:

\[ e(n+1) = A (x^M(n) - x^S(n)) + 2K (y^M(n) - y^S(n)) \]
\[ e(n+1) = (A + 2KC) e(n) \]

The asymptotic convergence of the synchronization error is guaranteed if the error matrix has eigenvalues in the unit circle. The exact (finite time) convergence is achieved when the eigenvalues are at the origin. The aim is to find the values of \(K\) that synchronize both subsystems. As the error matrix is of second order, it can be achieved in two steps. Therefore, it has to be independent of the way of switching between the four regions of the plane. For \(l \in \{1;4\}, j \in \{1;4\} : \)

\[ (A_l + 2K_l C_l)(A_j + 2K_j C_j) = 0 \]

The solution for the matrix \(K\) presented in [7], has been adapted to the mutual synchronization:

For \(x_1 \in [0;1], x_2 \in [0;1]: \)

\[ A_1 = 2 * \begin{pmatrix} -1 - \epsilon_1 & -\epsilon_1 \\ \epsilon_2 & -1 - \epsilon_2 \end{pmatrix} K_1 = \begin{pmatrix} 2 - \epsilon_1 - \epsilon_2 & \epsilon_2 \\ 1 - 2\epsilon_2 + \epsilon_1 - \epsilon_2 \end{pmatrix} \]

for \(x_1 \in [-1;0], x_2 \in [0;1]: \)

\[ A_2 = 2 * \begin{pmatrix} 1 - \epsilon_1 & -\epsilon_1 \\ \epsilon_2 & -1 - \epsilon_2 \end{pmatrix} K_2 = \begin{pmatrix} \epsilon_1 - \epsilon_2 & \epsilon_2 \\ 1 - 2\epsilon_2 + \epsilon_1 - \epsilon_2 \end{pmatrix} \]

for \(x_1 \in [-1;0], x_2 \in [-1;0]: \)

\[ A_3 = -A_1 K_3 = -K_1 \]

and for \(x_1 \in [0;1], x_2 \in [-1;0] \)

\[ A_4 = -A_2 K_4 = -K_2 \]  

Figure 1. Parameter plane \((\epsilon_1, \epsilon_2)\) of the Lozi system. (1)
4. Results and Discussion

The first analysis deals with the synchronization of two identical chaotic subsystems (same parameters), starting from different initial conditions (belonging to different locally linear regions). The evolution of the error between $x_1^M - x_1^S$, shown in Figure 2, illustrates the exact convergence for $(\epsilon_1, \epsilon_2) = (0.3, 0.1)$ for both subsystems, and initial conditions $(x_1^M(0), x_2^M(0)) = (0.154, 0.289)$; $(x_1^S(0), x_2^S(0)) = (0.131, -0.085)$.

![Figure 2](image-url)

Figure 2: Evolution of the difference between $x_1^M$ and $x_1^S$ for the first ten iterations: exact synchronization.

![Figure 3](image-url)

Figure 3: Quadratic synchronization error, exact observer.

Figure 3 shows the exact quadratic synchronization error, and Figure 4 the quadratic synchronization error using an asymptotic observer with both poles placed at 0.9. It can be seen for Fig.2 that even though the two subsystems start from two different regions, the exact convergence is obtained in only two iterations as in the case of the classical master-slave synchronization [7]. The synchronization error

$$e = \sqrt{(x_1^M - x_1^S)^2 + (x_2^M - x_2^S)^2}$$

(13)

is quite satisfactory ($10^{-15}$, which is close to the computer precision). In the case of asymptotic observer the synchronization is reached as well, but the convergence is much slower (Fig.4). Moreover, the performed tests have shown that unlike the expectations, the asymptotic observer did not outstand the exact observer in the presence of noise.

![Figure 4](image-url)

Figure 4. Quadratic synchronization error, asymptotic observer.

The Lozi system is very complex and exhibits most of the features characterizing nonlinear dynamical systems, such as the extreme sensitivity to the initial conditions and small parameters variations, and multistability: coexistence of fixed point and chaotic attractor has been found at the bifurcation line $\epsilon_1 + \epsilon_2 = 1.5$. Since the map has no stable fixed point, the saddle fixed point (1/3, 1/3) has been selected to test the bidirectional synchronization. The saddle fixed point has been reached, initializing the first subsystem at the stable manifold, taking the initial conditions from $x_{10} = (1/3, 1/3)$. In this particular case, it easily achievable, because the first low rank preimages of the fixed point (1/3, 1/3) lie on the diagonal $x_{10}(n) = x_{20}(n)$, which is invariant by the map $f$, and also the preimage of the diagonal is the anti-diagonal $x_{10}(n) = -x_{20}(n)$, as already shown.

It should be noted that parameters have been tuned outside the usual parameter range of weak coupling [8], but this choice has been done deliberately, in order to analyze the mutual synchronization for different coexisting dynamical behaviors.

![Figure 5](image-url)

Figure 5. Exact synchronization between two subsystems (fixed point - chaotic attractor; identical parameters: alternated white noise).

Figure 5 and Figure 6 show the exact synchronization and the quadratic error between two subsystems with identical parameters. The first one was tuned to exhibit a fixed point trajectory and the second one a chaotic trajectory. The initial conditions for the fixed point had
been chosen on the stable manifold, using the low rank preimages. After the synchronization, both subsystems converge towards the fixed point \((1/3, 1/3)\) which turns out to be attractive after the synchronization. The same parameters have been chosen for both subsystems \((\epsilon_1, \epsilon_2) = (0.88, 0.62)\), i.e. at the bifurcation line \(\epsilon_1 + \epsilon_2 = 1.5\) and synchronization has been obtained for different “master” and “slave” initial conditions \((x_1^M(0), x_2^M(0)) = (0.17, 0.25), (x_1^S(0), x_2^S(0)) = (0.222, 0.429)\). Therefore, besides achieving the successful synchronization, it could be argued that the observer might have a stabilizing effect on the overall system’s behavior.

In addition, Gaussian white noise has been added between \((2500, 3000)\) and \((6500, 7000)\) with 5% amplitude in order to test the robustness of the observers in presence of noise. The obtained results show that the system resynchronizes back successfully when the noise has been removed. Compared to Fig. 6, the perturbation time intervals are longer in Fig. 5 because after the noise removal, there is a chaotic transient before the subsystems resynchronize at the fixed point. On the other hand, the noise could not be rejected; unlike the expectations, qualitatively similar results had been obtained with the asymptotic observer (not shown here for lack of space).

Figure 6. Quadratic synchronization error. (13)

In addition, a small variation of \(\epsilon_1, \Delta \epsilon_2 = (10^{-2})\) has been applied; \((\epsilon_1, \epsilon_2) = (0.88, 0.619)\). The synchronization is successfully achieved again, but towards the chaotic behavior; the chaotic attractor lies on the diagonal \(x_1 = x_2\). The quadratic synchronization error is represented in Fig. 7. If noise is added, the results are qualitatively similar to those in Fig. 6.

Figure 7. Synchronization error between the two subsystems tuned at fixed point and chaotic attractor (slightly different parameters).

Figure 7 shows the synchronization results for the same “master” but another choice of initial conditions and parameters for the “slave”. It exhibits again a chaotic behavior for \((x_1^M(0), x_2^M(0)) = (-0.189, 0.437)\) and in addition, a small variation of \(\epsilon_1, \Delta \epsilon_2 = (10^{-2})\) has been applied; \((\epsilon_1, \epsilon_2) = (0.88, 0.619)\). The synchronization is successfully achieved again, but towards the chaotic behavior; the chaotic attractor lies on the diagonal \(x_1 = x_2\). The quadratic synchronization error is represented in Fig. 7. If noise is added, the results are qualitatively similar to those in Fig. 6.

**Conclusion**

Mutual synchronization of two Lozi subsystems has been designed using symmetric Luenberger observers. Two different kind of observers have been used, exact and asymptotic one, and their performances have been tested in presence of noise. It has been shown that the two subsystems synchronize successfully when they start from different initial conditions, exhibit different permanent regimes, or have different parameters. Current works are carried out to generalize the results for more than two connected subsystems.

**Acknowledgements**

The authors would like to thank prof. Fournier-Prunaret et prof. Lozi for their useful suggestions and comments during the preparation of this paper.

**References**


Synthesis of a spiking oscillator with a desired inter-spike-interval density

Tadashi Tsubone

The Department of Electrical Engineering, Nagaoka University of Technology, 1603–1 Kamitomioka–cho, Nagaoka–shi, Niigata, 940–2188 Japan, tsubone@vos.nagaokaut.ac.jp

Abstract—In this paper, we consider a design problem of a simple spiking oscillator with a desired inter-spike-interval (ISI) density. Our approach to the problem is based on a mapping procedure. The spiking oscillator behaves chaotic motions and generates various spike trains. It is important to study the relationship between a dynamic property of the oscillator and a static property on ISI density not only for scientific investigations, but also for engineering developments. We provide a simple spiking oscillator with piecewise constant vector fields. The dynamics of the system is governed by 1-D piecewise linear return map, therefore the rigorous analysis can be performed. We show an example of spiking oscillators with a conditional ISI density, its synthesis strategy is based on a probability density function of the 1-D return map.

1. Introduction

Chaotic spiking oscillators (abbr. CSO) have been studied in interesting works [1]-[3]. CSOs relate considerably to some spiking neuron neuron models [4]. The spiking neuron models have integrate-and-fire operation that generates a instantaneous pulse signal and reset a state variable instantaneously at the moment when the various reaches a switching threshold. Repeating the operations, CSOs generates a variety of spiking-trains. Study of the spiking-trains is important to develop some researches for information processing of the human brain, for pulse based communication systems and so on[5][6]. Also CSOs are included in hybrid dynamical systems with various nonlinear phenomena and its coupled systems can be developed into efficient applications of neural networks [7]. The analysis of spiking-trains and the simple circuit implementation of CSOs are important.

First, this paper presents a statistical analysis of the inter-spike-intervals (abbr. ISIs) generated from a CSO with piecewise constant vector field. The circuit consists mainly of two capacitors, two nonlinear voltage-controlled current sources and dependent impulsive switches. From the state space description, we derive the embedded return map which governs the qualitative behaviour of the system. The statical analysis of the return map results in the calculation of the invariant density. We derive the invariant density of the 1-D return map and show that the probability distribution of the ISIs for the CSO under consideration can be calculated from the invariant density of the return map. Second, we shows an example of synthesis procedure for a spiking oscillator with a conditional ISI probability density.

2. A chaotic spiking oscillator with piecewise-constant vector field

Figure 1 shows the circuit model of the simple chaotic spiking oscillator. The triangle labeled 1 (−1, respectively) is a linear amplifier with gain 1 (−1, respectively). The triangles labeled "+ −" are comparators. These amplifiers and comparators are realized by an operational amplifier with sufficiently large input impedance. Trapezoids are differential voltage-controlled transconductance amplifiers and their output currents are $i_1$ and $i_2$, respectively. They are characterised by

$$
\begin{align*}
  i_1 &= I_a \cdot \text{sgn}(v_2 - E), \\
  i_2 &= I_a \cdot \text{sgn}(v_2 - v_1), \\
  \text{sgn}(x) &= \begin{cases} 
    1 & \text{for } x \geq 0, \\
    -1 & \text{for } x < 0.
  \end{cases}
\end{align*}
$$

where $v_1$ and $v_2$ are voltages across the capacitors $C_1$ and $C_2$, respectively. $I_a$ is constant which is controlled by a bias current of transconductance amplifiers. Connecting two capacitors to both output terminals of the
transconductance amplifiers, we obtain a two-dimensional nonlinear system. When $S$ is opened, the circuit dynamics is described by
\begin{equation}
\begin{aligned}
\dot{x} &= \alpha \cdot \text{sgn}(y - 1), \\
\dot{y} &= \alpha \cdot \text{sgn}(y - ax),
\end{aligned}
\end{equation}
where "\cdot" represents the derivative of $\tau$, $\alpha$ is a constant value and the following dimensionless variables and parameters are used.
\begin{equation}
\begin{aligned}
\tau &= \frac{I_0}{C_2E}, & x &= \frac{C_1}{C_2}v_1, & y &= \frac{1}{E}v_2, \\
a &= \frac{C_2}{C_1}, & \alpha = \frac{I_a}{I_0} > 0,
\end{aligned}
\end{equation}
where $I_0$ is a virtual constant current for normalizing.
Here, we assume the following parameter condition:
\begin{equation}
a > \frac{\sqrt{2} + 1}{\sqrt{2} - 1}.
\end{equation}
In this parameter range, Equation (2) has unstable rect-spiral trajectories as shown in Fig. 2. The trajectory on the phase space moves around the singular point $\left(\frac{1}{a}, 1\right)$ divergently and it must reach to the half line $b_{th} = \{(x, y) | y = ax, y < 0\}$ as shown in the left figure of Fig. 2.
In this circuit in Fig. 1, $M.M.$ is a monostable multivibrator which outputs pulse signals to close the switch $S$ and to open $\bar{S}$ instantaneously. Two comparators detect the impulsive switching condition. If $v_2 \leq v_1$ or $v_2 \geq 0$, the switch $S$ is opened and $\bar{S}$ is closed. For the meantime, the voltage $v_1$ and $v_2$ is stored to $C_{C1}$ and $C_{C2}$, respectively. If $v_2 > v_1$ and $v_2 < 0$, then $M.M.$ is triggered by the pair of comparators, and the switch $S$ is closed and $\bar{S}$ is opened instantaneously. At that time, the voltage $v_1$ and $v_2$ is reset instantaneously to the inverse voltage $-v_1$ and $-v_2$, respectively. That is,
\begin{equation}
[v_1(t^+), v_2(t^+)]^T = [-v_1(t), -v_2(t)]^T
\end{equation}
for $v_2(t) > v_1(t)$ and $v_2(t) < 0$, where $t^+ = \lim_{\epsilon \to 0}(t + \epsilon)$.
Because the parameter condition (4), the trajectory must reach $l_0 \equiv \{(v_1, v_2) | v_1 = v_2, v_2 < 0\}$ when the switchings occur. Namely, the normalized trajectory must hit $b_{th}$, and jumps from $(x(T_n), y(T_n))$ to $(-x(T_n^+), -y(T_n^+))$ as shown in the left figure of Fig. 2, where $T_n$ is the $n$-th switching moments.
Consequently, Eqn. (2) and (5) with the condition (4) are transformed into
\begin{equation}
\begin{aligned}
\dot{x} &= \alpha \cdot \text{sgn}(y - 1), & \text{for } S = \text{off}, \\
\dot{y} &= \alpha \cdot \text{sgn}(y - ax), \\
[x(\tau^+), y(\tau^+)]^T &= [-x(\tau), -y(\tau)]^T
\end{aligned}
\end{equation}
for $x(\tau) > a \cdot x(\tau)$ and $y(\tau) < 0$,
\begin{equation}
(a > \frac{\sqrt{2} + 1}{\sqrt{2} - 1}).
\end{equation}
Now the system is characterized by only two parameters $a$ and $\alpha$. The right figure of Fig. 2 shows a typical chaotic attractor with $a \simeq 5.84$.
The transconductances are implemented by OTAs (LM13700). Realization procedure of differential voltage-controlled transconductance amplifiers by using OTAs can be found in literature [8]. The monostable multivibrator, the comparators and the analog switches are implemented by IC package of 4538, LM339 and LF398, respectively.

3. Embedded return map
The exact piecewise solution of Eqn. (6) for $S = \text{off}$ can be depicted. Here, let us focus on a trajectory starting from origin at $\tau = 0$ (see Fig. 2). The trajectory rotates divergently around the singular point $(\frac{1}{a}, 1)$ and reaches the switching threshold. A $y-$coordinate of the reaching point is obtained as $-\frac{(\frac{1}{a} + 1)^2}{2} + 1$. Here we define $A \equiv \left(\frac{\alpha + 1}{\alpha}\right)^2 + 1$ and $l \equiv \left\{(x, y) | y < 0, y = ax\right\}$. And we consider the case of $A + 1 > -1$, that is, the minimum value of $y$ is greater than $-1$. In this case, the trajectory starting from $l$ must jumps instantaneously to the symmetric point of the origin, the trajectory rotates $k$-times $(k = 1, 2, 3, \cdots)$ around the singular point and it must return to $l$. We henceforth consider the following parameter range with (4):
\begin{equation}
1 < A \leq 2.
\end{equation}
If we choose $l$ as Poincaré-section, we can define one dimensional return map $f$ from $l$ to itself. Letting $(x(T_n), y(T_n))$ be the starting point, $(x(T_{n+1}), y(T_{n+1}))$ be the return point as shown in left figure of Fig. 2. And letting any points on $l$ be represented by its $y$-coordinate, $f$ is defined by
\begin{equation}
f : l \mapsto l, \quad y_{n+1} = f(y_n),
\end{equation}
where we rewrite $y_n = y(T_n)$.

Figure 2: Behavior of Trajectories on the phase space and a typical chaos attractor. ($a \simeq 5.84$)
By using piecewise-constant trajectories and linear algebraic procedure, we obtain an explicit expression for the function $f$:

$$f(y_n) = \begin{cases} 
-A(y_n+1) + 1 & \text{for } \frac{1}{A} - 1 < y_n \leq 0, \\
-A^2(y_n+1) + 1 & \text{for } \frac{1}{A^2} - 1 < y_n \leq \frac{1}{A} - 1, \\
-A^k(y_n+1) + 1 & \text{for } \frac{1}{A^k} - 1 < y_n \leq \frac{1}{A^{k-1}} - 1, \\
(k = 1, 2, 3, \cdots), 
\end{cases}$$

(9)

where each borders of the piecewise maps, $T_{th_k} = \frac{1}{A^k} - 1$, are derived by solving $0 = -A^k(T_{th_k} + 1) + 1$. Note that the return map $f$ does not depend on a parameter $\alpha$. Typical map $f$ are shown in Fig. 4. In this figure, $k$-th branch from the right corresponds to a trajectory with a $k$ turn spiral on the phase space.

Here, we give the proof for chaos generation of this system. From condition (7), $|\frac{\partial f}{\partial y}| > 1$ is satisfied without discontinuous points and $f(l)$ is obvious, hence $f$ exhibits chaos. In practice, if $1 < A < 4$ is satisfied, the system (6) behaves chaos rigorously. This paper omits the proof but it is easy in a similar way to [8].

4. Probability density of the inter-spike intervals

First, we derive the relationship function $\Delta T(y)$ between the inter-spike intervals $\Delta \tau$ and the state $y(T_n)$ at the moment when a spiking occurs. If the trajectory hits the threshold $l$ at $\tau = T_n$, a spiking occurs and the time interval until next spiking is determined uniquely by $y(T_n)$. By using return map (8) and linear algebraic procedure, we obtain the expression for the function $\Delta T(y)$.

$$\Delta \tau = \Delta T(y) = \begin{cases} 
\frac{1}{\alpha} (1+y)(1+\sqrt{A})^2 & \text{for } \frac{1}{A} - 1 < y_n \leq 0, \\
\frac{1}{\alpha} (1+y)(1+\sqrt{A})^2(1+A) & \text{for } \frac{1}{A^2} - 1 < y_n \leq \frac{1}{A} - 1, \\
\cdots & \text{for } \frac{1}{A^{k}} - 1 < y_n \leq \frac{1}{A^{k-1}} - 1, \\
(k = 1, 2, 3, \cdots), 
\end{cases}$$

(10)

The examples of function $\Delta T$ is depicted in Fig. 4.

Second, we consider an invariant measure of the return map $f$. In order to derive the invariant measure, Frobenius-Perron operator $P$ is well known to be useful. For the non-invertible map $f(y)$, its invariant measure can be obtained as the steady state of the iteration $f_{k+1}(y) = Pf_k(y)$ [9]. Here, we fix the parameter $A$ to 2 for simplicity. In this case, the invariant measure $INV(y)$ of return map $f$ must be uniform.

$$INV(y) = 1, \quad y \in [-1,0]$$

(11)

Finally, using the invariant measure of return map $INV(y)$ and the relationship function $\Delta T$, we can obtain directly the probability density function of ISI $d(\Delta \tau)$.

$$d(\Delta \tau) = \int_{-1}^{0} \left( \frac{\partial \Delta T(y)}{\partial y} \right)^{-1} \delta(y - y_T)INV(y)dy,$$

(12)

where $y_T$ is the value such as $\Delta \tau = \Delta T(y_T)$, namely,

$$y_T = \Delta T^{-1}(\Delta \tau),$$

(13)

where $\Delta T^{-1}$ represents a inverse function of $\Delta T$. The probability density function of ISI $d(\Delta \tau)$ with a parameter $a \simeq 5.84$ and $\alpha = 1$ is as shown in Fig. 5.
The right figure of Fig. 5 shows histograms obtained by 100,000 sampled data of a numerical simulation.

5. Synthesis of a spiking oscillator with a conditional ISI density

In this section, we provide an example of synthesis approach to realize a spiking oscillator with a desired probability density of ISI. Here, we show the procedure to realize uniform density of ISI by controlling the relationship function $\Delta T$.

We consider the $\alpha$ is a time variant parameter such as

$$
\alpha = \begin{cases} 
\alpha_v & \text{for } T_n \leq \tau < T_n + \tau_{\alpha}, \\
1 & \text{otherwise},
\end{cases}
$$

where $T_n$ is the $n$-th switching moments, $\tau_{\alpha}$ is a suitable constant delay-time and

$$
\alpha_v = 1 - \frac{1}{\tau_{\alpha}} \frac{(1 + \sqrt{A})^2}{1 - A} \left( y(T_n^+) - 1 \right).
$$

In this case, the relationship function $\Delta T$ changes as shown in Fig. 6, but the return map $f$ is same to the case of constant $\alpha$. From these the function and the return map, we obtain a uniform density of ISI as shown in Fig. 7.

6. conclusion

We considered a design problem of a simple spiking oscillator with a desired inter-spike-interval (ISI) density. Future subjects are a verification by experimental systems and an approach for generalisation.

References


Exponential Transient Oscillations and Standing Pulses in Rings of Coupled Symmetric Bistable Maps

Yo Horikawa
Faculty of Engineering, Kagawa University
2217 Hayashi-cho, Takamatsu, 761-0396 Japan
Email: horikawa@eng.kagawa-u.ac.jp

Abstract- The author studies transient states the duration of which increases exponentially with system size (exponential transients) in rings of coupled symmetric bistable maps. When coupling is unidirectional (one-way coupling), transient oscillations rotating in a ring of maps exist. When coupling is bidirectional (two-way coupling), transient pulse patterns exist. In both coupling the duration of transient states on the way to spatially homogeneous steady states increases exponentially with the number of elements. Further, the duration of transient states occurring from random initial states is distributed in a power law form.

1. Introduction

Exponential transients are transient states the duration (lifetime) of which increases exponentially with system size. They were first found in a one-dimensional bistable reaction-diffusion system, which is known as the time-dependent Ginzburg-Landau equation in the field of phase transitions [1]. In one-dimensional domain, a transient front (kink), a pulse (kink-antikink pair) and multiple pulse patterns are formed until a system reaches a spatially homogeneous steady state. The motion of these patterns extremely slow, which is called metastable dynamics, and the duration of them increases exponentially with domain length. Such dynamically metastable patterns have since been studied in various reaction-diffusion equations and convection-diffusion equations in multi-dimensional domains [2].

Another form of exponential transients is transient chaos or supertransients, which was found in coupled map lattices [3]. A lattice shows complex turbulent patterns retaining invariant statistical measures and suddenly falls onto a periodic attractor. The duration of complex behaviors also increases faster than exponentially with the number of elements. Such transient chaos has since been studied in various systems including reaction-diffusion equations, excitable media and complex networks [4].

Recently, exponential transient oscillations with the same mechanism as metastable dynamics were found in a unidirectionally coupled ring of sigmoidal neurons [5]. Qualitatively the same transient oscillations were also found in a discrete-time system, i.e. a ring of directly coupled bistable cubic maps [6]. These exponential transient patterns with dynamical oscillations are of interest since they lie between simple static spatial transient patterns and complicated spatiotemporal chaotic transient patterns.

In this study, the author considers exponential transients in rings of symmetric bistable maps with standard coupling. In unidirectionally coupled maps, traveling pulses rotating in a ring similar to those in a ring neural network arise in its transient state. In bidirectionally coupled maps, transient pulse patterns like metastable patterns in reaction-diffusion systems exist when coupling is strong and nonlinearity is weak. It is shown that both transient states last exponentially long time with the number of elements and that the duration of them under random initial conditions obeys a power-law distributions up to a cut-off.

2. Rings of Coupled Bistable Maps

The following rings of coupled maps are considered.

\[ f(x) = x + K \sin(2\pi x)/(2\pi) \]

where coupling is unidirectional in Eq. (1a) with strength \( \varepsilon \) and is bidirectional in Eq. (1b) with strength \( \varepsilon/2 \). Figure 1 shows a symmetric bistable map \( f(x) \) with some values of a parameter \( K \) of nonlinearity. The map as well as the rings have bistable fixed points \( x_n = \pm 1/2 \) and an unstable one \( x_n = 0 \).

![Symmetric bistable maps](image-url)

Fig. 1. Symmetric bistable maps.
Figure 2(a)(b) shows examples of transient oscillations occurring under random initial conditions with a uniform distribution: \( x_n(0) \sim U(-1/2, 1/2) \) in Eq. (1a) (unidirectional coupling) with \( N = 20, \varepsilon = 0.2 \) (a), 0.8 (b) and \( K = 0.5 \). Time courses of the states \( x_n(t) \) of the first elements are plotted in upper panels, and spatiotemporal patterns of the states of elements are plotted with black (white) for positive (negative) signs in lower two panels. Single traveling pulses are quickly generated from random initial states, while they are unstable and the states converge to one \( (x_n = -1/2) \) of spatially homogeneous bistable states after a long time. The speed of a traveling wave is slow when coupling strength \( \varepsilon \) is small (a), and vice versa (b).

Figure 2(c) shows an example of transient pulses in Eq. (1b) (bidirectional coupling) with \( N = 40, \varepsilon = 0.5 \) and \( K = 0.1 \), in which snapshots of the states of elements at \( t = 0, 100, 5000, 10000, \ldots, 35000 \) (by 5000) are plotted. A standing pulse is also generated quickly, keeps its form for a long time, and finally collapses.

3. Transient Oscillations with Unidirectional Coupling

3.1. Kinematics of Traveling Waves

When coupling is unidirectional and the number of elements is even: \( N = 2l_0 \), an unstable aperiodic (quasi-periodic) traveling wave exists in the subspace: \( x_n = x_{n+N/2} \) (\( 1 \leq n \leq l_0 \)). It is a symmetric pulse wave with equal pulse widths and is observed with computer simulation of Eq. (1a) under a symmetric initial condition:

\[
x_n = e^{\sqrt{2}t} \quad (1 \leq n \leq l_0), \quad x_n = -e^{\sqrt{2}t} \quad (l_0 < n \leq N)
\]

where \( l_0 \) and \( N - l_0 \) are initial spatial pulse widths, i.e. the numbers of elements in pulses, and both are set to be a half \( l_0 = (N/2) \) of the number of elements. The speed of the symmetric traveling wave depends on its spatial pulse width \( l_0 \). We define the propagation time (an inverse of the speed) of the traveling wave by time required for the propagation of pulse fronts over one unit distance (one element). Figure 3 shows a semi-log plot of \( \Delta t(l_0) - \Delta t_{\infty} \) obtained with computer simulation of Eq. (1a) under Eq. (3) with \( \varepsilon = 0.2 \) and \( K = 0.5 \) against \( l_0 \), where \( \Delta t_{\infty} = \Delta t(l_0 = 8) \) is propagation time for \( l_0 \) large enough. Difference in the propagation time decreases exponentially with \( l_0 \), hence the number \( N \) of elements. It is approximated by

\[
\Delta t(l_0) = \Delta t_{\infty} + b \exp(-al_0) \quad (l_0 = N/2)
\]

\[
a \approx 2.375, \quad b \approx 52.5, \quad \Delta t_{\infty} \approx 6.345
\]

Let us assume that the propagation time of each pulse front depends on its backward pulse width as was done in [6]. Changes in the spatial width \( l \) of one pulse are then expressed by difference between the inverses of the propagation times of two pulse fronts:

\[
dl/dt = 1/\Delta t(l) - 1/\Delta t(N-l)
\]

\[
= -\beta \{ \exp(-\alpha l) - \exp[-\alpha(N-l)] \}
\]

\[
\alpha = \alpha_0 \approx 2.375, \quad \beta = b/\Delta t_{\infty}^2 \approx 1.304
\]

It should be noted that the propagation time of a pulse fronts depends on the forward pulse width in a ring neural network [5]. The mechanism causing the dependence of the propagation time of a front on the backward pulse width in coupled map lattices is not clear at present.

3.2. Properties of Transient Oscillations

The solution \( h(t) \) of Eq. (5) with initial pulse width \( h(0) = l_0 \) is obtained as

\[
\exp(-\alpha h(t)) = \exp(\alpha N/2) \tanh[\exp(-\alpha N/2)\alpha \beta t + \arctanh[\exp(\alpha l_0 - N/2)]]
\]

The duration \( T \) of the traveling wave and transient oscillation is given by letting \( h(T) = 0 \):

\[
T(l_0; N) = \frac{\exp(\alpha N/2)}{\alpha \beta} \left[ \arctanh[\exp(\alpha l_0 - N/2)] - \arctanh[\exp(-\alpha N/2)] \right]
\]
Simpler forms of Eqs. (6) and (7) are given by letting \( N \) be infinity in Eq. (5):

\[
\frac{d}{dt} = -\beta \exp(-\alpha t)
\]

\[ l(t) = 1/\alpha \cdot \log[\exp(\alpha l_0) - \alpha \beta t] \quad (l(0) = l_0 < N/2)
\]

\[ T(l_0) = [\exp(\alpha l_0) - 1]/(\alpha \beta) \quad (l(T) = 0)
\]

Figure 4(a) shows a semi-log plot of the duration \( T \) of transient oscillations against initial pulse width \( l_0 \) in Eq. (1a) with \( N = 21 \), \( \varepsilon = 0.2 \) and \( K = 0.5 \). Plotted are the results of computer simulation of Eq. (1) under Eq. (3) (solid circles) and \( T(l_0) \) in Eq. (8) (a dashed line). The duration increases exponentially with initial pulse width, and Eq. (8) agrees with the simulation results.

Figure 4(b) shows a semi-log plot of the duration \( T \) of transient oscillations against coupling strength \( \varepsilon \) obtained by computer simulation of Eq. (1a) under Eq. (3) with \( N = 20 \) and three sets of \((K, l_0)\). One of interest is symmetry in \( T \) with respect \( \varepsilon = 1/2 \) despite the fact that the propagation time of a pulse front monotonically increases with \( \varepsilon \) as shown in Fig. 1. The other is that unstable asymmetric traveling waves are stabilized into stationary standing pulses as \( \varepsilon \to 0 \), while they become stable traveling waves as \( \varepsilon \to 1 \). That is, the duration \( T \) diverges near both sides of graphs in Fig. 4(b).

Further, the distribution \( h(T) \) of the duration \( T \) of transient oscillations occurring from random initial states is obtained by letting initial pulse width \( l_0 \) be distributed uniformly in \([0, N/2)\) with \( T(l_0; N) \) in Eq. (7):

\[
\int_0^l U(0, N/2) dl_0 = \int_0^T h(T') dt'
\]

\[
h(T) = \frac{1}{|dT(l_0; N)/dl_0|} \cdot \frac{2 N}{2} = \frac{d_0(T; N)}{dt} \cdot \frac{2}{N}
\]

\[
= 4\beta \exp(-\alpha N/2) \cosh[2\exp(-\alpha N/2)\alpha \beta T + \arctanh(\exp(-\alpha N/2))] / N
\]

There is a cut-off \( T_c \) in \( h(T) \) in Eq. (10), and the duration is distributed in an inverse power law form for \( T < T_c \):

\[
h(T) = \frac{\beta}{\alpha \beta T + 1} \cdot \frac{2}{N} \quad (0 < T < T_c)
\]

while it is distributed exponentially for \( T > T_c \):

\[
h(T) \approx 4\lambda \exp(-\lambda T)/(\alpha N)
\]

\[
\lambda \approx 2\alpha \beta \exp(-\alpha N/2) \quad (T > T_c)
\]

Figure 5 shows a log-log plot of the distribution \( h(T) \) of the duration \( T \) of oscillations, in which plotted are a histogram obtained with \( 10^4 \) runs of computer simulation of Eq. (1a) with \( N = 15 \), \( \varepsilon = 0.2 \) and \( K = 0.5 \) under \( x_0(0) \sim U(-1/2, 1/2) \) (solid circles), Eq. (10) (a solid line), Eq. (11) (a dashed line) and Eq. (12) (a dotted line). Equation (10) agrees with the simulation results, and Eq. (11) agrees with them also up to a cut-off: \( T_c \approx 1.76 \times 10^7 \).

Fig. 5. Distribution of the duration of oscillations.

The mean \( m(T) \) and variance \( \sigma^2(T) \) of the duration \( T \) of oscillations also increase exponentially with the number \( N \) of elements:

\[
m(T) = 2[\exp(\alpha N/2) - 1 - \alpha N/2]/(\alpha^2 \beta N)
\]

\[
\sigma^2(T) = [\exp(\alpha N) - 4 \exp(\alpha N/2) + 3 + \alpha N]/(\alpha^3 \beta^2 N) - m(T)^2
\]

and the coefficient of variation \( CV(T) \) increases with the square root of \( N \). Figure 6 shows the logarithms of \( m(T) \) and \( \sigma(T) \) as well as \( CV(T) \) of the duration against the
number $N$ of elements for Eq. (1a) with $\varepsilon = 0.2$ and $K = 0.5$. Plotted are estimates with $10^4$ runs of computer simulation of Eq. (1a) (symbols) and Eq. (13) (lines), in which they agree with each other.

\begin{align*}
\log_{10}(m(T)) & \text{ with Eq. (1)} \\
\log_{10}(m(T)) & \text{ in Eq. (13)} \\
\log_{10}(\sigma(T)) & \text{ with Eq. (1)} \\
\log_{10}(\sigma(T)) & \text{ in Eq. (13)} \\
\log_{10}(C(V(T))) & \text{ with Eq. (1)} \\
\log_{10}(C(V(T))) & \text{ in Eq. (13)}
\end{align*}

Fig. 6. $m$, $\sigma$ and $CV$ of the duration of oscillations.

4. Transient Pulses with Bidirectional Coupling

When coupling is bidirectional, computer simulation of Eq. (1b) can show that standing pulses are stabilized when pulse width is over a threshold depending on parameter values. It is known that such stabilization of spatially inhomogeneous patterns occurs generally in spatially discrete systems. However, pulses of smaller width than a threshold are still unstable, and the duration of them increases exponentially with pulse width in the same way as that for unidirectional coupling. Figure 7 shows a semi-log plot of the duration $T$ of pulses against initial pulse width $l_0$ obtained by computer simulation of Eq. (1b) under Eq. (3) with $N = 100$ and three sets of $(K, \varepsilon)$. The duration increases exponentially, and a threshold length increases as $K$ decreases and $\varepsilon$ increases.

\begin{align*}
\log_{10}(m(T)) & = \alpha + \beta \log_{10}(l_0) \\
\log_{10}(\sigma(T)) & = \alpha + \beta \log_{10}(l_0) \\
\log_{10}(C(V(T))) & = \alpha + \beta \log_{10}(l_0)
\end{align*}

Fig. 7. Duration of pulses vs initial pulse width $l_0$.

Figure 8 shows a log-log plot of the distribution $h(T)$ of the duration of pulses, in which plotted are a histogram obtained with $10^4$ runs of computer simulation of Eq. (1b) under $x_0(0) \sim U(-1/2, 1/2)$ with $N = 40$, $\varepsilon = 0.5$ and $K = 0.1$ (solid circles) and Eqs. (10) - (12) (lines). The values of parameters are estimated by fitting $T(l_0)$ in Eq. (8) to $T(l_0)$ in Fig. 7 and are set to be $\alpha = 0.651$ and $\beta = 0.487$. The duration is distributed in an inverse power law form up to a cut-off $T_c = 1.14 \times 10^6$, and Eqs. (10) and (11) agree with the simulation results except for small $T (< 10)$.

\begin{align*}
\log_{10}(h(T)) & = \alpha - \beta \log_{10}(T) \\
\log_{10}(h(T)) & = \alpha - \beta \log_{10}(T)
\end{align*}

Fig. 8. Distribution of the duration of pulses.

5. Conclusion

Properties of transient oscillations in a ring of unidirectionally coupled symmetric bistable maps and transient pulses in a ring of bidirectionally coupled maps were studied. Changes in pulse width were described by qualitatively the same equation (Eq. (5)) as those in reaction-diffusion systems [1] and ring neural networks [5]. The duration of both transient states increased exponentially with pulse width and the number of elements. Further, the duration of them under random initial conditions obeyed a power law distribution function.

Analytical derivation of the kinematics (Eqs. (4) and (5)) of the motion of pulses and analysis of intrinsic duality causing the symmetry in Fig. 4(b) and bifurcations causing changes in the stability of traveling (Fig. 4(b)) and standing (Fig. 7) pulses are future studies.

References

Clustering synchronization in pulse-coupled oscillators with a refractory period and frequency distribution

Takuya Okuda, Keiji Konishi, Naoyuki Hara and Hideki Kokame

Dept. of Electrical and Information Systems, Osaka Prefecture University
Email: konishi@eis.osakafu-u.ac.jp

Abstract—This paper investigates phenomena in pulse-coupled oscillators with a refractory period and frequency distribution in the parameter regions where the complete synchronization does not occur. Although our previous studies, [Konishi and Kokame, 2008] and [Okuda, Konishi, and Kokame, 2009], investigated the complete synchronization theoretically and experimentally, they did not mention other phenomena. The present paper numerically reveals that two types of clustering synchronization occur not in the complete-synchronization parameter region.

1. Introduction

Wireless sensor networks, which consist of a large amount of sensor nodes, have received considerable attention in the field of information and computer engineering [1]. In a wireless sensor network, each node autonomously gathers physical information around itself, and communicates with each other. For practical implementation of the networks, the time synchronization of all the nodes is one of the important problems, since it plays crucial role on analysis of information, communication timing control and time division multiple access. A lot of researchers proposed the various rules to overcome this problem [2]; pulse-coupled oscillators have been regarded as one of the effective schemes [3].

Wireless sensor networks consist of a large amount of tiny sensor nodes which are spatially distributed and whose number changes from hour to hour: some nodes sometimes get out of order and some nodes newly get plugged-in to the network. Thus, we have to consider the following three specifications. The first specification is that the communication time for each node should be shortened as much as possible. Since each node is autonomous and they do not have an external electric power-supply, its power consumption must be reduced to extend its battery life. The second one is that the nodes have to use standard popular-priced electronic devices in order to reduce hardware costs. The final one is that time synchronization should be robust for some disturbance. For example, the number of sensor nodes changes one after another.

In the fields of electronic engineering and nonlinear physics, synchronization in coupled oscillators have been investigated over many years [4, 5]. Mirollo and Strogatz reported that a globally pulse-coupled oscillators exhibits synchronization for almost all initial conditions [6]. Since then, this type of network has been investigated for many situations: considering a frequency distribution [7], a refractory period [8], a local coupled network [9], and transmission delay of the pulse signals [10]. In addition, in order to demonstrate the synchronization in coupled oscillators, electronic circuits for the integrate-and-fire models were proposed [11, 12, 13, 14, 15].

The pulse-coupled oscillators with both a refractory period and frequency distribution, which take the above three specifications into consideration, were proposed in our previous study [16]. Furthermore, this theoretical result was verified by circuit experiments [17]. Although these works focused only on the stability conditions of local and global synchronization, they did not investigate the other nonlinear phenomena. The present paper numerically investigates such nonlinear phenomena in the pulse-coupled oscillators.

2. Pulse-coupled oscillators

This section shall review the coupled oscillators and their stability which were considered in our previous works [16, 17]. A network consists of \( N \) oscillators. The phase of \( i \)-th oscillator is denoted by \( \phi_i(t) \in [0, 1) \) and developed by parameter \( \omega_i \in [\omega, 1) \),

\[
\frac{d\phi_i(t)}{dt} = \omega_i \quad (i = 0, 1, ..., N-1).
\]

The upper limit and the lower limit of \( \omega_i \) are set to 1 and \( \omega \) respectively. The lower limit is within the range \( \omega \in (0, 1] \). Let \( k \in \mathbb{Z}_+ \) be the number of firings in the network. The phase \( \phi_i(t) \) reaches 1 at time \( t^k \) (i.e., just before \( k \)-th fire), and then the \( i \)-th oscillator fires and broadcasts a signal to all the other oscillators. Then, the phase \( \phi_i(t) \) is immediately reset to zero at time \( t^{k+} \) (i.e., just after \( k \)-th fire),

\[
\phi_i(t^k) = 1 \Rightarrow \phi_i(t^{k+}) = 0.
\]

At the same instant, \( j(\neq i) \)-th oscillator being active, \( \phi_j(t^k) \in [\delta, 1) \), are forced to be reset to zero by the
broadcast signal as follows:
\[
\phi_i(t^k) = 1 \quad \text{and} \quad \phi_j(t^k) \in [\delta, 1) \Rightarrow \phi_j(t^{k+}) = 0, \quad \forall j \neq i.
\]

(3)

The oscillators being sleep, \( \phi_j(t) \in [0, \delta) \), maintain their phase, that is,
\[
\phi_i(t^k) = 1 \quad \text{and} \quad \phi_j(t^k) \in [0, \delta) \Rightarrow \phi_j(t^{k+}) = \phi_j(t^k), \quad \forall j \neq i.
\]

(4)

A definition of synchronization is provided below [16].

[Synchronization] All the oscillators are said to synchronize if they repeat to fire simultaneously,
\[
\phi_i(t^{k+}) = 0, \quad \forall i \in \{0, 1, \ldots, N - 1\}, \quad \forall k \geq n,
\]

with a constant period,
\[
t^{(k+1)+} = \frac{1}{\omega}, \quad \forall k \geq n,
\]

(6)

after \( n \) fires. \( \hat{\omega} \) is the maximum frequency of all the oscillators,
\[
\hat{\omega} := \max_{i \in \{0, 1, \ldots, N - 1\}} \omega_i.
\]

(7)

The following two stabilities are defined for synchronization in these pulse-coupled oscillators [16].

[Local stability] Suppose that the initial conditions of all the oscillators are
\[
\phi_i(0) \in [\delta, 1), \quad \forall i \in \{0, 1, \ldots, N - 1\}.
\]

(8)

The synchronization is said to be locally stable if all the oscillators have synchronized since the first fire \( (n = 1) \).

[Global stability] Suppose that the initial conditions of all the oscillators are
\[
\phi_i(0) \in [0, 1], \quad \forall i \in \{0, 1, \ldots, N - 1\}.
\]

(9)

The synchronization is said to be globally stable if all the oscillators have synchronized since \( n \)-th fire.

It is obvious that the local stability is a necessary condition of the global stability. The design procedure which allows us to obtain locally stable and globally stable is given by the following two conditions.

[Local stability condition] If the lower bound of the parameter \( \omega \) is greater than or equal to the refractory period \( \delta \),
\[
\delta \leq \hat{\omega},
\]

(10)

then the synchronization is locally stable.

Figure 1: Local and global stability conditions in the \( \omega - \delta \) space.

[Global stability condition] If the lower bound of the parameter \( \omega \) and the refractory period \( \delta \) satisfy
\[
\delta \leq g(n, \omega),
\]

(11)

\[
g(n, \omega) := \begin{cases} \frac{n - \frac{n}{\gamma}}{\gamma}, & (\omega \geq \omega^*_n) \\ \omega, & (\omega \leq \omega^*_n) \end{cases},
\]

(12)

then all the oscillators synchronize after at most \( n \) \((\geq 2)\) fires for any initial condition.

Figure 1 sketches the local and global stability conditions in the \( \omega - \delta \) space. The previous works guarantee only that the complete synchronization occurs in the global stability region [16, 17]; in other words, we do not know what kind of phenomena occurs in the other region. The next section shall investigate the phenomena numerically.

3. Numerical Simulation

The number of oscillators is set to \( N = 3000 \), and their initial phases are randomly chosen from \( \phi_i(0) \in [0, 1] \). All the oscillators are governed by Eqs. (1) (4), and run until the 5000th fires. Figure 2 shows that the steady-state situations depend on the parameters \((\omega, \delta)\).

The time series data of all the oscillators at point A in Fig. 2 is illustrated in Fig. 3(a). The solid and dash lines represent the upper and lower phase limits of the oscillators. It can be seen that all the oscillators fires simultaneously. Such synchronization is achieved if the parameters are chosen from the blue region in Fig. 2. It is obvious that the blue region completely agrees with the global stability condition with
Figure 2: Stead-state situations in the $\omega - \delta$ space for $N = 3000$. (i) blue region: synchronization, (ii) green region: clustering synchronization, (iii) red region: fluid clustering synchronization, (iv) black region: no-synchronization.

$n \to +\infty$ shown in Fig. 1. The time series data at point B in the green region is shown in Fig. 3(b). An oscillator belonging to a cluster (i.e., a vertical thick strip in Fig. 3(b)) remains in the cluster. This phenomenon, so called clustering synchronization, is observed when the parameters are within the green region. Remark that the green region satisfies the local stability condition, but not global one. It is assumed that each cluster in Fig. 3(b) has the oscillator whose angular frequency is equal to 1. However, such assumption does not have a reality in practical situations. In real system, each cluster is locked with its own frequency; as a result, the clusters would repeat their mergence and separation. The time series data at point C in the red region is shown in Fig. 3(c). An oscillator belonging to a cluster changes its cluster from hour to hour. Such fluid clustering synchronization is observed in the red region, where the local stability condition does not hold. The time series data at point D in the black region is shown in Fig. 3(d). All the oscillators fire independently.

In order to investigate the influence of the steady-state situations on the size of coupled oscillators, the number of oscillators is reduced to $N = 500$. Figure 4 shows the steady-state situations for small size coupled oscillators. There are some red points and blue points in the green region. This fact allows us know that, for not large enough number of oscillators, the clustering synchronization in Fig. 2 is not always achieved even if the parameters satisfied the local stability condition.

Figure 3: Time series data of oscillators.
Figure 4: Steady-state situations in the $\omega - \delta$ space for $N = 500$.

4. Conclusion

This paper investigated the clustering synchronization in the pulse-coupled oscillators with a refractory period and frequency distribution. From our numerical simulations, we can see that the clustering synchronization can be achieved when the parameters satisfy the local condition and but not satisfy the global condition. In addition, the fluid clustering synchronization is observed when the parameters do not satisfy the local condition. We think that these results deserve further research, as they could be used for practical systems such as wireless sensor networks and neural networks.

References

Statistical characters of synchronization-optimized oscillator networks

Tatsuo Yanagita† and Alexander S. Mikhailov‡

†Research Institute for Electronic Science, Hokkaido University,
Sapporo 001-0020, Japan
‡Fritz-Haber-Institut der Max-Planck-Gesellschaft
Faradayweg 4-6, 14195 Berlin, Germany
Email: yanagita@nsc.es.hokudai.ac.jp

Abstract—Starting with an initial random network of oscillators with a heterogeneous frequency distribution, its autonomous synchronization ability can be largely improved by appropriately rewiring the links between the elements. Ensembles of synchronization-optimized networks with different connectivities are generated and their statistical properties are studied.

1. Introduction

In the last decade, much interest has been attracted to studies of complex networks consisting of dynamical elements involved in a set of interactions [1]. Particular attention has been paid to problems of synchronization in network-organized oscillator systems [2, 3]. Investigations focused on understanding the relationship between the topological structure of a network and its collective synchronous behavior [4]. Recently, it has also been shown that the ability of a network to give rise to synchronous behavior can be greatly enhanced by exploiting the topological structure emerging from the growth processes [5]. However, full understanding of how the network topology affects synchronization of specific dynamical units is still an open problem.

One possible approach is to use evolutionary learning mechanisms in order to construct networks with prescribed dynamical properties. Several models have been explored, where dynamical parameters were modified in response to the selection pressure via learning algorithms, in such a way that the system evolved toward a specified goal [6, 7]. In our study, this approach is employed to design phase oscillator networks with synchronization properties. We consider adaptive evolution of a network of coupled heterogeneous phase oscillators. The question is how to connect a set of phase oscillators with given natural frequencies, so that the resulting network would exhibit the strongest synchronization, under the constraint that the total number of available links is fixed.

To design optimal networks, stochastic Markov Chain Monte Carlo (MCMC) method with replica exchange is used by us. Large ensembles of optimal networks are constructed and their common statistical properties are analyzed.

2. Model and the Optimization Method

We consider $N$ oscillators with different natural frequencies placed onto the nodes of a network. The evolution of this system is given by

$$\frac{d\theta_i}{dt} = \omega_i + \frac{\lambda}{N} \sum_{j=1}^{N} w_{ij} \sin(\theta_j - \theta_i),$$  \hspace{1cm} (1)

where $\omega_i$ is the natural frequency of oscillator $i$ and $\lambda$ is the coupling strength. The weights $w_{ij}$ define the adjacency matrix $w$ of the interaction network: $w_{ij} = 1$ if oscillator $i$ interacts with oscillator $j$, and $w_{ij} = 0$ otherwise. The adjacency matrix is generally asymmetric.

To quantify synchronization of the oscillators, the Kuramoto order parameter

$$r(t) = \frac{1}{N} \left| \sum_{i=1}^{N} \exp(i\theta_i) \right|$$ \hspace{1cm} (2)

is employed. Under perfect synchronization, we have $r = 1$, whereas $r \sim O(N^{-1/2})$ in absence of coupling for randomly drawn natural frequencies. A second-order transition takes place at some critical coupling strength $\lambda_c$ from the desynchronized to the synchronized states [8].

To measure the degree of synchronization, we numerically integrate Eq. (1) for given initial conditions $\theta(t = 0) \in [0, 2\pi)$ and calculate the average modulus of $r(t)$ over a long time $T$.

$$R(w) = \left\{ \frac{1}{T} \int_{0}^{T} r(t) dt \right\}_{\text{init.}},$$ \hspace{1cm} (3)

where $\langle \ldots \rangle_{\text{init.}}$ represents an average over many realizations with different initial conditions $\theta(0)$.

Our aim is to determine the network $w$ which would exhibit the highest degree of synchronization, provided that the total number of links is fixed and a set of natural frequencies is given. The network construction can be seen as an optimization problem. The optimization task is to maximize the order parameter and, possibly, bring it to unity by changing the network $w$. An approximate standard approach to the problems of complex combinatorial optimization, such as the traveling salesman problem, is provided by
In a REMC simulation, a number of replicas \( \{w_m\} \) with different inverse temperatures \( \beta_m \) are evolved in parallel. At regular evolution time intervals, the performances of a randomly selected, adjacent pair of replicas are compared. The running configurations of the two selected replicas are exchanged with the probability min \([1, \exp(\Delta \beta \Delta R)]\), where \( \Delta \beta = \beta_{m+1} - \beta_m \) is the difference of the inverse temperatures of the pair and \( \Delta R = R(w_{m+1}) - R(w_m) \) is the difference of their performances.

Explicitly, the algorithm is defined as follows:

1. The states of replicas \( \{w_0\} \) are initialized by random networks (which is chosen as a random Erdős - Rényi network).

2. The candidate for the next network \( w_m \) at iteration step \( n \) is obtained from the current network \( w_m^{(n)} \) by rewiring one of its links. A randomly chosen link is moved to a randomly chosen link vacancy, so that the total number of links remains conserved.

3. The evolution equations (1) for the network \( w_m^{(n)} \) are integrated using the standard Euler algorithm. The order parameter is then calculated and averaged over the time interval \( t \in [0, T] \) and over a fixed number of realizations starting from different random initial conditions. Thus, the synchronization property \( R(w_m^{(n)}) \) of the candidate network is determined.

4. Next, a random number \( x \in [0, 1] \) is uniformly drawn. If

\[
x < \frac{\exp(\beta R(w_m^{(n)}))}{\exp(\beta R(w_m^{(n-1)}))},
\]

the candidate is accepted and taken as \( w_m^{(n+1)} = w_m^{(n)} \); otherwise nothing is changed, so that \( w_m^{(n+1)} = w_m^{(n)} \).

5. At regular evolution time intervals, the performances of a randomly selected, adjacent pair of replicas are compared. The running configurations of the two selected replicas are exchanged with the probability

\[
\min \left[1, \exp \left( (\beta_{m+1} - \beta_m) (R(w_{m+1}^{(n+1)}) - R(w_m^{(n+1)}) \right) \right].
\]

6. Return to step (2) until the statistical average Eq. (4) converges.

### 3. Numerical analysis

To determine the synchronization degree of a given network at each iteration step of the optimization procedure, Eq. (1) was numerically integrated with the time increment \( \Delta t = 0.05 \). Averaging over five independent realizations started from different random initial conditions has been furthermore performed at each iteration step. Oscillator ensembles of sizes \( N = 10 \) and 20 were considered. Natural frequencies of the oscillators were always chosen as \( \omega_i = -\gamma + 2\pi i/N \), so that they uniformly distributed within the interval \([-\gamma, \gamma]\). For time averaging, intervals of length \( T = 100 \) and 200 were typically used. The results did not significantly depend on \( T \) when sufficiently large lengths \( T \) were taken. Using the order parameter, graphs \( w \) were sampled by the REMC optimization method. In parallel, evolution of \( M \) replicas with the inverse temperatures...
3.1. Optimization at Different Temperatures

Synchronization-optimized networks were obtained by running the evolutionary optimization. In this process, the order parameter was progressively increasing until a stationary state has been achieved. When using replicas with the larger inverse temperature $\beta$, the larger values of the order parameter could be reached, although the optimization process was then slower. After the transients, statistical averaging of the order parameter over the ensemble with the Gibbs distribution has been performed, according to Eq. (4).

In Fig. 1(a), the averaged order parameter $\bar{R}$ is displayed as a function of the connectivity $p$ for several different inverse temperature $\beta$. The solid circle symbols show the averaged order parameter corresponding to the replica with $\beta_0 = 0$, i.e., for an infinitely high temperature. We see that the averaged order parameter increases with the network connectivity $p$ even if the networks are produced by only random rewiring. The open circles show the average order parameters for the ensemble corresponding to the replicas with the lowest inverse temperature $\beta_M$. Generally, greater order parameters are obtained by running evolution at higher inverse temperatures $\beta$ at any network connectivity $p$. At each connectivity $p$, the order parameter is gradually increased with increasing $\beta$ and is approximately saturated at $\beta_M$. This means that, even if one further increases $\beta$, only slight improvements of the averaged order parameter can be expected. Thus, the networks sampled by the replica with the largest inverse temperature $\beta_M$ are already yielding a representative optimal ensemble.

Figure 1(b) shows, depending on the network connectivity $p$, the ratio $R_{\beta_0}/R_{\beta_0}$ of the averaged order parameters sampled by the optimal network ensemble with $\beta_M$ to those obtained for the ensemble with purely random rewiring. Since there is no room for the improvement of the order parameter when the number of links is small, the ratio tends to unity as the connectivity $p$ is decreased. On the other hand, when $p = 1$, global coupling is realized, for which, under the chosen coupling strength, full synchronization occurs. As evidenced by this figure, the difference between the syn-
chronization capacities of the optimized and random networks is most pronounced at the intermediate connectiv-
ties $p$.

3.2. Architectures of Synchronization-Optimized Networks

When the connectivity $p$ is small, typical structure of synchronization-optimized networks usually represent chain fragments. At a higher connectivity, the network becomes more complexly organized.

To statistically characterize the architecture of constructed networks, ensemble averages of their adjacency matrices over the Gibbs ensemble, i.e.,

$$
\overline{w}_p = \sum_w w \exp(\beta R(w))/Z(\beta),
$$

for different connectivities $p$ were computed for $\beta = \beta_M$, as shown in Fig. 2. Clearly, the optimal network structure is changing with the number of links. When the number of links is small, the elements of the mean adjacency matrix, obtained by averaging over many realization from the synchronization-optimized ensemble, are large near the diagonal. Hence, elements with close natural frequency tend to connect and form a chain fragment. Moreover, oscillators with the natural frequencies near the center of the interval are often connected. Increasing the number of links, the network becomes more complicated and off-diagonal elements begin to dominate instead. The network with the larger $p$ tends to have interlaced structures, seen in Figs. 2(b) and 5(c), where the oscillators with roughly opposite natural frequencies are coupled.

4. Summary

We have designed synchronization-optimized networks with a fixed number of links for a heterogeneous oscillator population. This has been done by using the Markov chain stochastic Monte Carlo method complemented by the replica exchange algorithm. A transition from the linear to bipartite-like networks has been found under increasing the number of links. At low connectivity, synchronization-optimized networks typically represent small chains connecting oscillators with close natural frequencies. As the number of links increases, the networks become interlaced and oscillators with opposite natural frequencies tend to be connected. Therefore, synchronization-optimized network begin to resemble bipartite graphs.

Acknowledgments

This study has been partially supported by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Scientific Research (Grant No. 21540376) and the Volkswagen Foundation (Germany).

References

An efficient algorithm for the evaluation of Master Stability Function in networks of coupled oscillators

Marco Righero, Fernando Corinto and Mario Biey

Electronics Dept., Politecnico di Torino, Corso Duca degli Abruzzi 24, I-10129, Torino, Italy
Email: {marco.righero, fernando.corinto, mario.biey}@polito.it

Abstract—An efficient methodology to study stable in-phase synchronization in networks of identical nonlinear oscillators is proposed. The problem of investigating synchronization properties is reduced to an eigenvalue problem by means of the joint application of the Master Stability Function and the Harmonic Balance technique. The proposed method permits to achieve huge reduction in computational time respect to traditional time-domain approaches. In addition, such method can be extended to study synchronization patterns in networks of nonlinear oscillators described by differential-algebraic equations.

1. Introduction

The existence of (locally) stable synchronous states in networks of coupled nonlinear oscillators is mainly shown by computing the spectrum of Lyapunov Exponents (LE). Unfortunately, specially when dealing with continuous time high-order systems, such a computation requires long CPU time and may show numerical instabilities [1–3].

In 1998, Pecora and Carroll [4] proposed a technique, subject to some constraints, to simplify this task. The problem of synchrony detection was split in two parts: one related to network topology and the other requesting the computation of LE of the single (generally low-order) uncoupled oscillator. This second part requires the evaluation of the so called Master Stability Function (MSF) [4]. Even if in this case we have to deal with low order nonlinear oscillators, the computational effort remains important because steady-state periodic solutions of the uncoupled oscillator are required. Such solutions may be obtained by means of time-domain methods by discarding transient behavior. On the other hand spectral methods (for instance Harmonic Balance - (HB)) provide an accurate approximation of steady-state periodic oscillations in nonlinear oscillators [5].

The main aim of this paper is to present an efficient method, based on the joint use of HB and MSF techniques, in order to evaluate synchronization properties in networks composed of coupled identical nonlinear oscillators with at least a stable limit cycle. In particular, the HB-based method allows one to identify limit cycles and, beyond the reduction in CPU time, makes it possible to investigate nonlinear oscillators described by implicit differential equations [6].

The manuscript is organized as follows. In Section II, a brief summary of the MSF approach is presented and in Section III the new algorithm is described in details. The impressive reduction in CPU time is presented through examples in Section IV. Some conclusions are drawn at the end of the paper.

2. The Master Stability Function

The Master Stability Function permits to study synchronization conditions for networks of coupled nonlinear systems [4]. We summarize the main ideas to point out how spectral methods can be successfully used to conceive efficient algorithms for evaluating synchronization on limit cycles.

We consider networks of $N$ cells described by the model ($n = 1, \ldots, N$)

$$\dot{x}_n = f(x_n) + \epsilon \sum_{n'=1}^{N} A_{nn'} H(x_{n'} - x_n), \quad (1)$$

where $x_n \in \mathbb{R}^D$ is the $D$-dimensional state of the $n$th cell, $f : \mathbb{R}^D \to \mathbb{R}^D$ accounts for the nonlinear self dynamics, $\epsilon \in \mathbb{R}$ is the global coupling strength, $A \in \{0, 1\}^{N \times N}$ is the adjacency matrix describing the coupling among cells (so that $A_{nn'} = 1$ if cell $n$ is connected to cell $n'$ and $A_{nn'} = 0$ otherwise), and $H \in \mathbb{R}^{D \times D}$ select which variables are used in the coupling. Even if Eq. (1) gives a more explicit idea of how $x_{n'}$ is linked to $x_n$, for the following calculation it is better to use the laplacian matrix $L$ associated to $A$ (namely $L_{nn'} = A_{nn'}$ when $n \neq n'$, $L_{nn} = -\sum_{n'=1}^{N} A_{nn'}$) to describe the system

$$\dot{x}_n = f(x_n) + \epsilon \sum_{n'=1}^{N} L_{nn'} H x_{n'}. \quad (2)$$

We are interested in studying conditions on the coupling strength $\epsilon$ and on the network topology described by $L$ so that the synchronous manifold $x_1 = x_2 =$
\[ x_n = \ldots x_N \text{ is stable} \] It is worth noting that the synchronous manifold corresponds to the in-phase periodic oscillation, i.e. there are zero phase shifts among all oscillators.

Denoting by \( x \) the common dynamics, we consider the variational equations associated to Eq. (2)

\[ \dot{\delta}_n = Df(x)\delta_n + \epsilon \sum_{n'=1}^N L_{nn'} H \delta_{n'}, \]

where \( \delta_n \) is an infinitesimal increment with respect to \( x \) of the \( n \)-th cell and \( Df \) is the Jacobian of \( f \). By considering the linear transformation which diagonalizes \( \epsilon L \), we get \( N \) uncoupled equations

\[ \dot{\delta}_n = (Df(x) + (\sigma_n + i \beta_n)H)\delta_n, \]

where \( \sigma_1 + i \beta_1, \ldots, \sigma_N + i \beta_N \) are the \( N \) eigenvalues of \( \epsilon L \), considered with their multiplicities. The stability of the synchronous manifold with respect to the direction identified by the eigenvalue \( \sigma_n + i \beta_n \) is then determined by evaluating the maximum Lyapunov exponent of Eq. (4) on the synchronous manifold \( \dot{x} = f(x) \). If we consider \( \sigma_n + i \beta_n \) as a parameter in \( C \), we can perform the same calculation for every value of \( \sigma \) and \( \beta \) in a certain interval, to get a broad sweep of possible configurations. The result is the Master Stability Function (MSF) \( \Lambda(\sigma, \beta) \). If, given \( \epsilon \) and \( L \), we have that \( \Lambda(\sigma_n, \beta_n) < 0 \) for each eigenvalue of \( \epsilon L \), then the considered configuration and coupling strength give a (locally) stable synchronous manifold.

In the following we focus on symmetric coupling, so that all the eigenvalues of \( \epsilon L \) have vanishing imaginary part. It follows that we are interested in computing \( \Lambda(\sigma, 0) = \Lambda(\sigma) \). In addition, ranking the eigenvalues in decreasing order, we have that \( 0 = \sigma_1 > \sigma_2 \geq \ldots \geq \sigma_N \): the first eigenvalue is 0, as we consider laplacian matrices, and the second is strictly negative, since the corresponding graph is connected.

### 3. Efficient algorithm based on HB and MSF

The MSF is often evaluated using a numerical algorithm that computes the maximum Lyapunov exponent of Eq. (4) on the synchronous manifold. This approach usually requests a numerical integration of the differential equations, discarding the transient to reach the synchronous state, and further time to have the algorithm to estimate the Lyapunov exponents at the convergence. In [7] the HB technique was successfully employed to study bifurcation phenomena in limit cycle systems. In this Section, we present an alternative efficient algorithm, based on MSF and HB, using the numerical ideas developed in [6].

Before describing the algorithm, we should briefly recall how HB technique is used for analyzing limit cycles and their Lyapunov exponents.

Given a scalar function \( x(t) \) which is \( T \)-periodic, it can be expressed, in an approximated form, as a truncated Fourier series

\[ x(t) \simeq a_0 + \sum_{k=1}^K \left( a_k \cos(k\omega t) + b_k \sin(k\omega t) \right), \]

where \( a_0, a_k, \) and \( b_k \) are the Fourier coefficients and \( \omega = 2\pi/T \). Using \( M = 2K + 1 \) equally spaced time samples \( x(t_m) \) in \([0, T]\), with \( t_m = mT/M \), \( m = 1, \ldots, M \), we can link the Fourier coefficients and the time samples \( x(t_m) \) using an appropriate matrix \( \Gamma \) with inverse

\[ \Gamma^{-1} = \begin{pmatrix} 1 & \gamma_{11}^C & \gamma_{12}^C & \cdots & \gamma_{1K}^C \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \gamma_{H1}^S & \gamma_{H2}^S & \cdots & \gamma_{HK}^S \end{pmatrix} \]

whose entries are given by

\[ \gamma_{p,q}^C = \cos \left( \frac{q\pi p}{2K+1} \right), \quad \gamma_{p,q}^S = \sin \left( \frac{q\pi p}{2K+1} \right). \]

We then have, \( X^F = \Gamma X \) where (denoting transposition by an apex ') \( X = [x(t_1), \ldots, x(t_T)]' \), and \( X^F = [a_0, a_1, b_1, \ldots, a_K, b_K]' \). We can link the Fourier coefficient of the time derivative \( \dot{x}(t) \) to the ones of \( x(t) \) as \( \dot{X}^F = \omega \Omega X^F \), where \( \Omega \) is a \( M \times M \) matrix full of 0 except \( \Omega_{2k,2k+1} = k \) and \( \Omega_{2k+1,2k} = -k \), for \( k = 1, \ldots, K \). As described in [7], this formalism can be used to look for the Fourier coefficients of limit cycles for a system \( \dot{x} = f(x) \).

We recall now that, given a system of dimension \( D \)

\[ \dot{\delta} = B(t)\delta, \]

with \( B(t) \) a \( T \)-periodic matrix, the corresponding \( D \)-Floquet multipliers (FMs) \( \lambda_d = e^{\mu_d T} \) are such that \( u_d(t) e^{\mu_d t} \), with \( u_d(t) \) \( T \)-periodic, are \( D \) linearly independent solutions of Eq. (8). The Lyapunov (or characteristic) Exponents are \( \mu_d \). If \( B(t) \) in Eq. (8) is the Jacobian matrix of the vector field \( f \) on a limit cycle solution, then the FMs carry information about the stability of the limit cycle. Following [6], we can express the FMs in the HB setting as an eigenvalue problem:

\[ (\Gamma_D B_{M} \Gamma_D^{-1} - \omega \Omega_{M}) U^F = \mu_d U, \]

\[ -574 - \]
where $B_M$ is a $DM \times DM$ matrix constructed expanding each element of $B$ in a diagonal block of time samples $B(t_1), \ldots, B(t_M)$, and $\Omega_M$ is a $DM \times DM$ block-diagonal matrix built up of $D$ copies of $\Omega$. The solution of Eq. (9) gives $DM$ different eigenvalues, which correspond to a subset of the infinite $\mu_d$ determining $D$ independent FM $e^{\mu_d T}$. These eigenvalues are distributed along $D$ vertical lines in the complex plane (see Fig. 1 for an example with $D = 3$ and $K = 5$). As noted in [6], to get the more stable results, we should look for the ones with smaller imaginary part.

The proposed algorithm to determine the MSF $\Lambda(\sigma)$ for $\sigma \in [\sigma_+, \sigma_-]$ on a limit cycle is then made of four steps: (1) given the uncoupled oscillator $\dot{x} = f(x)$, find an accurate approximation of the limit cycle of interest $X^F$; (2) for $\sigma \in [\sigma_+, \sigma_-]$ solve Eq. (9) with $B_M$ constructed using $B$ given by $DF + \sigma H$ evaluated on the limit cycle $X^F$, obtaining $DM$ eigenvalues; (3) among them, select the $D$ eigenvalues with smaller imaginary part $\mu_1, \ldots, \mu_D$; (4) $\Lambda(\sigma) = \max\{\Re(\mu_1), \ldots, \Re(\mu_D)\}$.

4. Numerical results

In order to verify the accuracy of the proposed algorithm, we study the stability of synchronous states in two networks composed of simple and well-known oscillators: Van der Pol and Chua oscillators.

To this end, the MSF for Van der Pol and Chua oscillators are derived according to the proposed algorithm. Then the results are compared to those obtained by means of time-domain methods.

Van der Pol oscillator is described by the following differential equations

\[
\begin{align*}
\dot{x}_1 &= \frac{1}{C}(x_1 - kx_2 - x_2), \\
\dot{x}_2 &= \frac{1}{L}x_1.
\end{align*}
\]

Assuming $C = 1, L = 4/5, k = 1$ we get a smooth limit cycle which can be approximated using $K = 5$ harmonics. Fig. 2 shows the approximation superimposed to the numerical integration in the time-domain. Chua’s circuit (oscillator) can be described by the following set of normalized differential equations

\[
\begin{align*}
\dot{x}_1 &= \alpha[-x_1 + x_2 - n(x_1)], \\
\dot{x}_2 &= x_1 - x_2 + x_3, \\
\dot{x}_3 &= -\beta x_2.
\end{align*}
\]

We assume $\alpha = 8, \beta = 15, n(x_1) = -8/7x_1 + 4/63x_1^3$ and we focus on one of the two asymmetric limit cycles [7], that can be accurately approximated using again $K = 5$ harmonics.

Besides $K = 5$, we also consider different choices of $K$ (i.e. $K = 1$ and $K = 3$) to check how the algorithm is influenced by the number of harmonics. Steps (2)–(4) of the algorithm outlined in Sec. 3 are executed with $\sigma \in [-20, 0]$. Finally, the results are compared with those obtained by applying standard MSF approach in the time-domain.

Fig. 3 (Van der Pol oscillator) and Fig. 4 (Chua’s oscillator) make clear that, in both cases MSFs are practically coincident with those derived in time-domain even if a few number of harmonics ($K = 5$) are exploited. It is also important to note that even $K = 1$ permits to identify accurately the synchronization region $\Lambda(\sigma) < 0$. The low number of harmonics necessary for a satisfactory approximation is mainly due to
the properties of the oscillators chosen to test the proposed algorithm. However, it gives remarkable results also in the case of nonlinear oscillators having limit cycles whose approximation requires several harmonics.

The main advantages of the proposed algorithm are its numerical stability and short computation time. In the considered example, the reduction of CPU time $^2$ is about $90\%$ compared to standard time-domain techniques [2, 3].

5. Conclusion

We have proposed an efficient method to study condition for stable local synchronization in networks of identical nonlinear oscillators exploiting the MSF approach in the frequency domain. Once limit cycles of uncoupled oscillators are approximated by means of HB technique, an eigenvalue problem is solved to study stability. The proposed method presents huge advantages in terms of computational time and can also be extended to oscillators described by implicit differential equations.

6. Acknowledgments

This work was partially supported by the CRT Foundation, under the project no. 2009.0570, by the Istituto Superiore Mario Boella and the regional government of Piedmont.

$^2$The comparison is based on several numerical simulations run on standard desktop under MatLab.

References


Linearity and nonlinearity within recurrence plots

Yoshito Hirata and Kazuyuki Aihara
Institute of Industrial Science, The University of Tokyo, Tokyo 153-8505, Japan
Email: yoshito@sat.t.u-tokyo.ac.jp

Abstract—Recurrence plots are widely used for analysing time series generated from complex systems. While it is argued that the tau-recurrence rate is similar to the autocorrelation, it is still not clear which characteristics within recurrence plots represent their linear and nonlinear properties. In this paper, we observe how fast the number of diagonal lines decreases when their length increases. We found that in linear stochastic systems, the logarithm of the number scales exactly, while in typical nonlinear deterministic systems, the decreasing rate of the number tends to approach from the above. We illustrate our finding with numerical examples.

1. Introduction

Nonlinear time series analysis [1, 2, 3] has been developed for the last three decades. Among various methods, recurrence plots [4, 5] have been intensively investigated for the last two decades.

Recurrence plots visualize given time series. Recurrence plots are two-dimensional graphs both axes of which represent time. For a pair of two times, if the corresponding states are similar, then one plots a point at the corresponding place. Otherwise, one plots nothing. Due to the simple definition, we can extend recurrence plots to marked point processes [6]. Recent trends in recurrence plots are their quantifications, which are called Recurrence Quantification Analysis (RQA) [5]. Quantities that characterize vertical and horizontal lines in recurrence plots were proposed. Using these quantities, the properties of given time series can be characterized from various viewpoints.

Although recurrence plots were studied well, it is still not clear which characteristics represent linear and nonlinear features of given time series within a recurrence plot. There is a paper by Zbilut and Marwan [7] that showed the tau-recurrence rate is similar to the autocorrelation, which characterizes linear stochastic systems.

In this paper, we show that the distribution of the diagonal line length of recurrence plots can distinguish linear stochastic systems from nonlinear deterministic systems. In linear stochastic systems, the logarithm of the number of the diagonal line length scales even if the length of the diagonal line is short. On the other hand, in typical nonlinear deterministic systems, the decreasing rate of the number tends to converge from the above.

The rest of the paper is organized in the following way. In Section 2, we introduce recurrence plots and their quantifications. In Section 3, we present numerical examples showing that the distribution of diagonal line length is different between linear stochastic systems and nonlinear deterministic systems. In Section 4, we apply our finding to a dataset of squid giant axon. In Section 5, we conclude the paper.

2. Recurrence plots and their quantifications (RQA)

Let \( x_i \in \mathbb{R}^m \) \( (i = 1, 2, \ldots, n) \) be a point of a time series at time \( i \). Let \( d : \mathbb{R}^m \times \mathbb{R}^m \rightarrow \mathbb{R} \) be a distance function. Here we use the Euclidean distance. A recurrence plot is defined as

\[
R(i, j) = \begin{cases} 
1, & \text{if } d(x_i, x_j) < \varepsilon, \\
0, & \text{otherwise}. 
\end{cases}
\]

When \( R(i, j) = 1 \), a point is plotted at \((i, j)\). If \( R(i, j) = 0 \), nothing is plotted at \((i, j)\). It is demonstrated that from recurrence plots one can reproduce rough shapes of the original time series [8, 9]. In fact, it was shown that if two recurrence plots are the same, then the corresponding dynamics are equivalent [10]. By using recurrence plots, driving forces affecting observed elements can be reconstructed [14]. Therefore, recurrence plots are good methods for describing and analyzing time series.

Recently recurrence plots are applied to identify network topology from time series [11, 12, 13].

A good point of recurrence plots is that we can understand the characteristics of dynamics from the patterns of the plotted points. For example, from white noise, we can obtain a recurrence plot in which points spread uniformly randomly. From a periodic time series, we can obtain a recurrence plot in which diagonal lines are running in parallel with an equal space. As for a time series generated from a deterministic system, the corresponding recurrence plot contains short diagonal segments of lines. Therefore, diagonal lines characterize the determinism of given time series.

On the other hand, vertical and horizontal lines appear when a point in a time series tends to stay in a particular state in phase space.

There are various quantities characterizing plotted patterns of points in recurrence plots. Analysis calculating these quantities is called Recurrence Quantification Analysis (RQA). The quantities of RQA can be classified into two groups [5]: One of the groups characterizes the distribution of diagonal lines. Let \( c(l) \) be the number of diagonal
lines with at least length \( l \) in a recurrence plot \( R \). This \( c(l) \) can be calculated as

\[
c(l) = \frac{1}{n} \sum_{i,j=1}^{l-1} R(i + k, j + k).
\]

It is known that from the asymptotic decreasing rate of \( c(l) \), one can estimate the correlation entropy [15]. In the next section, we look at the distribution of diagonal line length \( c(l) \) closely.

### 3. Distribution of diagonal line length

We investigate the distribution of diagonal line length for the following four models.

The first model is the first order autoregressive linear model:

\[
x(t) = -0.7x(t - 1) + \eta_t,
\]

where \( \eta_t \) is the Gaussian distribution of the mean 0 and the standard deviation 1.

The second model is the 100th order autoregressive linear model:

\[
x(t) = \sum_{i=1}^{100} a_i x(t - i) + \eta_t.
\]

We chose \( a_i \) from the Gaussian distribution of the mean 0 and the standard deviation 0.05 so that the system becomes stable.

The third model is the Ikeda map:

\[
\begin{pmatrix}
x(t) \\
y(t) \\
\theta(t)
\end{pmatrix} =
\begin{pmatrix}
1 + 0.9(x(t - 1)\cos(\theta(t)) - y(t - 1)\sin(\theta(t))) \\
0.9(x(t - 1)\sin(\theta(t)) + y(t - 1)\cos(\theta(t))) \\
0.4 - \frac{6.0}{1 + x^2 + y^2}
\end{pmatrix}.
\]

We use \( x(t) \) as a scalar time series.

The fourth model is the Rössler model:

\[
\frac{d}{dt}
\begin{pmatrix}
u \\
w
\end{pmatrix} =
\begin{pmatrix}
-(v + w) \\
u + 0.36v \\
0.4 + w(u - 4.5)
\end{pmatrix}.
\]

We used \( u(t) \) sampled every 1 as a scalar time series.

We used the absolute distance for calculating distances. We also chose thresholds so that the recurrence rates became 0.05.

The results are shown in Fig. 1. In the case of linear stochastic systems, the logarithm of the number of the diagonal line length scales almost exactly, while in the nonlinear deterministic systems, the decreasing rate of the number tends to converge from the above.

We showed analytically that in stationary ergodic linear stochastic systems, the logarithm of the number of the diagonal line length scales even if the line length is short. On the other hand, in nonlinear deterministic systems whose dimension is more than 1, due to the effects of false nearest neighbors [16], the decreasing rate of the number tends to converge from the above.

We obtained a recurrence plot of the dataset using a threshold such that the recurrence rate is 0.05. The diagonal line length distribution of this dataset is shown in Fig. 2. Since the length of time series is short, we cannot obtained the error bars by using different time series as we did in Fig. 1. Therefore, instead, we obtained the error bars by using bootstrapping the diagonal line length distribution [19]. We can see that the distribution of the diagonal line length is different from that of linear stochastic systems.

### 5. Conclusion

In this paper, we have shown that the diagonal line length distribution is different from linear stochastic systems with typical nonlinear deterministic systems. In linear stochastic systems, the logarithm of the number of diagonal line length scales even if the line length is short. On the other hand, in nonlinear deterministic systems, the decreasing rate of the number of diagonal line length tends to converge from the above. As for the actual squid giant axon dataset, we found that it is different from linear stochastic systems. The proposed method is simpler than the other existing methods for testing nonlinearity such as ones using time-reversibility [20] and the Fourier transform-based surrogate data [21, 22]. Therefore, we believe that it will contribute to advancements of nonlinear science.
Figure 1: The distribution of number of diagonal line length. (a) the first order autoregressive linear model. (b) the 100th order autoregressive linear model. (c) the Ikeda map. (d) the Rossler model. The error bars were obtained by using 20 different realizations of time series generated from different initial conditions. The broken line was obtained by fitting the means of the logarithms of the numbers for lengths 1 and 2.
Acknowledgement

This study is partially supported by A Grant in Aid for Young Scientists (B), No. 21700249, from the Japanese Ministry of Education, Culture, Sports, Science and Technology to Y. H..

References


Transformation of growing networks to time series and its nonlinear time series analysis

Yutaka Shimada†, Yuta Haraguchi‡, and Tohru Ikeguchi†‡

†Graduate School of Science and Engineering, Saitama University
255 Shimo-ohkubo, Sakura-ku, Saitama, 338-8570 Japan
‡Saitama University Brain Science Institute
Email: sima@nls.ics.saitama-u.ac.jp, haraguchi@nls.ics.saitama-u.ac.jp, tohru@mail.saitama-u.ac.jp

Abstract—In this paper, we propose a novel method to generate a time series from a complex network to identify an underlying evolution process behind the complex network. Based on the adjacency information between two nodes, we first arrange the nodes in the complex network in the Euclidean space, and show that their arrangements clearly correspond to their network structure. We further show that if the complex network is a growing network, namely complex network with temporal evolutions, we can regard the obtained arrangements as a time series. Thus, we can transform the complex network to a time series and analyze the time series using the methods based on the nonlinear dynamical system theory.

1. Introduction

In the real world, there exist several complex networks, for example, human relationships, synaptic connections in neural systems, the world-wide-web, ecological food webs, and so on. These networks have complex structures and recent researches on real complex networks have been clarified that some of these structures are led by temporal evolutions of the networks [1, 2, 3]. In such networks, nodes are added to the network at each temporal step, then the network grows with its complex structure formed. Thus, it is an important issue to clarify how the networks evolve.

Here, we propose a novel method which elucidate an underlying evolution process of growing complex networks. In the proposed method, we use geometric arrangement of nodes and temporal information when the nodes are added to the network. First, the nodes are arranged in the Euclidean space based on their adjacency relationships [5]. In the type-1 method, the distances between non-adjacent nodes are determined by the shortest distances in the Euclidean space based on their adjacency relationships [5]. In the type-1 method, the distances are given as correctly as possible. Here, our method is originated from the method proposed by Hirata et al. [5] in which an original time series can be reproduced only from its recurrence plot. We extend this concept to the complex network analysis. Using the obtained arrangements and temporal information, we can transform the network to a time series. Our method consists of the following three steps:

1. Generating a distance matrix.
2. Applying the CMDS to the distance matrix.
3. Transforming the network to a time series by using the arrangement of nodes and the temporal information.

2.1. Generation of distance matrix

To obtain an arrangement of nodes, we first need to define distances between any two nodes. Here, we introduce two methods for generating the distance matrix $D = (d_{ij})$.

The type-1 method is originally proposed by Hirata et al. that determines the distances between two nodes based on their adjacency relationships [5]. In the type-1 method, the distance $d_{ij}$ between two adjacent nodes $i$ and $j$ is determined by the following equation:

$$d_{ij} = 1 - \frac{|G_i \cap G_j|}{|G_i \cup G_j|}, \tag{1}$$

where $G_i$ is the set of labels of adjacent nodes of the node $i$, $|G|$ is the number of elements in the set $G$, and $\cap$ and $\cup$ are the union and intersection of two sets. The distances between non-adjacent nodes are determined by the shortest
path length by using the distances between the adjacent two nodes.

The type-2 method uses the adjacency information of each nodes directly [6]. If the $i$th and $j$th nodes are connected, these nodes are similar and we set $d_{ij}$ to unity but if two nodes are not connected, these two nodes are dissimilar and we set $d_{ij}$ to zero. If $i = j$, we set $d_{ij}$ to zero. Then, in the type-2 method, the distance matrix has only three values.

2.2. Classical multidimensional scaling

Next, we apply the CMDS to the distance matrix generated from the complex networks. In the CMDS, there are mainly two procedures[4]: the centering of matrix $A = \frac{1}{2}D^{(2)} = (-d_{ij}^2)/2$ by the centering matrix $J$ and its eigenvector decomposition, namely $AJ = V\Lambda$, where $\Lambda = $\text{diag}(\lambda_1, \ldots, \lambda_d)$, $V = (v_1 \cdots v_d)$, and $A_i$ and $v_i$ are the $i$th eigenvalue and the $i$th eigenvector. Here, let the coordinate value of the $i$th node be $x_i$ and $X = (x_1 \cdots x_N)$. From the relationship between the innerproduct and the distance[4], $X = V\Lambda^{1/2}$, where $\Lambda^{1/2} = \text{diag}(\lambda_1^{1/2}, \ldots, \lambda_d^{1/2})$. Then the dimension, $d$, of the Euclidean space in which nodes are arranged is the rank of $X$. By the above mentioned procedure, we can obtain the coordinate values of the nodes $X$ only from the distance matrix $D$.

2.3. Complex networks to time series

To transform the network to a time series, we use the obtained arrangements and the temporal information when each node is added to the network. We consider that this temporal order is described as labels of the nodes. For example, if a node is the first node added to the network, the label of the node is one, if the second, the label is two, if the third, the label is three, and so on. Thus, considering the coordinate values of each node as the amplitude of the time series and the labels as the temporal order, we can transform the complex networks to the time series. In this paper, for the sake of simplisity, we consider that only one node is added to the network at each temporal step.

3. Experimental settings

To evaluate the validity of our method, we conduct two numerical experiments. First, we confirm how does the nodes in the networks are arranged in the Euclidean space. We generated the networks by using three models: the Watts and Strogatz (WS) model [7], the Barabási and Albert (BA) model [1], and the Dorogovtsev, Goltsev, and Mendes (DGM) model [8].

In the WS model, introducing the link-rewiring probability $p$ from zero to unity, we varied the network structure from the regular network to random networks. In the numerical simulations, we generated the networks with the WS model, which have 1,000 nodes with the average degree of ten. We set the rewiring probability $p$ to zero in the regular, 0.01 in the small world, and unity in the random networks. In the BA model, the network starts with a complete graph consist of $m_0$ nodes and only one node with $l$ links is added to the network at each temporal step. The additional nodes are easy to be connected to nodes with high degree. We set both $m_0$ and $l$ to four and generate a network with 1,000 nodes in the numerical simulations. The DGM model generates the network having a pseudo-fractal structures [8]. In the numerical simulations, we generate a network with 1,095 nodes by the DGM model.

Next, we transform the networks to time series and analyze them. To transform the networks to time series, we need temporal information. The BA model satisfies this condition, because each node is added to the network at temporal step and thereby the nodes have the temporal order. Here, modifying the DGM model, we can generate a model which deterministically grows and satisfies the condition, namely only one nodes is added to the network at each temporal step. In the DGM model, an initial graph $G(t=0)$ consists of two nodes with one link. At each temporal step, one node for each link is added to the network. Thus, the number of nodes which are added to the network at time $t$ is $2 \cdot 3^{t-1} (t > 0)$. We modified the DGM model as follows:

1. Start with an initial graph consisted of two nodes and one link. Here, the label of one node is zero, that of another node is one, and that of the link is zero. We set time $t$ to zero. Let the link be $l_1$, which has the smallest label.

2. Add one new node with two links to the network and increase time $t$ by one. This new node is connected to two nodes at both ends of the link $l_1$.

3. Update the label of the link by $l_1 \leftarrow 3t - 2$.

4. Set labels of the new additional node and two links. We set the label of the new node to $t + 1$. We set one of two links which is connected to the node having the smaller label to $l_1 + 1$ and the larger label to $l_1 + 2$.

5. Set the link $l_1$ to the link with the smallest label.

6. Repeat steps 2–5.

Because the modified DGM model deterministically grows with the pseudo-fractal structure formed in a bounded region, its evolution process might show a chaotic-like behavior. On the other hand, the BA model grows based on the stochastic rules. Thus, these two networks have different evolution processes. In the numerical simulations, we generate growing networks from the BA and the modified DGM models and transform these growing networks to time series. By using the time series prediction, we elucidate the difference of the evolution processes between them[9].

To determine the evolution processes of networks by the prediction, we use the first eigenvector obtained from the
networks and reconstruct it in the \( s \)-dimensional state space by the time delay coordinate[10]. In the numerical simulations, we set \( s \) to eight and determine the temporal delay \( \tau \) by mutual information [11].

Let \( x(t) \) be a point in the \( s \)-dimensional reconstructed state space at time \( t \). In the prediction, we estimate the point \( \hat{x}(t + T_p) \), the \( T_p \) steps after \( x(t) \) by the following equation:

\[
\hat{x}(t + T_p) = \sum_{i=1}^{M} \exp(-d_i)x(m_i + T_p) = \sum_{i=1}^{M} \exp(-d_i)
\]

where \( x(m_i) \) is the \( i \)th nearest neighbor of the \( x(t) \), \( d_i \) is the Euclidean distance between \( x(t) \) and \( x(m_i) \), and \( M \) is the number of the nearest neighbors.

We use the first half of the time series as the database for the predictions and predict the last half. To evaluate the prediction accuracy, we investigate the correlation coefficient between the true time series, namely the last half of the time series, and its predicted time series.

4. Results and discussions

Figure 1 shows examples of the arrangements of nodes in the networks. To describe the arrangements in the plane, we use the two eigenvectors with the first and second largest eigenvalues. Figures 1(a), (b), and (c) are the arrangements for the regular, small world, and random networks generated from the WS model. Figure 1(d) is the arrangements for the scale free network generated by the BA model. Figure 1(e) is the arrangements for the scale free network generated by the DGM model. In Fig. 1, the left column shows the results for the type-1 method, the right column shows those for the type-2 method.

From Figure 1, in both the type-1 and type-2 methods, we can see clear correspondences of the arrangements to the network structures; the arrangement of the regular network is regular, that of the random network is random, and that of the small world is in between them (Figs. 1(a), (b), and (c)). As shown in the random network (Fig. 1(c)), the network generated from the BA model also has the random arrangement because each additional node randomly connects to other nodes and the connections depend on the probability determined by the degree of each node. Because the network generated from the DGM model has the pseudo-fractal structures[8], its arrangements shows a characteristic structure.

Here, the significant differences between the type-1 and the type-2 methods are shown in the scale free network. In the results for the type-2 method, a few points are arranged far from other points (Figs. 1(d) and (e) in the right column). These points correspond to hub nodes which have significantly higher degree than the other nodes. In the type-2 method, because the adjacency relationship is directly used, the differences between the hub nodes and the other nodes are emphasized in their arrangements. On the other hand, in the arrangement by the type-1 method, the hub nodes and the other nodes are indistinguishable due to indirect use of the adjacency relationship (Figs. 1(d), (e) in the left column).

Figure 2 shows the prediction results for time series ob-
obtained from the scale-free networks generated from the BA and the modified DGM models. Figures 2(a) and (b) are the results for the type-1 and type-2 method, and both methods show the same tendency. From Fig. 2, the correlation coefficients for the BA model are low for all prediction steps $T_p$. On the other hand, those of the modified DGM model are also low in the large $T_p$, but high in the small $T_p$. These tendencies indicate that the time series obtained from the BA model might be random and that from the modified DGM model might not be random but chaotic. These results imply that the time series obtained from the networks include the information of evolution processes of the original networks, and our method can clarify the underlying information of evolutions of the complex networks by only using static network structures and the temporal information.

5. Conclusion

In this paper, we proposed a novel method which elucidate underlying evolution processes of growing complex networks. In the proposed method, using geometric arrangements of nodes and temporal information when the nodes are added to the network, we can transform the networks to time series.

In numerical simulations, we evaluate the proposed method with mathematical models. In addition, to generate the network which deterministically grows, we modified the model proposed by the Dorogovtsev et al. We first showed that the arrangements of nodes in the complex networks clearly correspond to their network structures, and further showed that using the obtained arrangements and the temporal information, we can determine the evolution processes of the complex networks generated by the mathematical models.

The research of Y.S. was supported by JSPS research follow (No.22·8148). The research of T.I. was partially supported by a Grant-in-Aid for Exploratory Research (No.20650032) from JSPS.

References

Recurrence based complex network analysis of cardiovascular variability data to predict pre-eclampsia

Norbert Marwan¹, Niels Wessel², Holger Stepan ³, and Jürgen Kurths ¹,²

1 Potsdam Institute for Climate Impact Research, P.O. Box 60 12 03, D-14412 Potsdam, Germany
2 Department of Physics, Humboldt-Universität zu Berlin, Robert-Koch-Platz 4, 10115 Berlin, Germany
3 Department of Obstetrics and Gynecology, University of Leipzig, Philipp-Rosenthalstraße 55, Leipzig 04103, Germany
Email: marwan@pik-potsdam.de

Abstract—Pre-eclampsia in pregnancy is a serious disease with high risk of fetal and maternal morbidity. The usual positive predictive value is 20–30%. Including cardiovascular variability, it has been recently shown that this predictive power can be improved.

Here we propose a novel approach for analysing time series of systolic and diastolic blood pressure as well as heart rate variability measured in the 20th week of gestation in order to predict pre-eclampsia. For this aim, we identify the recurrence matrix (calculated from time series) with the adjacency matrix of a complex network and apply measures for the characterisation of complex networks to this recurrence matrix. We demonstrate the potential of the complex network measures for a further improvement of the positive predictive value of pre-eclampsia.

1. Introduction

Pre-eclampsia is a serious disorder in pregnancy and is related with a remarkable maternal and neonatal morbidity and mortality affecting 3–5% of pregnant woman. Characteristic symptoms are sudden hypertension (rise in blood pressure) and proteinuria, connected to life-threatening cramps for mother and fetus, under-supply of the fetus and its growth retardation. A typical treatment is early delivery, which can be problematic because of the prematurity of the fetus. Therefore, an early prediction is highly desirable as it would allow for earlier countermeasures to control this pregnancy-specific disorder. The standard diagnostic tool is a Doppler sonography with a positive predictive accuracy of 20–30%.

Recent studies have suggested to include heart rate, systolic and diastolic blood pressure in order to improve the positive predictive accuracy [9, 23]. Here we propose to apply a complex network based recurrence analysis of the available time series in order to predict the disorder.

Recurrence is a fundamental property of dynamical systems and has been studied in theory and real world applications by various methods. One of the more recent approaches is the recurrence plot [12]. A recurrence plot (RP) is the graphical representation of a binary symmetric square matrix which encodes the times when two states are in close proximity (i.e. neighbours in phase space). Based on such a recurrence matrix, a large and diverse amount of information on the dynamics of the system can be extracted and statistically quantified (using recurrence quantification analysis, dynamical invariants, etc.). Meanwhile this technique has been the subject of much interest from various disciplines [10] and has been successfully applied to a number of areas: the detection of dynamical transitions [21] and synchronisation [18], the study of cardiovascular health conditions [13], and economical dynamics [2, 8], or to monitor mechanical behaviour and damages in engineering [14, 19]. It is important to emphasise that recurrence plot based techniques are useful for the analysis of short and non-stationary data, which often presents a critical issue when studying real world data.

Besides, complex networks are powerful tools for the analysis of complex and, in particular, spatially extended systems [3, 20, 24]. Local and global properties (statistical measures) of complex networks are helpful to understand complex interrelations and information flow between different components in extended systems, such as social, computer or neural networks [24], food webs, transportation networks, power grids [1], or even in the global climate system [6]. The basis of complex network analysis is the adjacency matrix, representing the links between the nodes of the network. Like the recurrence matrix, the adjacency matrix is also square, binary, and symmetric (in the case of an unweighted and undirected network).

In fact, the recurrence matrix and the adjacency matrix exhibit a strong analogy: a recurrence matrix represents neighbours in phase space and an adjacency matrix represents links in a network; both matrices embody a pair-wise test of all components (phase space vectors resp. nodes). Therefore, we identify the adjacency matrix of a complex network with the recurrence matrix of a dynamical system [7, 11], allowing us to apply measures of complex network theory to a RP in order to quantify the RP’s structure and the corresponding topology of the underlying phase space trajectory and to obtain additional information on the underlying process.

In the following, we will summarise the concept of the recurrence network analysis. Then we will apply this concept on a study of pre-eclampsia in order to improve its early preditcion.
2. Recurrence plots and complex networks

A recurrence plot (and, thus, the adjacency matrix) is a representation of recurrent states of a dynamical system in its m-dimensional phase space. It is a pair-wise test of all phase space vectors $\bar{x}_i$ ($i = 1, \ldots, N, \bar{x} \in \mathbb{R}^m$) among each other, whether or not they are close:

$$R_{i,j} = \Theta(||x_i - x_j|| - \delta_{i,j}),$$

with $\Theta(\cdot)$ being the Heaviside function, $\delta_{i,j}$ the Kronecker delta, and $\varepsilon$ a threshold for proximity [12]. The binary matrix $R$ contains the value one for all close pairs $||\bar{x}_i - \bar{x}_j|| < \varepsilon$. In terms of a complex network, each state vector in phase space represents one distinct node and closeness of two states (i.e., recurrence) represents a link.

A phase space trajectory can be reconstructed from a time series $\{u_t\}_{t=1}^N$ by time delay embedding [16]

$$\tilde{x}_i = (u_i, u_{i+\tau}, \ldots, u_{i+(m-1)\tau}),$$

where $m$ is the embedding dimension and $\tau$ is the delay.

Small-scale features in a RP can be observed in terms of diagonal and vertical lines. The presence of such lines reflects the dynamics of the system and is related to divergence (Lyapunov exponents) or intermittency [13, 17, 21]. Following a heuristic approach, a quantitative description of RPs based on these line structures was introduced and is known as recurrence quantification analysis (RQA) [10].

For example, slowly changing states, as occurring during laminar phases (intermittency), result in vertical line structures in the RP. Therefore, the distribution $P(v)$ of vertical line lengths $v$ can be used to quantify laminar phases occurring in a system. A useful RQA measure for quantifying such laminar phases is the fraction of recurrence points forming vertical structures of minimal length $v_{\text{min}}$,

$$\text{LAM} = \frac{\sum_{v=v_{\text{min}}}^{N} v P(v)}{\sum_{v=1}^{N} v P(v)},$$

which is called laminarity [12].

Following the idea of complex networks, the RP can also be heuristically quantified by complex network measures, that are well studied in literature [3, 7]. A complex network is invariant under permutation of the node order. Consequently, network measures will not directly reflect the dynamical properties of the system studied with RPs, but topological properties of the attractor.

For example, the averaged clustering coefficient $C = \sum_v C_v/N$ gives the probability that two neighbours (i.e. recurrences) of any state are also neighbours [24]. It is obtained as the average of the local clustering coefficient

$$C_v = \frac{\sum_{j=1}^{N} R_{i,j} R_{i,v} R_{v,j}}{k_v (k_v - 1)},$$

with $k_v = \sum_{j=1}^{N} R_{i,j}$ the number of neighbours of the state at time $v$. $C_v$ characterises localised higher-order spatial

correlations along the phase space trajectory. Specifically, high values of $C_v$ often coincide with dynamically invariant objects, such as periodic or unstable periodic orbits or, more generally, invariant manifolds [7], and low values correspond to higher variability or less regularity of the phase space trajectory.

We illustrate the potential of the discussed measures by an analysis of the logistic map

$$x_{i+1} = ax_i (1 - x_i),$$

In the analysed range of $a$, various dynamical regimes and transitions between them occur (Fig. 1A), e.g., accumulation points, periodic and chaotic states, band merging points, period doublings, inner and outer crises [4, 15, 22], and can be identified with RQA [13]. The RQA measure $\text{LAM}$ reveals intermittent dynamics (laminar regimes) by sudden increase of its values (Fig. 1B). The clustering coefficient indicates periodic dynamics by high values up to value one (Fig. 1C, D). As the system’s dynamics behaves rather regular also during intermittency, the laminar regimes are clearly revealed by $C_v$ as lines in the bifurcation diagram – coinciding with supertrack functions (Fig. 1D).

![Figure 1](image.png)
Heart rate variability

We construct the phase space vectors $\mathbf{x}(t)$ not by embedding but using all three measurements $\mathbf{x} = (H, S, D)^T$. In the next step we calculate the RP from this phase space trajectory by using a constant recurrence threshold of 0.8.

4. Results

The RPs of the pre-eclampsia and the control group do not visually differ much (Fig. 3). A complex network representation based on a force directed placement algorithm [5] already shows some slight differences, nevertheless difficult to reliably explain just by visual inspection (Fig. 4).

![Figure 3: Exemplary recurrence plots for (A) pre-eclampsia and (B) control group.](image)

![Figure 4: Exemplary complex networks representation for (A) pre-eclampsia and (B) control group (based on RPs shown in Fig. 3).](image)

Table 1: Median (standard deviation) of the cardiovascular measurements and the recurrence based measures for pre-eclampsia and control group as well as their corresponding $p$-value.

<table>
<thead>
<tr>
<th></th>
<th>Pre-eclampsia</th>
<th>Control</th>
<th>$p$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H$ (ms)</td>
<td>734.5 (±110.8)</td>
<td>760.5 (±111.7)</td>
<td>n.s.</td>
</tr>
<tr>
<td>$S$ (mmHg)</td>
<td>123.0 (±15.4)</td>
<td>123.5 (±20.0)</td>
<td>n.s.</td>
</tr>
<tr>
<td>$D$ (mmHg)</td>
<td>75.5 (±10.4)</td>
<td>66.6 (±13.9)</td>
<td>n.s.</td>
</tr>
<tr>
<td>$LAM$</td>
<td>0.80 (±0.10)</td>
<td>0.83 (±0.08)</td>
<td>n.s.</td>
</tr>
<tr>
<td>$C_r$</td>
<td>0.60 (±0.03)</td>
<td>0.62 (±0.04)</td>
<td>0.0015</td>
</tr>
</tbody>
</table>

The calculation of RQA and complex networks measures allows a quantitative characterisation of the recurrence structure of the cardiovascular data of the pre-eclampsia and control group (Tab. 1). We apply a Mann-Whitney U test for testing whether the medians of the distributions of the calculated measures of pre-eclampsia and control group differ. Considering the measurements of heart rate variability, systolic, or diastolic blood pressure alone, the medians are not able to distinguish between the two groups. The RQA measure $LAM$, which has been found to be a good candidate for the detection of cardiovascular disorders [13], is also not able to distinguish between the two groups (we have also tested other RQA parameters: only the recurrence rate $RR$ performed slightly better than $LAM$). In contrast, the clustering coefficient $C_r$ is able to discriminate the two groups of pre-eclampsia and control with good confidence. The positive predictive accuracy is 60% and the negative accuracy value is 80%. Thus, using the complex network approach we could improve the early prediction of pre-eclampsia from currently 20–30% to now 60%. For pre-eclampsia, $C_r$ is slightly lower than for the control group, suggesting a less regular cardiovascular oscillating regime related to the disorder. However, these are very first and preliminary results and are still subject of thorough research.

5. Conclusions

We have linked the recurrence matrix with the adjacency matrix of a complex network. This allows us to calculate complex network measures of a time series. As most of the complex network measures have no direct counterpart in recurrence quantification analysis, they give additional insights into the recurrence structure of dynamical systems. In general, this method allows to distinguish between different dynamical regimes and also to detect corresponding dynamical transitions.

By applying this novel approach to the cardiovascular data of pregnant women, we have been able to early predict the serious disorder of pre-eclampsia with a positive predictive value of 60% (standard method: 20–30%).
References


Standard complex network measures of recurrence-based phase space networks constructed from time series

Ruoxi Xiang, Jie Zhang and Michael Small

Department of Electronic and Information Engineering
The Hong Kong Polytechnic University, Hung Hom, Hong Kong, China
Email: Ms.Ruoxi-Xiang@polyu.edu.hk

Abstract—Recently, intensive efforts have been made to transform time series into networks since the networks constructed from time series can provide complementary information to that of the phase space of different dynamical systems. The recurrence-based phase space network, which is built by linking $k$ nearest neighbors of every point in the reconstructed phase space, can be used to specify different types of dynamics in terms of the motif ranking. Unlike network size with different scaling exponents and the degree distribution mimics the behavior of a discrete Gaussian distribution with different bandwidths from periodic to chaotic Rössler systems. These results indicate that network statistics may offer deeper understanding about the organization of the points in phase space. Finally, local network properties such as the vertex degree, the clustering coefficients and betweenness centrality are found to be sensitive to local stabilities of the orbits and contain complementary information.

1. Introduction

Recently, a framework for analyzing time series by constructing associated complex networks has attracted research interests [1–7]. Representing the time series through a corresponding network provides new methods to explore the properties of time series based on complex network theory which may lead to deep insights into the dynamical processes of a system. Many methods for transforming time series into networks have been suggested. A natural way for building networks from time series is by adding links between segments with some similarities, based on which, recurrence networks [3], cycle networks [2], correlation networks [5, 6] have been proposed. The visibility graph suggested by Lacasa et al. [4] which maps each points of the time series to nodes and link two nodes if a partial convexity constraint is fulfilled, is one of notable alternative methods.

Many research efforts have been devoted to recurrence networks based on the concept of recurrences in the phase space. Traditionally, to construct a recurrence network following the idea of a recurrence plot by Eckmann et al. [8], the time series is mapping into a set of points in the embedded phase space. Each point represents a node of the network, and two nodes are linked together when their phase space distance are smaller than a selected threshold $e$. By choosing an appropriate $e$, the local phase space properties can be best preserved [7]. In 2008, Xu et al. suggested constructing networks linking each node with its $k$ nearest neighbors [1] in the reconstructed phase space. At the microscopic level, the building blocks of the complex networks such as the network motifs reflect different local structure properties [9]. Based on the occurrence frequencies of the motif patterns, a so-called superfamily phenomenon is observed which can distinguish time series with different dynamics.

In this paper, we quantify these recurrence-based phase space networks (using the $k$ neighbor method of [1]) via the standard topological statistics besides the motif ranking. Such global and local statistics can provide new information about the phase space geometry within time series.

2. Quantitative assessment of recurrence-based phase space networks

When transforming time series into a recurrence-based phase space network representation, one node in the phase space is connected to its $k$ nearest neighbors and the associated network inherits structure of the distinct local phase space properties from different dynamical systems. The key question is what information is contained inside the network representation. When characterizing the structure of a complex network, we intend to look at its global properties such as the average path length, the global clustering coefficient and the degree distribution. Starting from a low periodic time series, the associated network is regular as the points arrange orderly round an orbit in the phase space. The network forms a big loop shape and its size increases proportionately to the number of points of the embedded data. If we add noise to the data, the dimension of the dynamics increases while the homogenous distribution of the points remain unchanged. we still get the loop structure but the diameter of the network decreases For chaotic time series, the reconstructed phase space becomes heterogeneous and may have some fractal properties and thus the associating network generates heterogeneity.
2.1. Global network properties

To investigate the distinction among different dynamical types in more detail, we study the average path length as a function of the time series length. We make use of the data from the x component of the Rössler system $x' = -(y + z)$, $y' = x + ay$, $z' = b + (x - c)z$ with $a = 0.1$ and $b = 0.1$. By tuning the parameter $c$, low periodic, high periodic and chaotic data can be obtained. The data is mapping into a space of dimension $d_c = 10$ with the time delay $\tau$ which is chosen to be the first minimum of the mutual information. Each point in the embedded space represents a node and we link it with its 4 nearest neighborhoods to form the recurrence-based phase space network with the mean degrees equal to 8. Thus, the number of nodes of the network equals to $N - \tau \cdot d_c$, where $N$ is the time series length.

Figure 1 shows how the average path length of the phase space network change with respect to the increasing length of the time series. Note that the average path length can be affected by the choice of initial point of the flow data, so the desired results are taken as an average over several networks from different segments of the same parameter. We find average path length $L(N)$ scales with $N$ with different scaling exponents for low periodic and chaotic Rössler systems. The average path length $L(N)$ increases linearly with time series lengths $N$ for low periodic time series with periods 2 and 3, but exponentially for chaotic Rössler. For high periodic cases, we find transient behavior when the size $N$ is small because the corresponding time series can be too short to cover enough periodic information. Moreover, the slopes for average path length $L(N)$ are larger than the two chaotic Rössler cases but smaller than the low periodic cases due to the interactions between two nearby orbits appear in the phase space.

We also study other global properties including the clustering coefficient and the degree distribution for Rössler systems of different dynamical type as shown in Figs. 2 and 3. The global clustering is closely related to motif ranking. As motif F that denotes a fully-connected subgraph of order 4 indicates strong mutual coupling between nodes, we may expect that the clustering coefficient for the periodic Rössler will be larger than the chaotic Rössler, because motif F occurs more frequently in periodic phase space network rather than network from chaotic data of the same Rössler system as mentioned in Ref. [1]. From Fig. 2 we can see that the clustering coefficients are rather stable with respect to an increasing $N$ and converges to around 0.68 for low periodic with period $= 2$, 3 as network size increases. Similar behavior can be found in the chaotic Rössler with $c = 18$. The clustering coefficients are also stable but the curve converges to a lower value around 0.58. In comparison, significant transient effect can be found in the remaining three cases. The curve for the chaotic Rössler with $c = 9$ first rises to a same high level as the low periodic cases for the $N$ between 1000 and 2000. Then, it turns...
down and finally converges to a low level as the chaotic case. The curve for Rössler of period 6 goes up and down and finally goes to the same value as low periodic cases. Although the value of clustering coefficient for Rössler of period 8 is closer to that of the chaotic cases, we may infer that, as the size of the network increases, the curve will finally come up to the low periodic values.

Rather than the global properties, we also examine local vertex properties which can provide more detailed information which is otherwise buried in the geometry of the attractor. The first global property we study is the local vertex degree which is the number of neighbors of a given vertex \( v \) in the network. The degree of a vertex for the traditional recurrence networks which based on an appropriate choice \( e \) reflects exactly the local recurrence rate. When it comes to phase space network, we should recall that the average degrees are 8 due to the distinctive construction methods we use to obtain the network. Thus the local degrees depend closely on the spatial filling of the phase space attractor. On the one hand, in the region where the density of the phase space is homogenous, the corresponding structures of the network retain the homogeneity from the attractor and local degrees for the vertices will be around 8. That is, every point in these regions is connected to its 4 nearest neighbors and at the same time gains around 4 extra links from points which take it as a nearest neighbor. One the other hand, if points locate in the regions where there are significant changes in the phase space density, some of these points may gain more than 4 extra links besides its 4 nominal nearest neighbors, resulting in the lack of neighbors for some other points in the same region. These regions will have a heterogeneous degree distribution. Long periodic signals follow a uniform distribution of low dimension while the trajectories of chaotic signals tend to be trapped in the vicinity of the unstable periodic orbits that lead to density changes in the phase space. One may refer to color coded representation of vertex degree on the attractor for chaotic Rössler with \( c = 18 \) in Fig. 5 for details. Also, we can address the different bandwidths in Fig. 3 which shows the degree distribution for the Rössler of different dynamical types. Since more heterogeneity is generated by the phase space of chaotic Rössler, the bandwidths will be wider.

2.2. Local network properties

![Figure 3: The degree distribution for the Rössler system of different dynamical types.](image)

Figure 3 shows the average degree distributions for networks of several realizations associated with 10000 points of the Rössler data with different dynamics. We find that the average degree distribution mimics the behavior of discrete Gaussian distribution for networks from the Rössler system in different dynamical regimes. The degree at which \( p(k) \) reaches its maximum is 8, which is the average degree of the network. The peak values of \( p(k) \) for networks generated from periodic data are all above 0.3 and the bandwidths of the Gaussian distribution are rather narrow, indicating that the networks are homogenous. In contrast, for chaotic Rössler, \( p(k) \) will be lower than 0.3 and the bandwidths of the Gaussian distributions are wider.

![Figure 4: Color coded representation of vertex degree on the attractor for chaotic Rössler with \( c = 18 \).](image)

![Figure 5: Color coded representation of vertex clustering coefficient on the attractor for chaotic Rössler with \( c = 18 \).](image)

The second property we study is the clustering coefficient of a specific vertex which quantifies the relative den-
sity of connections between its neighbors and thus measures the network in a relative larger scale compared to the local degree. Figure 5 shows color representation of vertex clustering coefficient on the attractor for the chaotic Rössler with $c = 18$, from which we observe the heterogeneously distributed local clustering coefficients. As the heterogeneity comes from the heterogenous distribution in the density of the phase space, one might expect that a node in the dense region of the phase space will naturally have a larger clustering coefficient and vice versa. However, it is shown in Fig. 5 that the values of clustering coefficient for some nodes locate in the dense region of the phase space could be small while the values for nodes locate the boundary region of the attractor could be large. In a homogenously filled phase space region, things become simpler and we can see points with relatively uniform and higher clustering coefficients. However, when it comes to a region with large density fluctuations, we have to consider the interactions between several orbits which are close to each other. A strong connectivity between neighbors of the same point could be obtained only if its neighbors tend to lie in different orbits, because we have excluded links within the same orbit when constructing the phase space networks. We can conclude that the local clustering coefficient contains the complexity of the interaction between orbits in the phase space and can provide complementary information of the system.

Figure 6: Color coded representation of logarithm of vertex betweenness centrality, $\log(b_v + 1)$, on the attractor for the chaotic Rössler with $c = 18$.

The third property we study is the betweenness centrality of a specific vertex which quantifies the relative shortest path length pass through the vertex and thus it is more complicated because it measures the network in an even larger scale compared to the clustering coefficient and the local degree. What we are most interested in are the points with a large betweenness centrality which are of importance for the many shortest paths on the network. As shown in Fig. 6, the nodes with high betweenness centrality may correspond to the region with low phase space density between such regions because of an increasing number of shortest path between such regions. For vertices with large degree in the dense region of the phase space close to the unstable periodic orbits, low values of betweenness centrality can be found. A similar conclusion has been drawn for the traditional recurrence networks [7].

3. Conclusions

In this paper, we apply standard complex network measures to recurrence-based phase space network constructed by linking every point with its $k$ nearest neighbors in reconstructed phase space. In particular, global properties such as average path length, the global clustering coefficients and degree distribution can be used to distinguish system of different dynamical types. Local degrees, clustering coefficients and betweenness centrality reveal information on the spatial fillings of the phase space and are close to dynamical invariants such as the unstable periodic orbits. The specific vertex properties can provide more detailed information about the local attractor geometry in the phase space.

Acknowledgments

This work was supported by a Hong Kong Polytechnic University direct allocation (G-YG35).

References

Isle of Eden in 1D binary cellular automaton as a manifestation of Gödel incompleteness and a proposal for a bridge between analytical results and spatial-temporal logic patterns

Tamás Roska

Computer and Automation Research Institute of the Hungarian Academy of Sciences and the Faculty of Information Technology and the Jedlik Laboratories of the Pazmany P. Catholic University, Budapest, Hungary
Email: roska@sztaki.hu

Abstract—The richness of spatial-temporal dynamics is known since the Morphogenesis paper of Turing [1] and the introduction of the Cellular Automaton by Von Neumann and Ulam [2]. The recent book of S. Wolfram [3] showcases, by a wealth of examples and one theorem (rule 110), the richness of the simplest spatial temporal binary patterns in the one dimensional (1D) binary cellular automaton, a special case of standard CNN Dynamics [8].

The rigorous study of this model by L. O. Chua and others [4] led to a surprisingly simple and deep insight in the qualitative behavior of these models. The existence of the so called Isle of Eden, a sequence of states without predecessors and successors, is one interesting phenomenon.

The aim of this paper is twofold. (i) We show that the Isle of Eden is a simple manifestation of Gödel’s incompleteness theorem by using the way of the original proof of Gödel. (ii) We propose a way to generate results on binary spatial-temporal logic patterns via analytical proofs using the binary to continuous transformation introduced by L. O. Chua et al. [4].

We note that the 1D Cellular Automaton is a simple case of a CNN dynamics programmably embedded in a CNN Universal Machine [5] and it could be simply implemented on existing cellular camera computers [6], as well as on different cellular many-core chips like FPGAs.

1. Introduction

A brief history from Boole to Gödel shows how logicians starting from the formal logic algebra, the Boolean algebra, developed a massive theory of logic reasoning and this process culminated in the astonishing proof of Gödel’s theorem [7]. The latter is showing that in any formal system, defined by a set of axioms and logic operators, there exists at least one statement that is undecidable.

2 The 1D CNN and the binary Cellular Automaton [8]

Standard Cellular Nonlinear/neural Networks (CNNs) are defined on a 1D, 2D or 3D topographic grid, placing mainly identical 1st order nonlinear dynamic systems called cells on the grid points, and introducing mainly local interactions between the cells called cloning templates. If the inputs are binary and the templates are completely stable then the outputs are binary as well. Hence, the 1D Cellular Automaton is a special case of a standard CNN, the sequence of outputs being the input and initial state of the next step in the sequence.

3. The 1D Binary Cellular Automaton generating sequences of binary states (SBS) as Gödel’s recursive universe

Gödel’s metamathematical method is based on attaching numbers to the signs and series of signs existing in his formal system P via Gödel numbering [7]. It is a one dimensional coordinate metamathematics. There exists a one-to-one correspondence between a sign/string and a natural number g ∈ G (the set of Gödel numbers) G is a subclass of \( \mathbb{Z} \) (integers).

Assign natural numbers starting with 2, like 2, 3, 4, 5, etc., to signs, like \( \text{n}_1, \text{n}_2, \text{n}_3, \text{n}_4, \text{n}_k \), for \( k \) different signs. A series of \( k \) signs is represented by an integer \( g = 2^{n_1} * 3^{n_2} * 5^{n_3} * 7^{n_4} * pk^{n_k} \), \( pk \) being the \( k \)-th prime number starting from 2.

Hence for every class of strings of symbols and signs there is a one-to-one correspondence of a Gödel number.

To any relation \( R \) between strings there is a one-to-one correspondence \( R' \) between Gödel numbers.

For example the metamathematical statement that “the series \( s \) of formulae is a “proof” of formula \( f \)” is true if and only if a certain arithmetical relation holds between the Gödel numbers of \( s \) and \( f \), ( \( s \) and \( f \) ) being the proof of.

Notation: \( x \) \( \text{B} \) \( y \): \( x \) is a proof (Beweis) of formula \( y \)

Recursiveness: We are applying a series of operators, relations, all together \( n \)-times, the \( n \)-th one is denoted by \( R(n) \) is a recursive definition.

A class sign \( \alpha \): \( v \) is a series of signs which is a formula which contains exactly one free variable \( v \). A class-sign is recursive if it can be interpreted as a recursive arithmetical class and if \( v \) is substituted the number sign will be provable or disprovable. A recursive relation sign may contain several free variables.

Proposition VI of Gödel is more general than the existence of undecidable formulae in the formal System P.
since it contains deductions (not only proofs) from
formulae not included among the axioms.

Proposition VI: If the formal System P satisfies certain
conditions of consistency, then there exists at least one
recursive class sign r in P such that neither v Gen r nor
Neg (v Gen r) is provable within P, where v Gen r is the
generalization of r with respect to its free variable v.

Now, we are in a position to represent the 1D binary
CA via Gödel’s framework.

We are considering a 1D, 1 neighbor, length L, binary
state Cellular Automaton with periodic boundary
condition, denoted by 1D1nLbCA or abbreviated as
standard 1DCA.

In the table below, we assign the different notions of
1DCA to different notions of a formal system P.

<table>
<thead>
<tr>
<th>1D CA</th>
<th>Formal System P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Signs: states xi</td>
<td></td>
</tr>
<tr>
<td>(2,3,4...N)</td>
<td></td>
</tr>
<tr>
<td>x:[b_0, b_1,...,b_L-1]</td>
<td>N = 2^L</td>
</tr>
<tr>
<td>initial state: x_0</td>
<td></td>
</tr>
<tr>
<td>state sequences S^k ∈ S^*</td>
<td></td>
</tr>
<tr>
<td>S^k : k states, S^* : trajectory set</td>
<td></td>
</tr>
<tr>
<td>x_0, x_1, ......, x_k-1</td>
<td></td>
</tr>
<tr>
<td>a series of k signs,</td>
<td></td>
</tr>
<tr>
<td>represented by a Gödel number</td>
<td></td>
</tr>
<tr>
<td>One class of trajectories</td>
<td></td>
</tr>
<tr>
<td>CA Rule R</td>
<td></td>
</tr>
<tr>
<td>256 different rules</td>
<td></td>
</tr>
<tr>
<td>Trajectory: applying rule R recursively, initial state x_0</td>
<td></td>
</tr>
<tr>
<td>Relation R</td>
<td></td>
</tr>
<tr>
<td>Recursive definition</td>
<td></td>
</tr>
<tr>
<td>Recursive class sign with one free variable v</td>
<td></td>
</tr>
<tr>
<td>Isle of Eden can not be reached from outside of it, neither another state</td>
<td></td>
</tr>
<tr>
<td>outside of it can be reached from Isle of Eden (Reached: via recursive rules)</td>
<td></td>
</tr>
<tr>
<td>v Gen R can not recursively defined starting from axioms</td>
<td></td>
</tr>
<tr>
<td>One class of G numbers</td>
<td></td>
</tr>
</tbody>
</table>

This means that the Isle of Eden is a simple
manifestation in a spatial-temporal setting of the
incompleteness.

4. A method to generate results on binary spatial-
temporal logic patterns via analytical proofs using the
binary to continuous transformation.

The framework [4] analyzing the binary 1D CA
contains a very important one-to-one transformation from
the spatial binary representation to one continuous value
representation of single state. We denote it by BtoC (Binary to Continuous). This map is a bijection, hence the
BtoCtoB map is unique in both directions.

The following way would lead to discover new spatial-
temporal logic patterns via analytic techniques.

Logic representation of binary patterns ► binary state
and dynamics ► BC ► continuous state and dynamics ►
theorem on continuous dynamics ► CB ► theorem on spatial binary dynamics ► theorem on logic representation

In this way, a theorem proved rigorously in the real valued domain could be converted back to the logic
pattern domain. An example is the proof of fractal
properties [10].

References

Philosophical Transactions of the Royal Society of
(ed. A. W. Burks),” University of Illinois Press, Urbana-
London, 1966
Media, Inc. Champaing, IL, 2002
of Wolfram’s New Kind of Science,” World Scientific,
Principia Mathematica and related systems, Introduction
and Visual Computing,” Cambridge University Press,
of Wolfram’s New Kind of Science. Part IX: Quasi-
ergodicity,” International Journal of Bifurcation and
Symbolic Dynamics of Some Bernoulli-shift Cellular Automata Rules

Guanrong Chen1  Fangyue Chen2  Junbiao Guan2  Weifeng Jin3

1Department of Electronic Engineering, City University of Hong Kong
Kowloon, Hong Kong, China
2School of Science, Hangzhou Dianzi University
Hangzhou, Zhejiang 310018, China
3College of Pharmaceutical Sciences, Zhejiang Chinese Medical University
Hangzhou, Zhejiang 310053, China
Email: eegchen@cityu.edu.hk  fychen@hdu.edu.cn

Abstract—In our recent study, the dynamics of some elementary cellular automata rules are investigated in the bi-infinite symbolic sequence space. These rules, as members of the Wolfram class II and Chua’s topologically-distinct Bernoulli-shift rules, which were believed to be simply periodic before, actually display rich and complex dynamics. Rules 2 and 35, for example, to be discussed in this paper, are chaotic in the sense of Li-Yorke and/or Devaney; they are topologically transitive or topologically mixing; they have positive topological entropies; and they show a catalog of gliders and glider collisions.

1. Introduction

Cellular Automata (CA), formally introduced by Neumann in the early 1950’s, are a class of spatially and temporally discrete, deterministic mathematical systems characterized by local interactions and an inherently parallel evolution [1]. Twenty years later, Conway proposed his now-famous game of life [2]. From a theoretical point of view, it is interesting because it has the power of a universal Turing machine: anything that can be computed algorithmically can be computed within this game of life. Mathematical theory of CA was further developed by Hedlund in 1969 [3], who viewed one-dimensional CA (1D CA) in the context of symbolic dynamics as homomorphisms of the full shift. His main research is not directly related to the characterizations of surjective and open CA but with the current problems in symbolic dynamics. In the early 1980’s, Wolfram carried out intensive research on dynamical and computational aspects of CA [4-6]. He classified the 256 elementary cellular automata (ECA) informally into four classes using dynamical concepts like periodicity, stability, chaos and complexity. In 2002, Wolfram introduced his monumental work A New Kind of Science [7]. Based on this work, Chua et al. provided a nonlinear dynamics perspective to Wolfram’s empirical observations from the viewpoint of mathematical analysis using tools like characteristic function, forward time-τ map, basin tree diagram and Isle-of-Eden digraph [8-11].

Although there are 256 ECA rules, only 88 rules are globally independent from each other. These 88 rules are organized into four groups with distinct qualitative dynamics, corresponding to random initial configurations: period-k rules (k = 1, 2, 3, 6), Bernoulli-shift rules, complex Bernoulli-shift rules and hyper-Bernoulli-shift rules. By using random bit strings as testing signals, 30 topologically-distinct Bernoulli-shift rules have been found, via extensive computer simulations and analytical studies, to exhibit robust Bernoulli-shift steady-state behavior. Even more remarkable is their robustness in the sense that any random initial bit string could converge to one of these robust Bernoulli-shift attractors. In particular, each Bernoulli attractor is parameterized by three integers (σ, τ, β) satisfying β = ±2^τ. A Bernoulli-shift attractor with β > 0 simply implements a left shift (if σ > 0) or a right shift (if σ < 0) of |σ| bits in every τ iterations. If β < 0, the same operation is followed by complementation (i.e. changing symbol of all bits) during each operation (i.e. every τ iteration).

Due to the fact that many topological properties such as topological entropy, sensitivity and topologically mixing of CA are undecidable [12, 13], one should, in principle, separately analyze time-asymptotic dynamics for each Bernoulli-shift rule in the bi-infinite binary sequence space S^Z. At present, their long-term dynamical behaviors are analyzed from the viewpoint of symbolic dynamics [15-20].

2. Complex symbolic dynamics of Bernoulli-shift rules

For a finite symbol set S = {0, 1,..., k − 1}, a word over S is a finite sequence a = (a0, a1,..., ak) of elements in S. Denote by S^* = ∪n≥0 S^n the set of words over S, where S^n is the set of all words of length n. If a is a finite or infinite-length word and I = [i, j] is an interval of integers on which a is defined, then write a|[i,j] = (a_i,..., a_j) and a|[i] = (a_i,..., a_j−1). Also, b is a subword of a, denoted by b ≺ a, if b = a_i for some interval I ⊆ Z; otherwise, denoted by b ∦ a. The set of bi-infinite words is denoted by S^Z. In S^Z, the cylinder set of a word a ∈ S^Z is [a]|k = {x ∈ S^Z | x[|k+k+n|] = a}, where k ∈ Z. Obviously, the cylinder sets generate the topology on S^Z and form a countable basis for this topology. Every open set is a countable union of some cylinder sets. For a self-map f on S^Z, a set
$X \subseteq S^Z$ is $f$-invariant if $f(X) \subseteq X$, and strongly $f$-invariant if $f(X) = X$. If $X$ is closed and $f$-invariant, then $(X, f)$ or simply $X$ is called a subsystem of $(S^Z, f)$. For instance, let $\mathcal{A}$ denote a set of some finite words over $S$, and $\Lambda = \Lambda_\mathcal{A}$ be the set consisting of the bi-infinite words in $\mathcal{A}$. Then, $\Lambda_\mathcal{A}$ is a subsystem of $(S^Z, \sigma)$, where $\mathcal{A}$ is said to be the determinative block system of $\Lambda$ with $\sigma : S^Z \rightarrow S^Z$ denoted by $[\sigma(x)]_i = x_{i+1}$. For a closed $\sigma$-invariant subset $\Lambda \subseteq S^Z$, the subsystem $(\Lambda, \sigma)$ is a subshift of finite type. The topological dynamics of a subshift of finite type is largely determined by the properties of its transition matrix $A$ [14]. A matrix $A$ is positive if all of its entries are non-negative; irreducible if there exists $M$, such that $A^n > 0$, $\forall n > M$ and $\forall i, j$. If $\Lambda_\mathcal{A}$ is a 2-order subshift of finite type, then it is topologically mixing if and only if $A$ is aperiodic, where $A$ is its associated transition matrix with $A_{ij} = 1$, if $(i, j) < \Lambda$; otherwise $A_{ij} = 0$.

By a theorem of Hedlund [3], a map $f : S^Z \rightarrow S^Z$ is a cellular automaton if and only if it is continuous and commutes with the shift map $\sigma$, i.e. $\sigma f = f \sigma$. For any CA, there exists a radius $r \geq 0$ and a local rule $N : S^{2r+1} \rightarrow S$ such that $[f(x)]_i = N(x_{[i-r,i+r]})$. Moreover, $(S^Z, f)$ is a compact dynamical system. To enhance readability, it is desirable to write a CA as a global map $f_N$ for local rule $N$, where $N$ is the decimal notation in light of Wolfram’s scheme. For one-dimensional elementary cellular automata (ECA), $r = 1$ and $S = \{0, 1\}$. For example, rule 2 is the one for which $N(0, 0, 1) = 1$ and $N(0, 0, 1, 0) = 0$ for all other triples. Based exclusively on the rule, the following proposition is a direct consequence.

**Proposition 1** For rule 2, there exists a subset $\Lambda \subseteq S^Z$ such that $f_2|_\Lambda = \sigma\Lambda$, where $\Lambda = \{x \in S^Z | x_{[r-1,r+1]} \in \mathcal{A}, \forall i \in \mathbb{Z}\}$ and $\mathcal{A} = \{(0,0,0),(0,0,1),(0,1,0),(1,0,0)\}$.

According to Proposition 1, dynamical behaviors of $f_2$ on $\Lambda$ can be characterized via a subshift $\Lambda$, which is a subshift of finite type. Observe that the transition matrix $A$ of the 2-order subshift, which is topologically conjugate to $\Lambda$, is

$$A = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}.$$  

One can easily verify that $A$ is irreducible and aperiodic.

**Proposition 2** 

(a) $f_2$ is topologically mixing on $\Lambda$;

(b) $\Lambda$ is the nonwandering set $\Omega(f_2)$ and the global attractor of $f_2$;

(c) the topological entropy of $f_2$ equals $\log \rho(A) = \log \lambda_0 \approx 0.382$, where $\rho(A)$ is the spectral radius of the transition matrix $A$ and $\lambda_0$ is the positive real root of $\lambda^3 - \lambda^2 - 1 = 0$.

It is well known that positive topological entropy implies chaos in the sense of Li-Yorke, and topologically mixing is also an indication of complexity of dynamical systems. A system with topologically mixing property has many chaotic properties by different means. In conclusion, the above two propositions have led to the following theorem.

**Theorem 1**

(a) $f_2$ is chaotic in the sense of both Li-Yorke and Devaney on the global attractor $\Lambda$;

(b) $f_2$ is chaotic in the sense of Li-Yorke on $S^Z$.

**Remark 1** Rules 2, 10, 34, 42, 46, 130, 138 and 162 have the same shifting mode: shift the bit strings to the left by one bit in every iteration, i.e., they have the same Bernoulli-shift attractor parameters $(\sigma, r, \beta) = (1, 1, +)$. In order to describe the Bernoulli shifting dynamics in an unambiguous way, analogous results for these rules are listed in Table 1. Column 3 displays the approximation of topological entropy $\log \lambda_0$, where $\lambda_0$ is the positive real root of the characteristic equation of the corresponding transition matrix. Column 5 shows the topologically conjugate rules associated with rule $N$.

<table>
<thead>
<tr>
<th>$N$</th>
<th>topologically mixing</th>
<th>topological entropy</th>
<th>Li-Yorke and Devaney</th>
<th>equivalent rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>yes</td>
<td>0.382</td>
<td>Li-Yorke and Devaney</td>
<td>16,192,247</td>
</tr>
<tr>
<td>10</td>
<td>yes</td>
<td>0.481</td>
<td>Li-Yorke and Devaney</td>
<td>80,175,245</td>
</tr>
<tr>
<td>34</td>
<td>yes</td>
<td>0.481</td>
<td>Li-Yorke and Devaney</td>
<td>48,187,243</td>
</tr>
<tr>
<td>42</td>
<td>yes</td>
<td>0.609</td>
<td>Li-Yorke and Devaney</td>
<td>112,171,241</td>
</tr>
<tr>
<td>46</td>
<td>yes</td>
<td>0.382</td>
<td>Li-Yorke and Devaney</td>
<td>116,139,209</td>
</tr>
<tr>
<td>130</td>
<td>no</td>
<td>0.382</td>
<td>Li-Yorke and Devaney</td>
<td>144,190,246</td>
</tr>
<tr>
<td>138</td>
<td>yes</td>
<td>0.562</td>
<td>Li-Yorke and Devaney</td>
<td>208,174,244</td>
</tr>
<tr>
<td>162</td>
<td>no</td>
<td>0.481</td>
<td>Li-Yorke</td>
<td>176,186,242</td>
</tr>
</tbody>
</table>

**Table 1:** Summary of quantitative properties of robust Bernoulli-shift rules 2, 10, 34, 42, 46, 130, 138 and 162 with the same parameters $(\sigma, r, \beta) = (1, 1, +)$.

Similar analytic arguments can also be applied to all other robust Bernoulli-shift rules. In addition to the 8 Bernoulli-shift rules listed so far (with the same Bernoulli parameters), there are 22 additional ones possessing up to three robust Bernoulli attractors. Space limitation precludes a more detailed analysis of the others. Main dynamical properties of typical Bernoulli-shift rules 3, 7, 11, 24, 35, 142, and 74 have been derived and exhibited in Table 2, but for others it is referred to [15-19].

**Remark 2** It has been shown that some special examples of Chua’s robust period rules also exhibit non-robust Bernoulli-shift behaviors for both finite and bi-infinite cases [20].
Table 2: Summary of quantitative properties of robust Bernoulli-shift rules 7, 24, 35, 142 and 74 with distinct Bernoulli shifting types.

3. Gliders and collisions in Bernoulli-shift rules

ECA rules 54 and 110, which are classical examples of Wolfram’s class IV, have been an object of special attention due to their ability to generate rich varieties of periodic structures, known as particles or gliders [7, 21, 22]. A glider is a compact group of non-quiescent states travelling along cellular automata lattice. Gliders are believed to be the characteristic signatures allowing one to recognize class IV behavior, whereas all types of Bernoulli-shift dynamics can also be considered as gliders. Certainly, the catalog of all possible gliders and glider collisions is less plentiful than that of rules 54 and 110. These ideas are formalized under the framework of symbolic dynamics below.

For a CA rule N and its global map \( f_N \), an infinite cyclic configuration \( x \) is \( x = a^* = (\cdots, a, a, a, \cdots) \in S^Z \), where \( a = (a_0, a_1, \cdots, a_{n-1}) \in S^n \). Further, \( a_L^* = (\cdots, a, a) \) and \( a_R^* = (a, a, \cdots) \) stand for left-infinite cyclic configuration and right-infinite cyclic configuration, respectively. The evolution orbit \( \{x, f_N(x), f_N^2(x), f_N^3(x), \cdots \} \), denoted by \( \text{Orb}_N(x) \), is said to be a background ether or background pattern of rule N, if there exist nonnegative integers \( m, m', k \) and \( k' \) such that \( f_N^m(x) = \sigma_R^k(x) \) and \( f_N^m(x) = \sigma_L^{k'}(x) \), where the sequence \( a \) is called an ether factor.

Suppose that \( x = a^* \) is an infinite cyclic configuration with \( a \in S^n \) and \( b \in S^m \) such that \( b \neq (a, a, \cdots, a) \), \( k \in Z^* \), and \( \tilde{x} = (a_L^*, b, a_R^*) \). Then:

1. If there exist nonnegative integers \( m \) and \( k \) such that \( f_N^m(\tilde{x}) = \sigma_R^k(\tilde{x}) \), then the orbit \( \text{Orb}_N(x) \) is called a right glider of rule \( N \) with velocity \( k/m \);

2. If there exist nonnegative integers \( m \) and \( k \) such that \( f_N^m(\tilde{x}) = \sigma_L^k(\tilde{x}) \), then the orbit \( \text{Orb}_N(x) \) is called a left glider of rule \( N \) with velocity \( -k/m \);

3. If there exists nonnegative integers \( m \) such that \( f_N^m(\tilde{x}) = \tilde{x} \), then the orbit \( \text{Orb}_N(x) \) is called a fixation glider of rule \( N \) with zero velocity;

4. If the orbit \( \text{Orb}_N(\tilde{x}) \) is a right (left, fixation) glider, then the subsequence \( b < \tilde{x} \) is called a glider factor of rule \( N \);

5. If \( b \) and \( c \) are two glider factors of rule \( N \) with disagreed velocities, then the evolutionary orbit \( \text{Orb}_N(y) \) is called a glider collision of rule \( N \), where \( y = (a_{L}^*, b, a, a, \cdots, a, c, a_{R}^*) \).

In the following, the background ether, gliders and glider collisions of rule 35 are further discussed.

**Proposition 3** For ECA rule 35, the evolutionary orbit \( \text{Orb}_{35}(x) \) is a background ether, where \( x = (0, 0, 1)^* \); that is, \( (0, 0, 1) \) is an ether factor of rule 35.

**Fig. 1:** Background ether of rule 35.

**Proposition 4** Let \( y = (a_{L}^*, b, a_{R}^*) \), \( a = (0, 0, 1) \) and \( b = (0, 1) \). Then, the evolutionary orbit \( \text{Orb}_{35}(y) \) is a left-glider with velocity \(-1\) of rule 35.

**Proposition 5** Let \( y = (a_{L}^*, b, a_{R}^*) \), \( a = (0, 0, 1) \), and \( b = (0, 1, 1), (0, 0, 0), (1, 1, 0), (1, 0, 0, 0) \) and \((1, 1, 0, 0, 0, 0))\), respectively. Then, these evolutionary orbits \( \text{Orb}_{35}(y) \) are all right-gliders with velocity \(1/2\) of rule 35.

**Fig. 2:** A left-glider and six right-gliders of rule 35.

**Fig. 3:** Glider collisions in rule 35.

It is now clear that there are two types of collision outcomes in rule 35: (1) annihilation of gliders and stationary
localization; (2) collision between two gliders leading to another type of glider. For ease of visualization, the background ether, gliders and glider collisions of rule 35 are illustrated in Figs. 1, 2 and 3, respectively, where the black square or blue square stands for “1”, and the white square or orange square stands for “0” in the spatiotemporal evolution patterns (colors online).

4. Conclusion

The past few years have witnessed some dramatic advances in the field of cellular automata. One of the main challenges is to explore the quantitative dynamics of these dynamical systems. By taking advantage of robust Bernoulli-shift rules uncovered by Chua and his colleagues, we have developed an elementary yet rigorous proof to explain their chaotic dynamics in view of symbolic dynamics. That is, the intrinsic complexity of all of these Bernoulli-shift rules is quite high according to the usual features that quantify the complexity of discrete dynamics, such as topological entropy and topologically mixing. It is worth mentioning that the qualitative property of glider collision plays an important role in giving a quantitative interpretation of the global dynamics of some CA rules in the bi-infinite sequence space \( S^2 \).

Acknowledgments

This research was jointly supported by the City University of Hong Kong (Grant No. SRG 7002274) and NSFC (Grant No. 6087209 and No. 10832006).

References


Gardens of Eden: where Nonlinear Dynamics and Formal Languages meet

Giovanni E. Pazienza† and Marion Oswald‡

†Cellular Sensory and Wave Computing Laboratory, MTA - SZTAKI, Budapest, Hungary and Pazmany University, Budapest, Hungary
‡Cellular Sensory and Wave Computing Laboratory, MTA - SZTAKI, Budapest, Hungary and Vienna University of Technology, Vienna, Austria
Email: gpazienza@sztaki.hu, marion@sztaki.hu

Abstract—Many classifications of the Cellular Automata rules have been proposed in the last decades. Some of them are based on empirical observations, and hence their rigor is doubtful, while others are based on precise, though often complicated, mathematical methods. In this paper, we present two new classifications for Cellular Automata rules, which come from two different disciplines; namely, Nonlinear Dynamics and Formal Language Theory. These classifications are straightforward to obtain and give remarkably similar results.

1. Introduction

Cellular Automata (CA) have been thoroughly studied in the last decades, but they are far from being completely characterized. In the last few years, CA have been seen “through the eyes” of Nonlinear Dynamics [1]; this new perspective led, among other results, to a classification of CA rules into six groups, depending on their dynamics. In this paper, we propose a variant of such classification. We were inspired by a classical work [2] in Nonlinear Dynamics that proves that in a continuous function defined over an interval the presence of a period-3 orbit implies the presence of orbits with any other period. Obviously, this result does not apply directly to Cellular Automata because, in general, they describe a discontinuous function. Nevertheless, we noticed that all rules in the last two groups of Chua’s classification have at least one periodic orbit with \( \tau = 3 \) (the notion of ‘periodic orbit’ and the meaning of the parameter \( \tau \) will be explained in Sec. 2.2) and, at the same time, all rules with \( \tau = 3 \) included in the remaining four groups of Chua’s classification tend to have a more complex behavior than the others [3]. For this reason, in our classification we distinguish only two groups: i) rules having exclusively periodic orbits with \( \tau = 1 \) and/or \( \tau = 2 \); ii) rules having periodic orbits with \( \tau \geq 3 \).

Also, we propose yet another classification of the CA local rules, based on the formal languages formed by the so-called orphand patterns. In this paper, we will show that there is an excellent correspondence between these two classifications and discuss its implications.

The paper is structured as follows: in Sec. 2, we discuss the basic features of CA and our classification of the CA rules based on the properties of their periodic orbits; in Sec. 3, we introduce a few fundamental notions of Formal Language Theory; in Sec. 4, we define the concepts of Garden of Eden and orphan pattern, and show how they can be used to classify CA rules; in Sec. 5, we present the comparison between our two classifications; in Sec. 6, we draw the conclusions and outline our future work.

2. Brief notes on Cellular Automata

2.1. Generalities

Cellular Automata [4] consist of regular uniform lattice of cells assuming a finite number of states; here, we consider one-dimensional CA in which cells are arranged in an array of length \( L = I + 1 \) and can take only two states: 0 and 1. For instance, a bit string \( x \) at the generic time step \( n \) is

\[
x^{n} = (x_{0}^{n}, x_{1}^{n}, \ldots, x_{I}^{n})
\]

where the subscripts indicate the position of the cell in the array. Hereafter, letters in bold indicate bits strings, and letters in italics are used for the single bits. Cells are updated synchronously and the time evolution of a bit string can be effectively summarized by the notation

\[
x^{n+1} = f(x^{n})
\]

in which the superscript indicates the iteration. The state of each cell at iteration \( n + 1 \) depends on the states of its neighbors (here we consider only the nearest neighbors) at iteration \( n \),

\[
x_{i}^{n+1} = f(x_{i-1}^{n}, x_{i}^{n}, x_{i+1}^{n}).
\]

In the following, we use periodic boundary conditions, which means that

\[
x_{0}^{n+1} = f(x_{1}^{n}, x_{0}^{n}, x_{I}^{n}) \quad \text{and} \quad x_{I}^{n+1} = f(x_{I-1}^{n}, x_{I}^{n}, x_{0}^{n})
\]

Under the restrictions detailed above, there are only 256 possible functions \( f \), called rules, which we can be denoted by \( f_{0} \) up to \( f_{255} \). For instance, the notation:

\[
x^{n+2} = f_{110}(f_{110}(x^{n}))
\]

indicates the application of rule 110 to the bit string \( x^{n} \) two times to obtain the bit string \( x^{n+2} \). However, the 256 local rules show only 88 distinct dynamics, since each CA
rule can have up to four globally-independent rules, obtained by complementing and/or mirroring the original one, as described in [1]. Therefore, from now on we will often refer to the 88 globally-independent Cellular Automata rules, making it clear that the choice of the representative specimen for each of the 88 equivalence classes is arbitrary.

2.2. Complexity index and Periodic orbits

We can associate an index of complexity $\kappa$ to all CA local rules thanks to the procedure defined in [1]. There are 104 rules (38 of which are globally-independent) with $\kappa = 1$, 126 rules (41 of which are globally-independent) with $\kappa = 2$, and 26 rules (9 of which are globally-independent) with $\kappa = 3$. Observe that the universal rule 110 (and its three globally-equivalent rules 124, 137, and 193) has $\kappa = 2$. Therefore, the dynamics of rules with $\kappa = 3$ is not necessarily more complex than those with $\kappa = 2$.

If the functions $f_i$ are deterministic and the length $L$ of the bit string is finite, then the evolution of an arbitrary initial state under an arbitrary rule $f_k$ will end up in a periodic cycle, in the sense that there exist $p$ and $T$ such that

$$x^p = x^{p+T}$$

(6)

Obviously, $x^n = x^{n+T}$, for all $m > p$. The bit strings from $x^0$ to $x^{p-1}$ are said to belong to the transient, which has length $p$, while the bit strings from $x^p$ on are said to belong to the periodic orbit, which has length $T$. Furthermore, it may happen that the there exists a $\tau$ for which:

$$x^p = S^\tau(x^{p+\tau})$$

(7)

where $S^\tau$ indicates a shift, left or right, by $\sigma$ positions. Therefore, a periodic orbit can be characterized by its parameters $\tau > 0$ and $\sigma \geq 0$.

Among the 88 globally-independent 1D binary Cellular Automata rules, some are particularly relevant, such as rule 110 which has been proved to be universal, while others do not perform any interesting task, such as rule 0 because all possible initial bit strings evolve into $(0 \ldots 00)$ in one iteration. In order to group the rules according to the properties of their periodic orbits, both Wolfram and Chua proposed a classification according to the dynamics obtained by using a long ($L$ greater than 100) random bit string as initial state of the Cellular Automaton.

In this paper, we use a variant of Chua’s classification that, instead of using a long random bit string as initial state, uses all bit strings with length $L$, where $L \leq 15$, as initial states checking whether there is any periodic orbit with $\tau \geq 3$. In fact, as already pointed out by [5] and [6], there are some rules for which ‘interesting’ periodic orbits exist, but they cannot be found through the method used by Wolfram and Chua because the probability of obtaining them from random initial bit string is extremely low.

3. Formal Languages

In this section, we introduce a few basic concepts on formal languages, which will be used in the following; more details can be found in the classical literature, e.g., [7].

An alphabet $\Sigma$ is a non-empty set of symbols. A finite sequence of symbols from $\Sigma$ is called a word and its length is the number of symbols it contains. The word with length 0 is called the empty word and denoted by $\lambda$. The set of all words over an alphabet $\Sigma$ (including the empty word) is denoted by $\Sigma^*$, while the set of nonempty words is denoted by $\Sigma^+$, i.e., $\Sigma^+ = \Sigma^* \cup \{\lambda\}$. Any set of words $L \subseteq \Sigma^*$ is called a (formal) language. A formal language is called finite if it contains only a finite number of words. An infinite language in which the number of words with a given length $l$ is bounded by a constant independent of $l$, for all $l \in \mathbb{N}$, is called slender. For a word $w \in \Sigma^*$, where $w = xuy$ for $x,y \in \Sigma^*$, we call $x$ prefix, $u$ subword and $y$ suffix of $w$.

For a language $L$, the set $L' = \Sigma^* - L$ is called the complement of $L$ containing all the words over an alphabet $\Sigma$ that are not contained in $L$. In the following, we will be mainly concerned with factorial languages: a language $L$ is called factorial if for every word $w \in L$ all the subwords of $w$ also belong to $L$. Then any word $w \in \Sigma^*$ is called distinct excluded block (DEB, for short) or forbidden word, if $w \notin L$ but every proper substring of $w$ is in $L$. Let $L'$ be the set of distinct excluded blocks of a language $L$, then we can write $L = \Sigma^*L'S$ and $L = \Sigma^* - \Sigma^*L'S$. Hence a factorial language can also be defined by $L'$, sometimes called the antidictionary. It is worth noting that $L'$ eliminates far more words from $L$ then only the ones contained in $L'$ (namely, all those having a DEB as proper substring).

4. Gardens of Eden and orphans patterns

For most of the rules, there are some strings, called Gardens of Eden, that cannot be generated through the evolution of a given rule. In other words, the bit string $x$ is said to be a Garden of Eden of rule $N$ if there is no $x'$ such that $x = f_N(x')$.

In this paper, we consider the set of the Gardens of Eden of length $L$ that do not contain any Garden of Eden of length $L-1$ as proper substring, which are also called orphan patterns. Such set corresponds with the set $L'$ of DEB of the factorial language $L$, where $L$ is the language consisting of all bit strings of arbitrary sizes generated in one iteration by the rule $N$. We then analyzed the language $L'$ of DEB or orphans (see Sec. 3) for all 88 globally-independent rules, and grouped them according to their formal language theoretic properties:

1. $L'$ is empty, i.e., there are no orphans patterns;
2. $L'$ is finite, i.e., there is only a finite number of orphan patterns;
3. \( L' \) is **slender**, i.e., there is an infinite number of orphan patterns but their number for each given string length \( L \) is bounded by a constant independent of \( L \);

4. \( L' \) is **non-slender**, i.e., there is an infinite number of orphan patterns and their number for each given string length \( L \) is growing with \( L \).

5. **Results**

First of all, we classified the 88 globally-independent rules into the two groups, as defined in Sec. 1. The first group includes all rules having only \( \tau = 1 \) and/or \( \tau = 2 \), while the second group includes all rules having also \( \tau \geq 3 \). Second, we classified the same 88 globally-independent rules into the four classes defined in Sec. 4. The correspondence between these two classifications is very good, as it can be appreciated in Tables 1, 2, and 3.

In particular, from Table 1 we can observe that the orphan patterns of all rules having only orbits with \( \tau = 1 \) and/or \( \tau = 2 \) constitute either finite or slender languages. Furthermore, the first case corresponds to all rules with \( \kappa = 1 \). This is consistent with Chua’s definition of complexity index: the threshold of complexity beyond which rules can exhibit ‘interesting’ dynamics, and whose orphan patterns can form non-finite languages, is 2.

From Table 2 we notice that all CA rules having orbits with \( \tau \geq 3 \) constitute (infinite) non-slender languages. This implies that the number of words with a given length \( L \) is growing with \( L \).

The third and last group is formed by all rules having no orphans. Such rules are **surjective** when \( L \) is infinite, in the sense that given a rule \( N \), for any bit string \( x \) there will be another bit string \( x' \) such that \( x = f_N(x') \). Nevertheless, this is not true when \( L \) is finite. In this last case, some rules – namely, 15, 51, 170, and 204 – are still surjective, while other rules – namely, 30, 45, 60, 90, 105, 106, 150, 154 – have Gardens of Eden. From Table 2 we can see that also this characteristic links well the two classifications.

Unfortunately, there are six exceptions to such a well-built formal construction. On the one hand, the languages defined by the orphan patterns of rules 104, 134, and 152 seem to be non-slender but none of these rules has any orbit with \( \tau \geq 3 \); on the other hand, the languages defined by the orphan patterns of rules 9, 54, and 126 seem to be finite, but these rules have orbits with \( \tau \geq 3 \). We still do not know the reason for such a discrepancy, but possibly a more accurate study of the two classifications may explain it.

6. **Conclusions and future work**

In this paper, we have proposed two different classifications of Cellular Automata local rules. The first classification is a variant of the one proposed by Chua, and
Table 3: The rules with no orphans can be divided into two groups: those surjective for \( L \) finite (upper half of the table) and those non-surjective for \( L \) finite (lower half of the table). Look how the classification of rules according to their complexity index and their periodic orbits matches perfectly the classification made by using the Formal Language theory.

<table>
<thead>
<tr>
<th>No orphan patterns</th>
<th>Only orbits with ( \tau=1 ) and/or ( \tau=2 )</th>
<th>Also orbits with ( \tau\geq3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>15 51 170 204</td>
<td>30 45 60 90 106 154</td>
</tr>
<tr>
<td></td>
<td>105 150</td>
<td></td>
</tr>
</tbody>
</table>

it is based on the characteristics of the periodic orbits obtained by using all bit strings with \( L \leq 15 \), while the second classification is based on the languages defined by the orphan patterns which are found in the Gardens of Eden. We cannot overemphasize that these two classifications refer to two different characteristics of CA rules. For this reason, it is remarkable that the correspondence explained above exists. In the near future, we plan to give a rigorous explanation for such a phenomenon and find the reasons behind the six exceptions aforementioned.

References


Hard and Fuzzy \( c \)-Means Clustering Algorithms with Geodesic Dissimilarity

Yuki Kanzawa†, Yasunori Endo ‡ and Sadaaki Miyamoto †

†Shibaura Institute of Technology, Japan
Email: kanzawa@sic.shibaura-it.ac.jp
‡University of Tsukuba, Japan

Abstract—In this paper, the geodesic distance is applied to relational clustering methods. First, it is shown that conventional methods are based on respective three types of relational clustering algorithms among nine ones, and the six rests of the nine ones with the geodesic distance are proposed. Second, geodesic dissimilarity is proposed by assigning the power of the Euclidean distance to the weight of the neighborhood graph of data. Numerical examples show that the proposed geodesic-dissimilarity-based relational clustering algorithms successfully cluster the data that conventional squared-Euclidean-distance-based ones cannot.

1. Introduction

Fuzzy \( c \)-means (FCM) [1] is a well-known fuzzy clustering method that is derived from hard \( c \)-means (HCM), also called as \( k \)-means. Among the many FCM variants proposed thus far, one is the FCM algorithm based on the concept of regularization by entropy [2]. This algorithm is called entropy regularized FCM (eFCM) and is discussed not only because of its usefulness but also because of its mathematical relationships with other techniques. We call the FCM proposed in [1] standard FCM (sFCM) in order to distinguish it from eFCM.

For HCM, sFCM, and eFCM, similar clustering models for relational data can be developed. It should be noted that neither object data nor cluster centers are available in relational clustering. Hence, the distance between data points and cluster centers that appear in HCM, sFCM, and eFCM cannot explicitly be computed. There are two methods to overcome this problem: one is to restrict the solution space and the other is to implicitly compute the object data and cluster centers. The former method is called hard \( c \)-medoids (HCMdd), standard fuzzy \( c \)-medoids (sFCMdd) [3], or entropy regularized fuzzy \( c \)-medoids (eFCMdd), respectively, based on HCM, sFCM, or eFCM. The latter is called relational HCM (RHCMM), standard relational FCM (sRFCM) [4], or entropy regularized relational FCM (eRFCM) [5], respectively. The kernelization of HCM, sFCM, and eFCM, called kernel HCM (K-HCM) [6], kernel sFCM (K-sFCM), and kernel eFCM (K-eFCM) [7], can be also applied to relational data if a dissimilarity-based kernel function, for example, a Gaussian kernel, is used.

The correct identification of clusters depends on the definition of dissimilarity. The choice of the dissimilarity measure determines the cluster shape, and therefore, it determines the success of a clustering algorithm on the specific application domain. As one such choice, the geodesic distance has been applied to sFCMdd [8], K-eFCM [9], and RHCMM [10]. One of our two objectives is to apply the geodesic distance to other six types of relational clustering methods such as HCMdd, eFCMdd, K-HCM, K-sFCM, sRFCM, and eRFCM.

The geodesic distance is computed as the total weight of the shortest weighted path on the neighborhood graph of a data set, where we have the degree of freedom, that is, the number or maximal distance of a neighborhood, and the weight of the edge. While the Euclidean distance is usually assigned as the weight of the edge as in [8] and [9], a density scaling is used in [10]. In this paper, we consider another weight, the power of Euclidean distance, typically the squared-Euclidean distance. The considered measure is no longer the geodesic distance but geodesic dissimilarity because it does not satisfy the triangular inequality. The other of our two objectives is to apply this geodesic dissimilarity to nine types of relational clustering methods such as HCMdd, sFCMdd, eFCMdd, K-HCM, K-sFCM, K-eFCM, RHCMM, sRFCM, and eRFCM.

The remainder of this paper is organized as follows. In the second section, we define some notations, and introduce some relational clustering algorithms and the concept of geodesic distance; these are used in our proposed methods. In the third section, we propose applying the geodesic distance to six types of relational clustering methods, and also propose new geodesic dissimilarity that can be applied to nine types of relational clustering algorithms. In the fourth section, we present some numerical examples. In the last section, we conclude this paper.

2. Preliminaries

2.1. Fuzzy Clustering

In this subsection, we introduce nine conventional methods of relational clustering; these are used in our proposed methods that is described in the next section. The introduced methods are classified according to the three original clustering algorithms — HCM, sFCM, or eFCM — or according to how the relational data is used — by using medoids, by transforming the optimization problem, or by using dissimilarity-based kernel function.

For a given data set \( x = \{x_i \mid i \in \{1, \ldots, N\}\} \), HCMdd, sFCMdd, eFCMdd, RHCMM, sRFCM, and eRFCM assume that the dissimilarity data matrix \( R \in \mathbb{R}^{N \times N} \) is given, and K-HCM, K-sFCM, and K-eFCM assume that the kernel matrix \( K \in \mathbb{R}^{N \times N} \) is given. The membership by which \( x_i \) belongs to the \( j \)-th cluster is denoted by \( u_{i,j} \) (\( i \in \{1, \ldots, N\}, j \in \{1, \ldots, C\} \)) and the set of \( u_{i,j} \) is denoted by \( u \in \mathbb{R}^{N \times C} \); this is called the partition matrix.

Hard \( c \)-means (HCM) is the algorithm obtained by solving the following optimization problem:

\[
\begin{align*}
\text{minimize} & \quad J_{HCM}(u, v) \\
\text{subject to} & \quad \sum_{j=1}^{C} u_{i,j} = 1, \\
\end{align*}
\]

where

\[
J_{HCM}(u, v) = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{i,j} d_{i,j},
\]

\[
d_{i,j} = ||x_i - v_j||^2.
\]
Standard fuzzy c-means (sFCM) is the algorithm obtained by solving the following optimization problem:

\[ \text{minimize} \sum_{i=1}^{N} \sum_{j=1}^{C} u_{i,j}^m d_{i,j} \tag{5} \]

subject to Eq. (2) and (4). Entropy regularized fuzzy c-means (eFCM) is the algorithm obtained by solving the following optimization problem:

\[ \text{minimize} J_{\text{HCM}} + \lambda^{-1} \sum_{i=1}^{N} \sum_{j=1}^{C} u_{i,j} \log(u_{i,j}) \tag{6} \]

subject to Eq. (2) and (4).

Hard c-medoids (HCMdd) is the algorithm obtained by solving the optimization problem (1) subject to Eq. (2), (4), and (7).

\[ v_j \in x. \tag{7} \]

Standard fuzzy c-medoids (sFCMdd) is the algorithm obtained by solving the optimization problem (5) subject to Eq. (2), (4), and (7). Entropy regularized fuzzy c-medoids (eFCMdd) is the algorithm obtained by solving the optimization problem (6) subject to Eq. (2), (4), and (7).

Kernel hard c-means (K-HCM) is the algorithm obtained by solving optimization problem (1) subject to Eq. (2) and (8).

\[ d_{i,j} = \| \Phi(x_i) - W_j \|_2. \tag{8} \]

Kernel standard fuzzy c-means (K-sFCM) is the algorithm obtained by solving optimization problem (5) subject to Eq. (2) and (8). Kernel entropy regularized fuzzy c-means (K-eFCM) is the algorithm obtained by solving optimization problem (6) subject to Eq. (2) and (8).

Relational hard c-means (RHCM) is the algorithm obtained by solving the following optimization problem:

\[ \text{minimize} J_{\text{RHCM}}, \tag{9} \]

where

\[ J_{\text{RHCM}}(u) = \sum_{j=1}^{C} \sum_{i=1}^{N} \sum_{k=1}^{N} u_{i,j} u_{k,j} r_{1,k} / \left( 2 \sum_{i=1}^{N} u_{i,j} \right), \tag{10} \]

subject to Eq. (2), and \( r_{1,k} \) is given. Standard relational fuzzy c-means (sRFCM) is the algorithm obtained by solving the following optimization problem:

\[ \text{minimize} \sum_{j=1}^{C} \sum_{i=1}^{N} \sum_{k=1}^{N} u_{i,j}^m u_{k,j} r_{1,k} / \left( 2 \sum_{i=1}^{N} u_{i,j}^m \right), \tag{11} \]

subject to Eq. (2), and \( r_{1,k} \) is given. Entropy regularized relational fuzzy c-means (eRFCM) is the algorithm obtained by solving the following optimization problem:

\[ \text{minimize} J_{\text{RHCM}}(u) + \lambda^{-1} \sum_{i=1}^{N} \sum_{j=1}^{C} u_{i,j} \log(u_{i,j}) \tag{12} \]

subject to Eq. (2), and \( r_{1,k} \) is given.

HCMdd, sFCMdd, eFCMdd, RHCM, sRFCM, eRFCM, K-HCM, K-sFCM, and K-eFCM are given by the following algorithm.

Algorithm 1

1. Set \( C \); set \( m \) for sFCMdd, K-sFCM, and eRFCM; set \( \lambda \) for eFCMdd, K-eFCM, and eRFCM; set \( r \) for RHCM, sRFCM, and eRFCM; set \( K \) for K-HCM, K-sFCM, and K-eFCM; and set \( u \).

2. Calculate

\[ v_j = \text{arg min}_k \sum_{i=1}^{N} u_{i,j} d_{i,k} \tag{13} \]

for HCMdd and eFCMdd;

\[ v_j = \arg \min_k \sum_{i=1}^{N} u_{i,j}^m d_{i,k} \tag{14} \]

for sFCMdd;

\[ v_j = \sum_{i=1}^{N} u_{i,j} e_i / \left( \sum_{i=1}^{N} u_{i,j} \right) \tag{15} \]

for RHCM, eRFCM, K-HCM, and K-eFCM; and

\[ v_j = \sum_{i=1}^{N} u_{i,j}^m e_i / \left( \sum_{i=1}^{N} u_{i,j}^m \right) \tag{16} \]

for sRFCM and K-sFCM.

3. Calculate the membership

\[ u_{i,j} = \begin{cases} 1 & (j = \text{arg min}_k \{ d_{i,k} \}) \\ 0 & (\text{otherwise}) \end{cases} \tag{17} \]

for HCMdd;

\[ u_{i,j} = 1 / \left( \sum_{k=1}^{C} \left( \frac{d_{i,j}}{d_{i,k}} \right)^{(m-1)/2} \right) \tag{18} \]

for sFCMdd;

\[ u_{i,j} = \frac{\exp(-\lambda d_{i,j})}{\sum_{k=1}^{C} \exp(-\lambda d_{i,k})} \tag{19} \]

for eFCMdd;

\[ u_{i,j} = 1 / \left( \sum_{k=1}^{C} \left( \frac{R W_{j,i} - W_{j,k} R W_{k,i}}{(R W_{k,i} - W_{j,k} R W_{k,i})} \right)^{(m-1)/2} \right) \tag{20} \]

for K-sFCM;

\[ u_{i,j} = \frac{\exp(-\lambda(e_i - W_{j,k})^T K(e_i - W_{j,k}))}{\sum_{k=1}^{C} \exp(-\lambda(e_i - W_{j,k})^T K(e_i - W_{j,k}))} \tag{21} \]

for K-eFCM;

\[ u_{i,j} = \frac{\exp(-\lambda((R W_{j,i} - W_{j,k} R W_{k,i}))}{\sum_{k=1}^{C} \exp(-\lambda((R W_{j,i} - W_{j,k} R W_{k,i}))} \tag{22} \]

for sRFCM.

4. If \((u, v)\) is convergent, terminate this algorithm. Otherwise, return to STEP 2.

2.2. Geodesic Distance

In this subsection, the two types of geodesic distance, \(k\)-geodesic distance and \(\epsilon\)-geodesic distance, are introduced.

The \(\epsilon\)-neighborhood of a point \(x \in X\) is defined as \(N_\epsilon(x) = \{ z \in X | \| x - z \|_2 \leq \epsilon \}\). The \(k\)-neighborhood of a point \(x \in X\) is the set of \(k\) closest points to \(x\) in the \(l_2\) norm sense: \(N_k(x) \subset X\) such that \(|N_k(x)| = k\) and \(\max_{x \in N_k(x)} \| x - z \|_2 \leq \min_{x \in \mathbb{R}^n} \| x - z \|_2\).

The \(k\)-neighborhood graph of \(X\) is an undirected graph whose the vertices are \(X\) and whose the edges \((x_i, x_j)\) exist if \(x_i \in N_k(x_j)\) or \(x_j \in N_k(x_i)\). The \(\epsilon\)-neighborhood graph of \(X\) is an undirected graph whose the vertices are \(X\) and whose the edges \((x_i, x_j)\) exist if \(x_i \in N_\epsilon(x_j)\) or \(x_j \in N_\epsilon(x_i)\). Assume a symmetric matrix \(D \in \mathbb{R}^{N \times N}\) of non-negative weights on the edges of \(G_\epsilon\). The \(k\)-geodesic distance from any point \(x\) to any point \(\tilde{x}\) is defined as the
total weight of the shortest weighted path from \(x\) to \(\tilde{x}\) on \(G_k\), which is denoted by \(\delta_{G_k,D}(x, \tilde{x})\). The \(\varepsilon\)-geodesic distance from any point \(x\) to any point \(\tilde{x}\) is defined as the total weight of the shortest weighted path from \(x\) to \(\tilde{x}\) on \(G_{\varepsilon}\), which is denoted by \(\delta_{G_{\varepsilon},D}(x, \tilde{x})\).

3. Proposed Method

One of our two objectives in this paper is to show that conventional geodesic-distance-based clustering methods are based on respective three types of relational clustering algorithms among nine ones, and to propose the six rests of the nine ones with the geodesic distance. The other is to propose a new dissimilarity based on the geodesic distance, and to apply conventional relational clustering methods, as described in the second section.

3.1. Six types of Geodesic-Distance-based Clustering

The geodesic distance has been applied to some relational clustering algorithms [8]–[10]. sFCMdd in [8], K-eFCM in [9], and RHCM in [10] were used with the geodesic distance. The survey of relational clustering methods with the geodesic distance is summarized in Table 1. The symbol “–” in this table indicates that we propose these methods by applying the geodesic distance, that is, HCMdd, eFCMdd, K-HCM, K-sFCM, sRFCM, and eRFCM (Algorithm 1).

<table>
<thead>
<tr>
<th>Method</th>
<th>Meods</th>
<th>Relational</th>
<th>Kernel</th>
</tr>
</thead>
<tbody>
<tr>
<td>HCM</td>
<td>–</td>
<td>–</td>
<td>[10]</td>
</tr>
<tr>
<td>sFCM</td>
<td>[8]</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>eFCM</td>
<td>–</td>
<td>–</td>
<td>[9]</td>
</tr>
</tbody>
</table>

3.2. New Dissimilarity based on Geodesic Distance

The proposed measure is no longer the geodesic distance but geodesic dissimilarity because it does not satisfy the triangular inequality. The proposed geodesic dissimilarity \(\delta_{G_{\varepsilon},D}\) has the following properties. \(\delta_{G_{\varepsilon},D}(x_i, x_j)\) is less than the squared-Euclidean distance between them if there exists at least one point \(x_k\) in the sphere with center \((x_i + x_j)/2\) and radius \(||x_i - x_j||/2\), as shown in Fig. 1.

![Figure 1](image-url)

Figure 1: \(\delta_{G_{\varepsilon},D}(x_i, x_j)\) is less than the squared-Euclidean distance between them if there exists a point \(x_k\) in the sphere with center \((x_i + x_j)/2\) and radius \(||x_i - x_j||/2\), as shown in Fig. 1.

The geodesic distance is computed as the total weight of the shortest weighted path on the neighborhood graph of a data set, where we have the degree of freedom, that is, the number or the maximal distance of the neighborhood, and the weight of the edge. While the Euclidean distance is usually assigned as the weight of the edge in [8] and [9] as

\[
D_{i,j} = ||x_i - x_j||_2,
\]

a density scaling is used in [10]. In this paper, we propose another weight, the power of Euclidean distance:

\[
D_{i,j}^{(q)} = ||x_i - x_j||_2^q
\]

with a parameter \(q\). If \(q = 2\), the weight is the squared-Euclidean distance

\[
D_{i,j}^{(2)} = ||x_i - x_j||_2^2.
\]

The proposed measure is no longer the geodesic distance but geodesic dissimilarity because it does not satisfy the triangular inequality. The proposed geodesic dissimilarity \(\delta_{G_{\varepsilon},D}\) has the following properties. \(\delta_{G_{\varepsilon},D}(x_i, x_j)\) is less than the squared-Euclidean distance between them if there exists at least one point \(x_k\) in the sphere with center \((x_i + x_j)/2\) and radius \(||x_i - x_j||/2\), as shown in Fig. 1. This sphere, the border being where \(\delta_{G_{\varepsilon},D}(x_i, x_j)\) is less than \(||x_i - x_j||/2\), is inflated in the direction orthogonal to the segment \(x_i - x_j\) with \(q > 2\), deflated with \(q < 2\), and degenerate with \(q \leq 1\).

![Figure 2](image-url)

Figure 2: This sphere, the border being where \(\delta_{G_{\varepsilon},D}(x_i, x_j)\) is less than \(||x_i - x_j||/2\), is inflated in the direction orthogonal to the segment \(x_i - x_j\) with \(q > 2\), deflated with \(q < 2\), and degenerate with \(q \leq 1\). We propose this geodesic dissimilarity for use in clustering algorithms (Algorithm 1).

\[
s \text{points}
\]

![Figure 3](image-url)

Figure 3: \(\delta_{G_{\varepsilon},D}(x_i, x_j)\) tends to zero as \(s \to \infty\) if there exist \(s\) points on the segment \(x_i - x_j\).

4. Numerical Example

In this section, we show some examples of clustering using our proposed methods. The fuzzifier parameters are fixed as \(m = 2\) in sFCM-based methods and \(\lambda = 1/2\) in eFCM-based methods. As a kernel function, a Gaussian kernel is selected with the kernel parameter \(\sigma^2 = 0.002\). In each example, 100 trials for the proposed algorithm with randomly different initializations are tested and the solu-
tion with the minimal objective function value is selected as the final result. We consider clustering the data shown in Fig. 4 into two moon-shaped clusters. This data set is constructed using 300 elements in the two-dimensional Euclidean space. First, we see that all algorithms (Algorithm 1) with the squared-Euclidean distance as the dissimilarity fail to cluster correctly; the result of sRFCM-based methods is shown in Fig. 5, where the plus symbol indicates one cluster and the cross symbol indicates the other. This figure indicates that the each inside edges of the moons are mis-clustered with each other.

Next, we use the proposed geodesic dissimilarity with $q = 2$ instead of the squared-Euclidean distance in Algorithm 1 and we obtain the desired clustering results, as shown in Fig. 6. Thus, this example shows that the proposed geodesic-dissimilarity-based algorithms achieve successful clustering results for the data for which the conventional squared-Euclidean-distance-based algorithms fail. In future works, (1) we intend to investigate the property of the parameter in the proposed geodesic dissimilarity, and (2) test the differences between clustering features based on the differences among relational clustering algorithms.

5. Conclusion

In this paper, we considered applying the geodesic distance to clustering methods. First, we showed that conventional methods were based on respective three types of relational clustering algorithms among nine ones, and we proposed the six rests of the nine ones with the geodesic distance. Second, we proposed the geodesic dissimilarity by assigning the power of the Euclidean distance to the weight of the neighborhood graph of data. Through numerical examples, we found that the proposed geodesic-dissimilarity-based relational clustering algorithms achieve successful clustering results for the data for which the conventional squared-Euclidean-distance-based algorithms fail.

References

Kernelized Fuzzy \(c\)-Means Clustering for Uncertain Data with \(L_1\)-Regularization Term of Penalty Vectors Using Explicit Mapping

ENDO Yasunori‡, TAKAYAMA Isao‡, HAMASUNA Yukihiro§† and MIYAMOTO Sadaaki‡

‡Department of Risk Engineering, University of Tsukuba, Tennodai 1-1-1, Tsukuba, Ibaraki, 305-8573, Japan Email: {endo@, yhama@soft, miyamoto@}.tsukuba.ac.jp
‡Graduate School of Systems and Information Engineering, University of Tsukuba, Tennodai 1-1-1, Tsukuba, Ibaraki, 305-8573, Japan Email: takayama@soft.risk.tsukuba.ac.jp
§Research Fellow of the Japan Society for the Promotion of Science Email: yhama@soft.risk.tsukuba.ac.jp

Abstract—Recently, fuzzy \(c\)-means clustering with kernel functions is remarkable in the reason that these algorithms can handle datasets which consist of some clusters with nonlinear boundary. However, the algorithms have the following problems: (1) the cluster centers cannot be calculated explicitly, (2) it takes long time to calculate clustering results. By the way, we have proposed the clustering algorithms with regularization terms of penalty vectors to handle uncertain data. In this paper, we propose new clustering algorithms with \(L_1\)-regularization term by introducing explicit mapping of kernel functions to solve the following problems.

1. Introduction

In recent years, according to large-scale and complicated data of computer systems by progress of the hardware technology, the importance of data analysis techniques has been increasing. Clustering, one of the data classification techniques, is the method without any external criterion and it classifies data automatically. One of most typical and useful clustering is fuzzy \(c\)-means clustering (FCM) [1] and we consider FCM in this paper.

Now, studies of clustering methods for detecting nonlinear boundary have been widely discussed and kernel trick [2, 3] is focused as a very useful tool to handle datasets which consist of some clusters with nonlinear boundary. The kernel trick uses some function called “kernel function”.

Here, we consider a mapping \(\phi : \mathbb{R}^p \rightarrow \mathbb{R}^s\) \((p \ll s)\). The kernel function is an operator \(K : \mathbb{R}^p \times \mathbb{R}^p \rightarrow \mathbb{R}\) which satisfies \(K(x, y) = \langle \phi(x), \phi(y) \rangle\). We can calculate the value of the inner product in the high-dimensional feature space with low-cost calculation by the kernel functions.

However, special problems frequently arise when we introduce kernel functions into FCM. First, the cluster centers cannot be calculated directly. The reason is that the mapping \(\phi\) can be not represented explicitly in general. The second problem is caused by the first one, that is, it takes long time to calculate clustering results. The procedure to calculate the belongingness or membership grades of each data to clusters becomes complicated. To solve the above problems, Miyamoto et al. introduced an explicit mapping into kernel trick and succeeded to calculate the cluster centers directly [4].

By the way, the data we handle have uncertainties in many cases. Until now, we have two methodologies to analyze such data, deterministic and parametric ones.

In deterministic methodology, the uncertainty introduced data is represented as an interval. Distance or dissimilarities are the most essential measures for data analysis and therefore, the measures between intervals must be defined when we handle uncertainty of data as interval. There are many kinds of such measures, e.g., Hausdorff distance. However, we have two problems when we try to classify the dataset of such data. First, we don’t have clear indication to select which measures to use. Second, we can’t discuss clustering in the framework of optimization with such measures.

On the other hand, the uncertainty is represented by some probabilistic density function (PDF) in parametric methodology. In this methodology, the clustering is regarded as a method to determine some parameters of the PDF. However, the methodology have at least one problem, that is, selection which type of PDFs.

Therefore, we have proposed and developed the concept of tolerance in Ref. [5, 6, 7] to solve the above problems. In our concept, tolerance vectors [5] and penalty ones [6, 7] play main role. Uncertain data is allowed to allocate any position by those vectors as far as the constraints for those vectors are satisfied and the position is derived as an optimal solution of a given objective function. Hence, we can say that this concept is in the framework of methodology of soft computing. Penalty vectors are similar to tolerance ones and the methods using penalty vectors become more flexible than tolerance vectors because no constraint for the vectors is needed. Moreover, the concept has been developed by using kernel trick in Ref. [8]. The method can classify datasets which consist of clusters of uncertain data with nonlinear boundary.

In this paper, we try to construct new FCMs for uncertain data with \(L_1\)-regularization term of penalty vectors using explicit kernel mapping (EK-FCMP\(_p\)). The proposed methods have the following advantages:

1. The method can classify datasets which consist of clusters of uncertain data with nonlinear boundary because of kernel trick.

2. It takes shorter time to calculate clustering result than the conventional method because of explicit mapping.
2. Kernelized FCM for Uncertain Data with $L_1$-Regularization Term of Penalty Vectors Using Explicit Mapping

In this section, we try to construct FCM for uncertain data with $L_1$-regularization term of penalty vectors using explicit mapping of kernel functions (EK-FCMP$_{L_1}$). First, we describe kernelized FCM for uncertain data with $L_1$-regularization term of penalty vectors (K-FCMP$_{L_1}$). Next, we introduce an explicit mapping of kernel functions into K-FCMP$_{L_1}$ to construct EK-FCMP$_{L_1}$. Last, we show an algorithm of EK-FCMP$_{L_1}$.

2.1. Kernel Functions

First, we define some symbols to introduce kernel functions. A mapping from the pattern space to the feature space is expressed as $\phi : \mathbb{R}^p \rightarrow \mathbb{R}^s (p \ll s)$. Each data in the feature space is denoted $\phi(x_k) = \phi^k = (\phi^k_1, \ldots, \phi^k_s)^T \in \mathbb{R}^s$ and the dataset $\mathcal{X}^\phi = \{\phi^1, \ldots, \phi^m\}$ is given. Each cluster $C_i (i = 1, \ldots, c)$ has a cluster center $\phi = (\phi^1_i, \ldots, \phi^c_i)^T \in \mathbb{R}^s$. $\phi$ means a set of cluster centers $\{\phi^1, \ldots, \phi^c\}$. A membership grade for $x_k^\phi$ to $C_i$, which means belongingness of $x_k^\phi$ to $C_i$, is denoted by $u_{ki}$. $U$ means a penalty matrix $(u_{ki})_{1 \leq k \leq c, 1 \leq i \leq c}$. Moreover, $\phi^\phi_k = (\phi^\phi_{11_k}, \ldots, \phi^\phi_{1c_k})^T \in \mathbb{R}^s$ and $w_k \in [0, +\infty)$ mean a penalty vector and a weight of the penalty vector $\phi^\phi_k$, respectively. $\Delta$ and $W$ mean a set of penalty vectors $\{\phi^\phi_1, \ldots, \phi^\phi_c\}$ and a set of weights $\{w_1, \ldots, w_n\}$, respectively.

Now, a kernel function $K : \mathbb{R}^p \times \mathbb{R}^p \rightarrow \mathbb{R}$ satisfies the following relation:

$$K(x, y) = \langle \phi(x), \phi(y) \rangle$$

From Mercer’s theorem [10], $K$ is a continuous symmetric non-negative definite kernel function if a mapping $\phi$ which satisfies the above relation exists. We should notice that $\phi$ is not explicit.

2.2. Objective Functions

Kernelized fuzzy c-means clustering (K-FCM) is FCM in which a kernel function is introduced into dissimilarity. An extended K-FCM by introducing penalty vectors of the quadratic term have been proposed in Ref. [8]. Here, we consider the objective function $J_{kep}$ of the standard EK-FCMP$_{L_1}$ (sEK-FCMP$_{L_1}$) and another function $J_{kep}$ of the entropy based EK-FCMP$_{L_1}$ (eEK-FCMP$_{L_1}$) as follows:

$$J_{kep1}(U, \Delta^\phi, V^\phi) = \sum_{k=1}^n \sum_{i=1}^c (u_{ki})^m d_{ki} + \sum_{k=1}^n w_k \|\phi^\phi_k\|_1,$$  \(2\)

$$J_{kep2}(U, \Delta^\phi, V^\phi) = \sum_{k=1}^n \sum_{i=1}^c u_{ki} d_{ki} + \lambda^{-1} \sum_{k=1}^n \sum_{i=1}^c u_{ki} \log u_{ki} + 2 \sum_{k=1}^n w_k \|\phi^\phi_k\|_1.$$  \(3\)

The dissimilarity $d_{ki}$ is the square of the norm as follows.

$$d_{ki} = d(x_k + \delta_k, v_i) = \|x_k + \delta_k - v_i\|^2$$

$$= \sum_{j=1}^c (x_{kj} + \delta_{kj} - v_{ji})^2.$$  \(4\)

$\| \cdot \|_1$ means the $L_1$-norm.

2.3. Optimal Solutions of sEK-FCMP$_{L_1}$

We can derive the optimal solutions of the membership grades $u_{ki}$ and the cluster centers $\phi$ by using Lagrange multiplier. The optimal solutions of standard K-FCMP$_{L_1}$ (sEK-FCMP$_{L_1}$) are as follows:

$$u_{ki} = \frac{(1/d_{ki})^{1/m}}{\sum_{j=1}^c (1/d_{kj})^{1/m}},$$  \(5\)

$$\phi = \frac{1}{\sum_{k=1}^n u_{ki})^m (\phi^\phi_k + \delta^\phi_k)}.$$  \(6\)

Next, we derive the optimal solution of penalty vectors $\delta^\phi_k$. Now, we define a semi-objective function $J_k(\delta^\phi_k)$.

$$J_k(\delta^\phi_k) = \sum_{i=1}^c (u_{ki})^m (x^\phi_{ki} + \delta^\phi_k - v_i)^2 + 2w_k \|\delta^\phi_k\|^2.$$  \(7\)

The $\delta^\phi_k$ which minimizes $J_k$ is the optimal solution of $J_{kep}$, because $J_{kep}(U, \Delta^\phi, V^\phi) = \sum_{k=1}^c J_k(\delta^\phi_k)$. However, the term $\|\delta^\phi_k\|^2$ is piecewise linear and $J_k$ cannot be differentiated at $\delta^\phi_k = 0$. Therefore, we have to consider another way to find the solutions.

Here, we consider the value of $\frac{\partial J_k}{\partial \delta^\phi_k}$. We can find the optimal solution as the point at which the value of $\frac{\partial J_k}{\partial \delta^\phi_k}$ turns into positive from negative. Thus, We consider the two cases, $\delta^\phi_k > 0$ and $\delta^\phi_k < 0$.

We denote the semi-objective functions when $\delta^\phi_k > 0$ and $\delta^\phi_k < 0$ as $J'_k$ and $J''_k$, respectively and show the functions as follows:

$$J'_k(\delta^\phi_k) = \sum_{i=1}^c (u_{ki})^m (x^\phi_{ki} + \delta^\phi_k - v_i)^2 + 2w_k \delta^\phi_k,$$  \(8\)

$$J''_k(\delta^\phi_k) = \sum_{i=1}^c (u_{ki})^m (x^\phi_{ki} + \delta^\phi_k - v_i)^2 - 2w_k \delta^\phi_k.$$  \(9\)

Therefore, we can get $\delta^\phi_k$ and $\delta^\phi_k$ from $\frac{\partial J'_k}{\partial \delta^\phi_k} = 0$ and $\frac{\partial J''_k}{\partial \delta^\phi_k} =$
0 as follows:
\[
\begin{align*}
\delta^+_{kj} &= -\left( \sum_{i=1}^c (u_{ki})^m (x_k^i - v_j^i) + w_k \right), \\
\delta^-_{kj} &= -\left( \sum_{i=1}^c (u_{ki})^m (x_k^i - v_j^i) - w_k \right) + \sum_{i=1}^c (u_{ki})^m . 
\end{align*}
\]
Therefore, we obtain the optimal solution of \(\delta^o_{kj}\) as follows:
\[
\delta^o_{kj} = \begin{cases} 
\delta^+_{kj}, & (\delta^+_{kj} > 0) \\
\delta^-_{kj}, & (\delta^-_{kj} < 0) \\
0, & \text{(otherwise)}
\end{cases}
\]

### 2.4. Optimal Solutions of \(eEK-FCMP_L\)

We can derive the optimal solutions by using Lagrange multiplier. The optimal solutions of entropy based K-FCMP\(_L\) are as follows:
\[
\begin{align*}
u_i &= \frac{\exp(-\lambda d_k)}{\sum_{i=1}^n \exp(-\lambda d_k)}, \\
v_i^o &= \frac{\sum_{k=1}^n u_k (x_k^i + \delta_k^o)}{n}, \\
\delta_k^o &= \begin{cases} 
\delta^+_{kj}, & (\delta^+_{kj} > 0, m = 1) \\
\delta^-_{kj}, & (\delta^-_{kj} < 0, m = 1) \\
0, & \text{(otherwise)}
\end{cases}
\end{align*}
\]

We can formaly obtain the optimal solutions as above. However, we cannot calculate \(v_i^o\) and \(\delta_k^o\) directly by this method. The reason is that the map \(\phi\) is not explicit and we have to use \(x_k\), not \(x_k^\phi\). Hence, We introduce an explicit mapping into K-FCMP\(_L\) to calculate \(v_i^o\) directly.

### 2.5. Optimal Solutions with Explicit Mapping of Kernel Functions

In this paragraph, we introduce an explicit mapping of kernel functions [4]. As above, the kernel function \(K\) satisfies \(K(x, y) = \langle \phi(x), \phi(y) \rangle = \langle x^\phi, y^\phi \rangle\). Now, we consider a kernel function with the kernel matrix \(K = (K_{ij}) = (K(x_i, x_j)).\) As \(K\) is a positive symmetric matrix, we can introduce an inner product \(\langle x, y \rangle_K = x^T K y\). We assume that \(\phi : X \rightarrow \mathbb{R}^c\) and \(\phi(x_k) = x_k^\phi = e_k\). Here \(e_k = (e_{k1}, \ldots, e_{kn})\) is a unit vector and \(e_j\) is Kronecker delta. Because \(\phi\) can be represented explicitly, we call \(\phi\) using the kernel matrix \(K\) explicit mapping. Therefore, the inner product is as follows:
\[
\langle e_k, e_j \rangle_K = K_{kj}.
\]

Thus, the dissimilarity \(d_{ki}\) can be calculated as follows:
\[
\begin{align*}
d_{ki} &= d(x_k^\phi + \delta_k^o, v_j^o) = d(e_k + \delta_k^o, v_j^o) \\
&= \langle e_k + \delta_k^o - v_j^o, e_k + \delta_k^o - v_j^o \rangle_K \\
&= (e_k + \delta_k^o - v_j^o)^T K (e_k + \delta_k^o - v_j^o).
\end{align*}
\]
We can calculate the optimal solutions by using the above inner product and dissimilarity and replacing \(x_k\) and \(e_k\) by \(e_k\) and \(e_{kj}\), respectively. We had to omit the details for want of space.

### 2.6. Algorithm

Here, we construct an algorithm for \(eEK-FCMP_L\) using the above optimal solutions with the explicit mapping as follows.

**Algorithm 1 (sEK-FCMP\(_L\) and eEK-FCMP\(_L\))**

1. **Step 1** Set the initial values \(U\) and \(\Delta^o\). Generate the kernel matrix \(K\).
2. **Step 2** Calculate \(V^o\) by the optimal solutions with the explicit mapping on fixing \(U\) and \(\Delta^o\).
3. **Step 3** Calculate \(U\) by the optimal solutions with the explicit mapping on fixing \(\Delta^o\) and \(V^o\).
4. **Step 4** Calculate \(\Delta^o\) by the optimal solutions with the explicit mapping on fixing \(V^o\) and \(U\).
5. **Step 5** Finish if the solutions are convergent, else go back to Step 2.

### 3. Numerical Examples

We show numerical examples in Fig. 1 ~ Fig. 8. The data size is 133 and the pattern space is \([0, 1] \times [0, 1]\). We use Gaussian kernel:
\[
K(x, y) = \exp(-\beta ||x - y||^2).
\]
We use the same initial values at all case and give the same weight of penalty vectors to all data.

Fig. 1 ~ Fig. 4, and Fig. 5 ~ Fig. 8 are the results by sEK-FCMP\(_L\) and \(eEK-FCMP_L\), respectively. We also show the cluster centers by using kernel principal component analysis (K-PCA) [12].

In Table 1 we show the range of \(w_k\) when desirable clustering results are obtained. In the range, the larger \(w_k\) gets, the faster the algorithm converges and the clearer the shape of the fuzzy classification functions becomes. In addition, we get a desirable clustering results by K-sFCMP\(_L\) even when \(w_k\) is very small, and the range of \(w_k\) hardly changes even if a value of \(\beta\) changes. Thus, we can say that K-sFCMP\(_L\) is robust for the parameters.

### 4. Conclusion

In this paper, we developed FCM for uncertain data with quadratic regularization term of penalty vectors by introducing explicit mapping of kernel functions. The proposed algorithm has the following features:

1. The algorithm can handle datasets which consist of some clusters with nonlinear boundary.
Table 1: The range of $w_k$ when obtaining desirable clustering results

<table>
<thead>
<tr>
<th>Method</th>
<th>$\beta$</th>
<th>$m$</th>
<th>$10^{-6} \leq w_k \leq$</th>
</tr>
</thead>
<tbody>
<tr>
<td>sEK-FCMP$_L$</td>
<td>70</td>
<td>1.2</td>
<td>$0.91$</td>
</tr>
<tr>
<td>sEK-FCMP$_L$</td>
<td>40</td>
<td>1.2</td>
<td>$0.94$</td>
</tr>
<tr>
<td>eEK-FCMP$_L$</td>
<td>70</td>
<td>10</td>
<td>None</td>
</tr>
<tr>
<td>eEK-FCMP$_L$</td>
<td>40</td>
<td>5.0</td>
<td>$0.94 \leq w_k$</td>
</tr>
</tbody>
</table>

2. Because of explicit mapping of kernel functions, we reduce the calculation time.

3. The algorithm becomes more simple than the conventional ones.

In the forthcoming paper, we will discuss how to find appropriate parameters and apply to real data.

Acknowledgment

This study is partly supported by the Grant-in-Aid for Scientific Research (C) (Project No.21500212) from the Ministry of Education, Culture, Sports, Science and Technology, Japan.

References


A Classification System based on Collaboration of Adaptive Resonance Theory Maps and Learning Vector Quantization

Yoko Enosawa†, Haruna Matsushita‡, and Toshimichi Saito†

†EE Dept., Hosei Univ., Tokyo 184-8584, Japan
yoko.enosawa.w4@gs-eng.hosei.ac.jp, {haruna.matsushita.pd, tsaito}@k.hosei.ac.jp

Abstract—This paper studies a novel classification system with unsupervised learning. First, the adaptive resonance theory map is used to make categories for input data. After that the learning vector quantization decides the category borders. In elementary classification problems, algorithm works better as the problem complexity increases.

1. Introduction

Learning vector quantization (LVQ) is a simple and universal classification algorithm [1][2]. The LVQ relates deeply to self-organizing maps: depending on problem complexity, the LVQ can realize flexible classification function that is impossible linear algorithm such as regression analysis and is suitable for many applications. However, the LVQ is a supervised learning algorithm and cannot work for data set without category information of each teacher signal. If we can add labeling function to the LVQ, the performance and flexibility of the LVQ can be improved further.

This paper considers collaboration of adaptive resonance theory map (ART) and LVQ, where the ART plays to elementary classification with unsupervised learning. As is well known, the ART is flexible unsupervised learning algorithm with many applications [3]-[5]. In our previous works, the ART has been used effectively to classify input space for parallel processing and have contributed to performance of Self-Organizing Maps (SOM) and ant colony optimizers (ACO) [6]-[8].

In order to consider the algorithm performance, we apply the ART-LVQ algorithm to elementary classification problems. The basic numerical results suggest that (1) the ART-alone is sufficient for simple classification problems and (2) classification function of ART-LVQ becomes better as the problem complexity increases. The ART-LVQ is novel and can be developed into efficient unsupervised learning system for classification problems. Also the ART-LVQ may help parallel processing of many algorithms including SOM and ACO.

2. Algorithm

Our classification system consists of an improved version of adaptive resonance theory map (IART) and LVQ. As a set of input data is given, the IART subroutine makes category information and the LVQ subroutine makes borders of the input space.

2.1. IART for labeling

Let the input data consist of \(N\) pieces of 2-dimensional points \((x_i, y_i), i = 1 \sim N\). In the IART, the \(i\)-th category at discrete time \(t\) is characterized by a circle at center \((x_i, y_i)\) with radius \(r_i\):

\[
W_i(t) = (x_i(t), y_i(t), r_i(t)), \quad i = 1 \sim N_c(t)
\]

where \(N_c(t)\) is the number of categories at time \(t\). The IART subroutine is defined in the following 6 steps.

Step 1 (Initialization): Let \(t = 0\), \(N_c(0) = 1\), \(r_i(0) = 0\) and let \((x_i(0), y_i(0))\) be selected randomly from some two-dimensional distribution \(P(x, y)\).

Step 2 (Selection): An input \((X, Y)\) is selected randomly from \(P(X, Y)\). If the input belongs to some category then goto Step 5, otherwise we find the closest category to the input.

\[
W_i(t) = (x_i(t), y_i(t), r_i(t))
\]

\[
T_c = \min_i(T_i(X, Y)), \quad T_i(X, Y) = \sqrt{(X - x_i(t))^2 + (Y - y_i(t))^2 - K \times r_i(t)}
\]

where \(T_i\) is the choice function of the \(i\)-th category. \(-1 \leq K \leq 1\) is the distance parameter that is a key to control the number of categories as suggested in Fig. 1. If \(T_c > \gamma\) then goto Step 3 where \(\gamma \in [0, 1]\) is the vigilance parameter. If \(T_c \leq \gamma\) then goto Step 4.

Step 3 (Birth of a new category): A new category is born at position of the input \((X, Y) \equiv P_i\) as shown in Fig. 2(a) and \(N_c(t) = N_c(t) + 1\). The radius of the new category is zero and the suffix \(N_c(t)\) is assigned to the new category.

\[
W_N(t) = (x_{N_c(t)}, y_{N_c(t)}, 0)
\]

2.2. LVQ subroutine

The LVQ subroutine makes borders in the feature space. The ART-LVQ can operate with radius \(r_i\). If we can add labeling function to the LVQ, the performance and flexibility of the LVQ can be improved further.

\[
W_N(t) = (x_{N_c(t)}, y_{N_c(t)}, 0)
\]
Step 4 (Category enlargement): The selected category \( W_c(t) \) is enlarged such that the input \((X, Y)\) is included in the border as shown in Fig. 2(b) and goto Step 5.

Step 5 (Iteration update): Let \( t = t + 1 \). Goto Step 2 and repeat until the time \( t = N \) where \( N \) is the number of inputs. At time \( t = N \), the \( N_c(t) \equiv N_c \) is declared as the number of categories.

Step 6 (Clean-up): If plural categories are overlapped then the overlapped region is divided by lines through the intersections. Using this division line, we can recognize the data category easily. This step does not exist in our previous work [8].

### 2.2. LVQ for border decision

In the LVQ, a weight vector \( w_i \) corresponds to a category and is updated depending on input data. Let \( w_i(t) = (x_i, y_i) \) be the weight vector at position \((x_i, y_i)\) at discrete time \( t \) and let \( N_c \) be the number of categories: \( i = 1 \sim N_c \). The LVQ is defined by the following 5 steps.

**Step 1 (Initialization):** Let the center \((x_i, y_i)\) of the \( i \)-th category \( W_i(t) \) of ART be changed into the position of the \( i \)-th weight vector \( w_i(t) = (x_i, y_i) \). Let \( t = 0 \).

**Step 2 (Input):** An input \((X, Y) \equiv P_t \) is applied. We select the category \( w_c(t) \) that is the closest to the input:

\[
\begin{align*}
\text{Step 3 (Update of category):} \quad & \text{The selected category } w_c(t) \text{ is updated as the following:}

w_c(t)_{\text{new}} &= \begin{cases} 
w_c(t)^{\text{old}} \pm a_t(P_j - w_c(t)^{\text{old}}), & (w_c(t) = w_i) \\
\{w_c(t)^{\text{old}} - a_t(P_j - w_c(t)^{\text{old}}), & (w_c(t) \neq w_i) 
\end{cases}
\end{align*}
\]

where \( w_i \) is the category of the input given by the IART. \( a_t \in [0, 1] \) is a learning parameter that is linearly monotone decreasing for time \( t \):

\[
a_t = -a_0(t/t_{\text{total}} - 1)
\]

where \( 0 < a_0 < 1 \) is a positive parameter and \( t_{\text{total}} \) is a total number of learning iterations.

**Step 4 (Border decision):** Let \( t = t + 1 \), goto Step 2 and repeat until the maximum time limit \( t_{\text{max}} \). At time \( t = t_{\text{max}} \), the borders of categories are decided by perpendicular bisector of the category position as shown in Fig. 3.

**Step 5 (Category reassignment):** If the LVQ-based category is different from the ART-based category, the LVQ-based category is to be valid.

### 3. Numerical Experiments

In order to consider the classification function, we apply the ART-LVQ algorithm to elementary problems whose the input space is given by union of three Gaussian distributions:

\[
N_i(\mu_i, \sigma^2_i), \quad i = 1 \sim 3
\]

where \( \mu \) and \( \sigma \) is the mean and the standard deviation, respectively. In the experiments, we have used the following two problems based on three Gaussians (\( N_c = 3 \) as shown in Fig. 4(a)):

**P1:**
\[
\begin{align*}
N_1(0.75, 0.45; 0.12^2, 0.12^2) \\
N_2(0.75, 0.45; 0.12^2, 0.12^2) \\
N_3(0.75, 0.45; 0.12^2, 0.12^2)
\end{align*}
\]

**P2:**
\[
\begin{align*}
N_1(0.75, 0.45; 0.12^2, 0.12^2) \\
N_2(0.75, 0.45; 0.12^2, 0.12^2) \\
N_3(0.75, 0.45; 0.12^2, 0.12^2)
\end{align*}
\]
3.1. Categories extraction by IART

In the numerical experiments, the parameters are fixed after trial-and-errors:

\[ \gamma = 0.35, \quad K = 0 \]

Fig. 4(b) shows an example of results for \( P_2 \). 100 inputs are selected randomly from \( N_1 \cup N_2 \cup N_3 \) and are applied to the IART. The IART has succeeded to extract three categories. In order to evaluate such results, we introduce two measures. The first one is the coincidence rate of the ART-based category with the right answer:

\[ CR_1 = \frac{\text{# input data with correct category for ART}}{\text{# input data}} \quad (6) \]

The second one is the extraction rate of the categories:

\[ CR_2 = \frac{\text{# categories extracted by ART}}{\text{# categories}} \quad (7) \]

Table 1 summarizes the results where \( CR_1 \) is given in average, min and max for 37 trials where the 3 categories are extracted. \( CR_2 \) is calculated for \( t_{\text{max}} \) trials. We can see that the case \( N = 100 \) gives the best \( CR_2 \) for both \( P_1 \) and \( P_2 \). It suggests that over-learning disturbs category extraction ability. The case \( N = 500 \) gives the best \( CR_1 \) in average: this \( CR_1 \) is a criterion to consider the ART-LVQ function. It should be noted that the three Gaussian functions overlap to each other and higher \( CR_1 \) is hard even if the classification is optimal.

3.2. Categories extraction by ART-LVQ

The parameters are fixed after trial-and-errors:

\[ \gamma = 0.35, \quad K = 0, \quad a_0 = 0.6, \quad t_{\text{total}} = 10^3 \]

Fig. 5 shows an example of results for \( P_2 \). Here we introduce the third measure that is the coincidence rate of the ART-LVQ-based category with the right answer:

\[ CR_3 = \frac{\text{# input data with correct category for ART-LVQ}}{\text{# input data}} \quad (8) \]

Table 1: Results of IART.

<table>
<thead>
<tr>
<th></th>
<th>( N )</th>
<th>( CR_1_{\text{avg}} )</th>
<th>( CR_1_{\text{min}} )</th>
<th>( CR_1_{\text{max}} )</th>
<th>( CR_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P_1 )</td>
<td>100</td>
<td>58.0</td>
<td>5.0</td>
<td>91.0</td>
<td>76.9</td>
</tr>
<tr>
<td></td>
<td>300</td>
<td>69.7</td>
<td>5.0</td>
<td>91.7</td>
<td>36.3</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>68.4</td>
<td>9.4</td>
<td>90.4</td>
<td>7.4</td>
</tr>
<tr>
<td>( P_2 )</td>
<td>100</td>
<td>55.4</td>
<td>12.0</td>
<td>90.0</td>
<td>79.0</td>
</tr>
<tr>
<td></td>
<td>300</td>
<td>56.6</td>
<td>6.7</td>
<td>90.7</td>
<td>69.0</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>60.8</td>
<td>6.8</td>
<td>90.6</td>
<td>50.2</td>
</tr>
</tbody>
</table>

4. Conclusions

We have presented the ART-LVQ algorithm that can realize unsupervised learning for classification. Performing elementary numerical experiment, the algorithm performance has been considered. We can suggest the following.

1) The IART can play good classification performance for simple problems such as \( P_1 \). However, over-learning may reduce the performance.

2) The ART-LVQ is effective in the case of relatively complex problems such as \( P_2 \). Also, fast classification is possible in that case.

Future problems are many, including the following: finding optimal algorithm parameter values, analysis of the learning process, and application to practical problems.
Figure 6: Time-dependence of CR1 for P1.

Figure 7: Time-dependence of CR3 for P2.

Table 2: Learning result CR3 of ART-LVQ. Repeat until the time $t_{max} = N$.

<table>
<thead>
<tr>
<th></th>
<th>N</th>
<th>$CR3_{avg}$</th>
<th>$CR3_{min}$</th>
<th>$CR3_{max}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>100</td>
<td>57.5</td>
<td>1.0</td>
<td>88.0</td>
</tr>
<tr>
<td></td>
<td>300</td>
<td>71.2</td>
<td>3.3</td>
<td>92.0</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>69.2</td>
<td>5.8</td>
<td>91.8</td>
</tr>
<tr>
<td>P2</td>
<td>100</td>
<td>56.3</td>
<td>1.0</td>
<td>91.0</td>
</tr>
<tr>
<td></td>
<td>300</td>
<td>56.8</td>
<td>5.0</td>
<td>91.3</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>62.8</td>
<td>5.6</td>
<td>91.6</td>
</tr>
</tbody>
</table>

References


Discrete Higher Order Inverse Function Delayed Network

Takahiro Sota\textsuperscript{y}, Yoshihiro Hayakawa\textsuperscript{z}, Shigeo Sato\textsuperscript{y} and Koji Nakajima\textsuperscript{z}

\textsuperscript{y}Research Institute of Electrical Communication, Tohoku University
2-1-1 Katahira, Aoba-ku, Sendai-shi, 980-8577, Japan
\textsuperscript{z}Sendai National College of Technology
4-16-1 AyashiChuoh, Aoba-ku, Sendai-shi, 989-3128, Japan
Email: sota@nakajima.riec.tohoku.ac.jp, hayakawa@sendai-nct.ac.jp, shigeo@riec.tohoku.ac.jp, hello@riec.tohoku.ac.jp

Abstract—The Inverse function Delayed model (ID model) is a neuron model that has dynamics effected by negative resistance. The negative resistance can destabilize local minimum states that are undesirable network responses. Actually, we have demonstrated that the ID network can perfectly remove all local minima with N-Queen problems or 4-Color problems where stationary states are only correct answers. Meanwhile, about the case of Traveling Salesman Problems or Quadratic Assignment Problems (TSP, QAP), we have demonstrated that the ID network can perfectly remove all local minima with N-Queen problems or 4-Color problems because the ID network requires much computation time to update neuron states. Consequently, we aim to implement the HC-ID network on hardware to avoid this problem.

In this paper, we introduce the discrete HC-ID network instead of the continuous HC-ID network to achieve the hardware implementation. The discrete network is superior for hardware implementation. We propose the discrete time network, at first, and then derive the binary output network from the discrete time network. Finally, we confirm the expected result by solving a TSP and a QAP numerically.

1. Introduction

Hopfield et al. proposed a neural network that had dynamics of moving along the gradient of the quadratic-form energy function [1]. This network can find the solutions of combinatorial optimization problems (COP) by assigning the optimal solution to the global minima of the energy function. However, the network state is often trapped into local minima except for the global minima, and it is known as the local minimum problem.

As an improvement way of this problem, the method of using the Inverse function Delayed model (ID model) [2] has been proposed. One of important properties of ID model is a negative resistance. The region of the negative resistance is controllable, then the effect of the negative resistance can destabilize the undesirable local minimum states of the energy function with appropriate setting. Especially, N-Queen problems or 4-Color problems are able to be solved with 100% success rate by using the ID model[3].

Meanwhile, in the case of Traveling Salesman Problems (TSP) or Quadratic Assignment Problems (QAP), we have proposed the Higher-order Connection ID model (HC-ID model) and the quartic-form energy function for these problems[4]. The HC-ID model is the ID model with the higher-order synaptic connections, and the HC-ID network with 3rd-order synaptic connections has a quartic-form energy function. This network can destabilize only the undesirable local minimum states of TSP or QAP by introducing the quartic-form energy function. Actually, optimal solutions of some 4-city TSP’s and 4-size QAP’s are obtained in numerical experiments. However, it is difficult to apply the HC-ID network with computer simulation to large size problems because the HC-ID network requires much computation time to update neuron states. Consequently, we aim to implement the HC-ID network on hardware to avoid this problem.

In this paper, we introduce the discrete HC-ID network instead of the continuous HC-ID network to achieve the hardware implementation. The discrete network is superior for hardware implementation. We propose the discrete time network, at first, and then derive the binary output network from the discrete time network. Finally, we confirm the expected result by solving a TSP and a QAP numerically.

2. Discrete HC-ID Network

2.1. Continuous HC-ID Network

The continuous type ID model with 3rd order synaptic connections is described by following equations[4]:

\[
\tau_u \frac{du_i}{dt} = \sum_{j=1}^{N} \sum_{k=1}^{N} \sum_{l=1}^{N} w_{ijkl} x_j x_k x_l + \sum_{j=1}^{N} w_{ij} x_j + h_i, \quad (1)
\]

\[
\tau_s \frac{dx_i}{dt} = u_i - g(x_i), \quad (2)
\]

where \(N\) is the number of neurons, and \(u_i\), \(x_i\) and \(h_i\) are the internal state, the output and the bias of neuron \(i\), respectively. \(w_{ijk}\) is the synaptic weight from neurons \(j,k,\ldots\) to neuron \(i\). \(\tau_u\) and \(\tau_s\) are the time constants of the internal state and the output, respectively.

Substituting Eq. (2) into Eq. (1), the following equation is obtained:

\[
\tau_s \frac{d^2x_i}{dt^2} + \eta(x_i) \frac{dx_i}{dt} = - \frac{\partial U}{\partial x_i}, \quad (3)
\]
where
\[ \eta(x_i) = \frac{dg(x)}{dx} \bigg|_{x=x_i}, \]  
\[ U = -\frac{1}{4\tau_u} \sum_{i} \sum_{j} \sum_{k} \sum_{l} w_{ijkl} x_i x_j x_k x_l \]  
\[ -\frac{1}{3\tau_u} \sum_{i} \sum_{j} \sum_{k} w_{ijk} x_i x_j x_k \]  
\[ -\frac{1}{2\tau_u} \sum_{j} w_{ij} x_i x_j - \frac{1}{\tau_u} \sum_{i} h_i. \]  
(5)

\[ \eta(x_i) \] and \( U \) denoting a friction coefficient and a potential, respectively, Equation (3) shows that the HC-ID model operates as the particle in potential \( U \). Moreover the negative resistance appears in the area where the gradient of the function \( g \) is negative. These dynamics of HC-ID model are similar to the normal ID model, except the potential \( U \) of 3rd HC-ID model is a quartic-form function and there is no effect of dissipation.

2.2. Derivation of Discrete Time Model

The discrete time HC-ID model can be derived by applying Euler method to Eqs. (1) and (2). The basic equations of the discrete time HC-ID model are

\[ u_i(t+1) = u_i(t) + T_\Delta \sum_{j=1}^{N} \sum_{k=1}^{N} \sum_{l=1}^{N} w_{ijkl} x_j(t)x_j(t) + \sum_{j=1}^{N} w_{ij} x_j(t) + h_i \]  
\[ x_i(t+1) = x_i(t) + T_\Delta \frac{T_u}{\tau_u} \left( u_i(t) - g(x_i(t)) \right). \]  
(7)

where
\[ T_\Delta = \Delta/\tau_u \leq 1, \]  
(8)
and \( \Delta \) is a time increment.

2.3. Limitation of Output and Inner State

Although the output range of HC-ID model has no limitation normally, the range should be between 0 and 1 to apply to COP. Hence the discrete time HC-ID model restricts the range of the output as follows when calculating the next step output:

\[ x(t+1) = \begin{cases} 0 & \text{if } x(t+1) < 0 \\ 1 & \text{if } x(t+1) > 1 \end{cases}. \]  
(9)

Moreover, the inner state \( u \) is also limited as follows due to the output restriction:

\[ u(t+1) = u(t) \begin{cases} 1 & \text{if } u(t+1) > u(t) < 0 \\ 0 & \text{if } u(t+1) < u(t) \end{cases}. \]  
(10)

2.4. Derivation of Binary Output Model

The output of discrete time HC-ID model becomes binary values when the parameter \( \tau_u \) in Eq. (7) approximates to 0. The output is obtained as follows under the limitation of \( \tau_u \to 0 \):

\[ x_i(t+1) = \begin{cases} 1 & (u_i(t) - g(x_i(t)) > 0) \\ 0 & (u_i(t) - g(x_i(t)) < 0) \end{cases} \]  
\[ x_i(t) \] (11)

Moreover, the \( g \)-function for binary output is defined by following equation:

\[ g(x) \begin{cases} \leq \alpha & (x = 0) \\ \geq -\alpha & (x = 1) \end{cases}. \]  
(12)

where \( \alpha \) is a positive parameter. This \( g \)-function no longer has negative resistance effect, it has only the hysteresis effect. That effect increases with increasing \( \alpha \). And the output of the binary model is described as follows from Eqs. (11) and (12):

\[ x_i(t+1) = \begin{cases} 1 & (u_i(t) > \alpha) \\ 0 & (u_i(t) < -\alpha) \end{cases}. \]  
(13)

We use this discrete HC-ID model afterward.

2.5. States Stability of Discrete HC-ID Network

Eq. (6) is able to rewrite as Eq. (14) by using the potential \( U \):

\[ u_i(t+1) = u_i(t) - T_\Delta \frac{\partial U}{\partial x_i} \bigg|_{x_i=x_i(t)}. \]  
(14)

Therefore, the inner state \( u \) keeps decreasing if \( \partial U/\partial x_i \) is positive, and finally the output \( x_i \) becomes 0. In contrast, the output becomes 1 eventually if \( \partial U/\partial x_i \) is negative.
3. Applying Quartic Energy Function

3.1. Quartic Energy Function

The quartic energy function for combinatorial optimization problems is [4]

$$E_{4TH} = \frac{A}{2} \sum_{i=1}^{n} \left( \sum_{j=1}^{n} x_{ij} - 1 \right)^2 + \frac{A}{2} \sum_{j=1}^{n} \left( \sum_{i=1}^{n} x_{ij} - 1 \right)^2$$

$$+ \frac{B}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \sum_{l=1}^{n} b_{i,j,k,l} x_{i,k} x_{j,l} (1 - x_{i,j} x_{k,l})$$

$$+ \frac{C}{2} \left( \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \sum_{l=1}^{n} b_{i,j,k,l} x_{i,k} x_{j,l} \right)^2,$$  \hspace{1cm} (15)

where $n$ is the size of a problem and $A$, $B$ and $C$ are coefficients that have positive value. $b_{i,j,k,l} \in b^{(N \times N)}$ is a cost value when the neuron $(i,j)$ and $(k,l)$ fire, and the cost matrix $b$ has to be modified by the applied problem. The first and second terms of Eq. (15) have minimum value if only one neuron fires in each row and column, and the third term of Eq. (15) is minimized when all of output values are 0 or 1. These three terms express constrained conditions, and they will be 0 when the conditions are satisfied. Under these conditions, the fourth term expresses a squared value of cost shown by the network state.

Moreover, the partial differentiation of $E_{4TH}$ with respect to a output $x_{ab}$ is expressed as follows if the constrained conditions are satisfied:

$$\frac{\partial E_{4TH}}{\partial x_{ab}} = \sum_{y} \sum_{k} p_{ahk} x_{y|k} x_{y|k}$$

$$\left\{ 2 \left( \frac{1}{2} - x_{ab} \right) + 2C \sum_{i} \sum_{k} \sum_{w} \sum_{l} b_{i,k,w} x_{i,k} x_{l,w} \right\}$$

$$= \left\{ \sum_{i} b_{ih,iF[y]} \cdot 4C \left( - \frac{B}{4C} + c_{sol}(x) \right) \right\} (x_{ab} = 0)$$

$$+ \left\{ \sum_{i} b_{ih,iF[y]} \cdot 4C \left( - \frac{B}{4C} + c_{sol}(x) \right) \right\} (x_{ab} = 1),$$  \hspace{1cm} (16)

where $F[y]$ is the column index of a firing neuron in row $y$ and

$$c_{sol}(x) = \frac{1}{2} \sum_{w} \sum_{z} b_{iz,F[y]\cdot x_{z,w}},$$  \hspace{1cm} (17)

that is, $c_{sol}(x)$ expresses the cost value obtained by using the solution represented by network state $x$. Equation (16) shows that the sign of $\partial E_{4TH}/\partial x_{ab}$ depends on the relationship between the coefficient $B/4C$ and the cost of solution $c_{sol}(x)$ when $x_{ab} = 1$. That is to say, $E_{4TH}/\partial x_{ab}$ has negative value if $B/4C < c_{sol}(x)$ holds, and $E_{4TH}/\partial x_{ab}$ is positive if $B/4C > c_{sol}(x)$ holds.

3.2. Relationship between Network States Stability and Coefficients of Quartic Energy Function

When the energy function is applied to the HC-ID network, the weight matrix and bias value of the neural network are constructed to make the potential $U$ equal to the energy function. Hence a condition $\partial U/\partial x_i = \partial E_{4TH}/\partial x_i$ is satisfied if the quartic energy function is applied to the HC-ID network. It means that $x_{ab} = 1$ is stable if $\partial E_{4TH}/\partial x_{ab}$ is negative, and $x_{ab} = 1$ is unstable if $\partial E_{4TH}/\partial x_{ab}$ is positive. Hence only the state whose cost is smaller than the value of $B/4C$ is stable, according to the discussion of the preceding section. Consequently we can obtain solutions whose cost is smaller than $B/4C$. The solutions contain both of the global and local minimum states. The local minimum state solutions decrease with decreasing $B/4C$. Finally, only the global minimum states are stabilized and all local minimum states are destabilized by using the discrete HC-ID network when following condition is satisfied:

$$c_{sol}(x_0) < B/4C < c_{sol}(x_1),$$  \hspace{1cm} (18)

where $c_{sol}(x_0)$ and $c_{sol}(x_1)$ are the cost value of optimal solution and the 2nd optimal solution, respectively. This condition being the same as the one of continuous[4], therefore the discrete HC-ID network can solve COP by the same way as the continuous network.

4. Simulation Result

This section shows the simulation results of 6-city TSP and 4-size QAP. In these simulation random initial states are used, and the result is obtained by 48 simulations. The coefficient $A$ of the energy function is set to satisfy the constrained condition, and $B/4C$ is set to satisfy Eq. (18).

4.1. Success Rate and Computation Time Dependency on Parameter $T_A$

At first, it is investigated how the computation time depends on the parameter $T_A$ by numerical experiments of a 6-city TSP. The success rate is always 100% for any value of $T_A$, and the variation of computation time is shown in Fig. 2. The solid circles in Fig. 2 show the average of computation time. This result suggests that $T_A$ has no impact to
the success rate, and the computation time decreases with increasing the parameter $\alpha$.

4.2. Success Rate and Computation Time Dependency on Parameter $\alpha$

Next, computation time dependency on the parameter $\alpha$ is investigated. At this time $T_\Delta$ is set to 1 from the foregoing result. In all $\alpha$ range, the optimal solution is obtained with 100% success rate. The computation time is shown as a function of $\alpha$ in Fig. 3. The computation time takes a minimum value at $\alpha \approx 400$, it is important to set an appropriate parameter $\alpha$ for solving COP with smaller iterations.

4.3. The Result of 4-QAP

Finally, we compare the success rate of the discrete network with that of the continuous network by solving a 4-size QAP. Figure 4 shows the success rate dependency on the parameter $\alpha$ in the 4-QAP. It shows that the discrete HC-ID network can solve the 4-QAP with 100% success rate at any $\alpha$ value.

However, in case of using continuous HC-ID network, it is not always that the network state reaches to global minimum states of the QAP [4]. Figure 5 shows the output of continuous HC-ID network applied for a 4-QAP. It shows the local minimum states of a 4-QAP. If the effects of negative resistance is small, the continuous HC-ID network often cannot reach the global minimum states because the output of firing neuron oscillate around $x = 1$ as shown in Fig. 5. The output of discrete HC-ID network, meanwhile, does not take intermediate value between 0 and 1, consequently the discrete HC-ID network can avoid the oscillation state with any $\alpha$.

5. Conclusion

We introduced the discrete HC-ID model instead of the continuous HC-ID model to achieve the hardware implementation. The introduced model operate in discrete time, and the output of neurons are binary. Moreover, we showed analytically that the only optimal solution states are obtained by using the discrete HC-ID network with the energy function similar to the continuous HC-ID network. Finally, we applied the discrete HC-ID network to solving a 6-TSP and a 4-QAP in numerical experiments.

References


Analyses of Coupled Hindmarsh-Rose Type Bursting Oscillators

Koji Kurose†, Takahiro Sota‡, Yoshihiro Hayakawa‡, Shigeo Sato† and Koji Nakajima†

†Laboratory for Brainware/Laboratory for Nanoelectronics and spintronics,
Research Institute of Electrical Communication, Tohoku University
2-1-1 aoba-ku kahara,sendai,980-8577 Japan
‡Sendai National College of Technology
Email: kurose@nakajima.rie.tohoku.ac.jp

Abstract— We proposed the Inverse function Delayed model(ID model) as one of neuron models [1]. In addition, we have propose Burst firing ID model [2] of Hodgkin-Huxley(H-H) type model that has the burst oscillating characteristics with three variables, because we consider that the burst dynamics has prospects of capabilities of effective tool for information processing consequently. The burst ID model is explained with a new concept that the neuron dynamics is expressed as a motion of a quasi particle in a potential with active areas, and then we are able to apply the concept to others. Through the technique, we are able to foresee the landscape of solutions with the curvature of the potential and design the wave forms if we place the active area on the potential properly [3]. In this paper, we apply this concept to coupling systems and the concept is effective in the interconnected systems too, and analyze the dynamics of the coupled models with the burst firing characteristics.

1. Introduction

We proposed the Inverse function Delayed model(ID model) as one of neuron models [1]. In addition, we have proposed Burst firing ID model of Hodgkin-Huxley(H-H) type model that has the burst oscillation characteristics with three variables, because we consider that the burst dynamics has prospects of capabilities of effective tool for information processing consequently. The burst ID model is explained with a new concept that the neuron dynamics is expressed as a motion of a quasi particle in a potential with active areas, and then we are able to apply the concept to others. We are able to obtain the landscape of the solution with the curvature of the potential. If the curvature is positive as a whole, the wave forms of the solutions are oscillations or resting state without divergence. On the contrast, when the curvature is negative, we can observe the divergence solution.

This paper is organized as follows. We explain our concept of potential with the active area in the next section. We analyze the dynamics with van der Pol model as one of the simplest nonlinear oscillator, and our concept is effective for analyses of coupled system in Sec.3. Last section, we use Hindmarsh-Rose type model for the analyses of the dynamics of coupled bursting oscillators.

2. Potential with Active Area

Neuron models are typically expressed in the form of multidimensional nonlinear differential equation. Several models can be transformed into higher differential equation with one variable.

\[
d\frac{dx}{dt} + b_1(x) \frac{dx}{dt} + \ldots + b_m(x) \frac{dx}{dt} + \frac{dx}{dt} = F(x, \theta) = -\frac{\partial U(x, \theta)}{\partial x} \tag{1}
\]

\(U(x)\) is a kind of the potential function, and has some equilibria \(x_0\) that depend on the external input \(\theta\). We have obtained the characteristic equation and analyze the stability at the neighborhood of equilibria determined in accordance with Hurwitz’s theorem, and the condition of \(x_0\) are stable(n=4), if

\[
b_0(x_0) = \frac{d^2U(x_0)}{dx^2} > 0, \tag{2}
\]

\[b_i(x_0) > 0, (i = 1, 2, 3) \tag{3}\]

\[B_1(x_0) = b_2(x_0, 0, 0)b_1(x_0, 0) - b_0(x_0)b_3(x_0, 0, 0, 0), b_0(x_0) > 0 \tag{4}\]

\[B_2(x_0) = B_1(x_0)b_1(x_0, 0) - b_3(x_0, 0, 0)b_0(x_0) > 0. \tag{5}\]

Equation (2) means that the curvature \(b_0\) of the potential \(U(x, \theta)\) is positive at \(x_0\). An equilibrium points is unstable where \(b_1(x) < 0\) or \(b_2(x) < 0\) or \(b_3(x) < 0\) or \(B_1(x) < 0\) or \(B_2(x) < 0\) is satisfied even if the global curvature of the potential is positive. In other words, if there is no equilibrium point satisfy the requirement of Hurwitz’s theorem and the global curvature is positive, it causes various oscillations, for example, periodic limit cycle, burst firing and so on. We identify the section, where coefficients and these minors are negative, as active area. The section of \(x\) where \(b_1(x) < 0\) is satisfied represents a negative damping area, we call this area \(b_1\) active area.

3. van der Pol Model

3.1. Basic Equation

we explain our concept of potential and active areas with stand-alone van der Pol model. Generally this model is
expressed as following equation

\[
\frac{d^2 x}{dt^2} - q(1-x^2) \frac{dx}{dt} = -x. \tag{6}
\]

In this paper, we consider the following model equation of van der Pol type, because we can set the active area arbitrarily considering the external input and time-delay.

\[
\tau \frac{d^2 x}{dt^2} + \varepsilon [(x-\alpha)^2 - \beta] \frac{dx}{dt} = W x + Z = -\frac{\partial U(x, \theta)}{\partial x}. \tag{7}
\]

We express in the form of differential equation with two variable,

\[
\tau \frac{dx}{dt} = u - \varepsilon \left[ \frac{1}{3} x^3 - \alpha x^2 + (\alpha^2 - \beta)x \right]
\]
\[
\frac{du}{dt} = W x + \theta. \tag{8}
\]

Where \(x, u, W, \theta, \) and \(\tau\) are the output of the unit, the internal state, the self-connection, the external input, the time constant respectively, \(\alpha, \beta, \) and \(\varepsilon\) are control parameters. \(U(x, \theta)\) is a kind of the potential function, described as

\[
U(x) = -\frac{1}{\tau} \left( \frac{1}{2} W x^2 + \theta x \right). \tag{9}
\]

Potential function has one equilibrium point \(x_0\) and the curvature of potential \(b_0 = \frac{d^2 U(x_0)}{dx^2} = -W\). Therefore the potential function became a convex function if \(W > 0\) and we obtain a divergence solution. In contrast, the potential function is concave function if \(W < 0\). The equilibrium point is located internally in active area, the output is continuous oscillation in this case. This system expressed by Eq.(7) has one active area \(b_1\) active area.

\[
b_1(x) = \frac{\varepsilon}{\tau} [(x-\alpha)^2 - \beta]. \tag{10}
\]

The \(b_1\) active area is the section of \(x\) that where \((x-\alpha)^2 - \beta < 0\) satisfied, where \(\beta > 0, \alpha, \) and \(\beta\) denote parameters that define the middle of \(b_1\) active area and its width.

### 3.2. Interconnected Models

We discuss the interconnected system through technique noted above, and we interconnect two units described by Eq.(7), however the interconnection and self-connection are equal \((\omega_{ij} = \omega, W_i = W_j = W)\),

\[
\tau_x \frac{d^2 x}{dt^2} + \varepsilon_x [(x - \alpha)^2 - \beta_x] \frac{dx}{dt} = W x + \theta x, \tag{11}
\]
\[
\tau_y \frac{d^2 y}{dt^2} + \varepsilon_y [(y - \alpha)^2 - \beta_y] \frac{dy}{dt} = W y + \omega x + \theta. \tag{12}
\]

Equation (11) and (12) can be transformed into the differential equation with one variable \(x\)

\[
\frac{d^4 x}{dt^4} + b_3(x) \frac{d^3 x}{dt^3} + b_2(x, \dot{x}) \frac{d^2 x}{dt^2} + b_1(x, \dot{x}) \frac{dx}{dt} + \frac{\partial U(x)}{\partial x} = 0. \tag{13}
\]

where

\[
b_1(x) = -\frac{W}{\tau_x} \frac{\eta(x)}{\tau_x} - \frac{W}{\tau_x} \frac{\zeta(x)}{\tau_x}, \tag{14}
\]
\[
b_2(x) = -\frac{W}{\tau_x} \frac{\eta(x)}{\tau_x} + \frac{W}{\tau_x} \frac{\zeta(x)}{\tau_x}, \tag{15}
\]
\[
b_3(x) = \eta(x) + \zeta(x), \tag{16}
\]
\[
B_1(x) = b_1(x) \eta(x) - b_3(x) \theta_0(x), \tag{17}
\]
\[
B_2(x) = b_1(x) \theta_0(x) - b_3(x) \theta_0(x), \tag{18}
\]

where \(\eta(x) = \frac{x}{\tau_x}((x-\alpha)^2 - \beta_x), \zeta(x) = \frac{x}{\tau_x}((W+\eta(x) + \alpha_x)^2 - \beta_x).\)

We can also obtain the potential function

\[
U(x) = \frac{1}{\tau_x} \left( \frac{1}{2} (W^2 - \omega^2) x^2 + (W\theta_x - \omega\theta) x \right). \tag{19}
\]

This potential has one equilibrium point as stand-alone model, and the equilibrium point \(x_0\) is

\[
x_0 = -\frac{W\theta_x - \omega\theta}{W^2 - \omega^2}, \tag{20}
\]

and it also depends on the external inputs and the connection strength. The curvature of the potential \(b_0\) is

\[
b_0(x) = \frac{1}{\tau_x \tau_y} \left( \frac{1}{2} (W^2 - \omega^2) \right), \tag{21}
\]

and hence we obtain a divergence solution if \( | W | < | \omega | \).

When the equilibrium point is located internally in some of the five active areas described by Eq.(14)~Eq.(18) with \( | W | > | \omega | \), we can obtain continuous oscillations.

### 3.3. Fast-Slow Dynamics

We simulate the coupled system where \(\beta_x = 1.0, \epsilon_x = 3.0, \epsilon_y = 1.0, \theta_x = 0.0, \tau_x = 1.0, \tau_y = 100, \) and observe the output \(x\) changing the parameters \(\beta_x, \theta_x\) and \(\tau_x\). Figure.1 shows the output and the position of active areas for the equilibrium point expressed by Eq.(20) of the quadratic potential.

The line L1 and L2 denote parameter sets when the equilibrium point of the potential is on the edge of active areas. It is within \(b_1\) active area all over the region in Fig.1. It is within \(b_1\) active area and \(b_2\) active area above L1 and L2. We are able to observe various oscillations near by L2. For example, spike pulse(SP), oscillation with two different frequencies (SFO) and bursting oscillation(FB). If we change the parameter \(\theta_x\) under an identical condition about parameter \(\beta_x\). In region (a), we observe the slow oscillation as a solution, we increase the external input \(\theta_x\) and cross the line L, then we are able to get the spiking pulse in very narrow region of diagram. When the more the equilibrium point of the potential approaches \(b_1\) active area with increasing the external input, the number of spikes per burst increases generated in a time between resting states. Figure (2) shows various wave forms. In other words, if the potential is single well form and there are active areas with two different frequency, we are able to obtain the spiking or bursting oscillation to set the active area with low frequency on the equilibrium point and the active area of fast oscillation near the equilibrium point.
4. Coupled Bursting Oscillators

4.1. Basic Equations for Hindmarsh-Rose Type Model

We are interested in coupled bursting oscillators, considering that our concept is very effective for the analyses of these systems. Therefore, we choose Hindmarsh-Rose Type[4] model as a bursting oscillator. We calculate the following equations, because we are able to set the active area optionally, interconnection between the units and time constant of units.

\[
\frac{dx}{dt} = u - z - g(x) + I, \\
\frac{du}{dt} = -u - W(x) + \theta, \\
\frac{dz}{dt} = r[Z(x) - z],
\]

\[g(x) = \epsilon\left(\frac{1}{3}x^3 - ax^2 + (a^2 - \beta)x\right)\] (25)

\[W(x) = dx^2\] (26)

\[z(x) = s(x - z_0)\] (27)

where \(x\) denotes the output, \(a, \beta, \epsilon, d, r, s\) and \(z_0\) denotes the control parameters, \(I\) denotes an external stimulus, \(\theta\) is the bias and \(\tau\) is the time constant. \(g(x), W(x)\) and \(Z(x)\) are defined according to the second section, we transform Eq.(22)–(24) into the following single equation.

\[
\frac{d^3x}{dt^3} + b_2(x)\frac{d^2x}{dt^2} + b_1(x, s)\frac{dx}{dt} \\
= -\frac{\epsilon}{\tau}(g(x) + z(x) + W(x) - I - \theta),
\]

where \(b_2(x)\) and \(b_1(x)\) functions are expressed as

\[b_2(x) = \frac{1}{\tau}\epsilon\left((x - \alpha)^2 - \beta\right) + r + 1,\] (29)

\[b_1(x) = \frac{1}{\tau}\left[(r + 1)\epsilon\left((x - \alpha)^2 - \beta\right) + 2dx + r(s + \tau)\right].\] (30)

This differential equation generate the spiking burst for \(0.82 < I_c < 3.25\).[5] At this time, the potential function forms single well, and a equilibrium point is covered by \(b_1\) active area that controls the slow oscillation. \(b_2\) active area is placed at beside equilibrium point. That is, the form of the potential and the position of active areas on the potential(Fig.3) are similar to coupled van der Pol system. Increasing the time constant \(\tau\), the dynamics of this model change dramatically. The output \(x\) is shown in Fig.4(b). One of the features of this output, it has rapid oscillations at the top of the saddle of the output. This type of bursting oscillation called tapered bursting[6] is observed with Burst ID model([3] Fig.6(b)). We make a connection be-
Figure 3: Potential and active area of Hindmarsh rose model where $I = 1.4$, when we are able to observe a regularly bursting oscillation.

Figure 4: The time series of output $x(t)$ and $y(t)$ for Hindmarsh-Rose type model without any connections, where $\alpha = 1, \beta = 1, d = 5, \theta = 1, s = 4, Z_0 = -1.6, r = 0.001, (a) I = 1.4, \tau = 1$, (b) $I = 3.0, \tau_y = 2$.

between this models in the form of the follows,

$$\frac{d^3x}{dt^3} + b_2(x) \frac{d^2x}{dt^2} + b_1(x, s) \frac{dx}{dt} = -\frac{r}{\tau_s} (g(x) + z(x) + W(x) - I_x - \theta_s) + \omega y,$$

(31)

$$\frac{d^3y}{dt^3} + b_2(y) \frac{d^2y}{dt^2} + b_1(y, y) \frac{dy}{dt} = -\frac{r}{\tau_y} (g(y) + z(y) + W(y) - I_y - \theta_y) + \omega x.$$

(32)

Figure 5 shows time series of the output $x(t)$ and $y(t)$, where $\tau_x = 1.0, \tau_y = 2, I_x = 1.4, I_y = 3, \epsilon = 3.0, \alpha = 1.0, \beta = 1.0, d = 5, s = 4, Z_0 = -1.6, \theta = 1, \omega = 0.0005, r_x = 0.001, r_y = 0.001$. Indexes $x$ and $y$ denote each units. This model generate the periodic spiking pulse in stand-alone, if $I_x = 1.4$. We are able to observe regular bursting spikes, the output $x$ and $y$ show two type burst firing oscillate in anti-phase. As we increase the connection strength $\omega$, the period of the bursting oscillations becomes longer, being infinity eventually. That is, the outputs $x$ and $y$ become a resting state. This dynamics is caused by a oscillation of the potential expressed in Fig.3. The potential is forced to oscillate by the bursting output $y(t)$.

5. Conclusion

We apply our concept of the potential with active areas to interconnected system. The bursting dynamics with undershoot in single well is revealed. This dynamics has relation to the position of active area that controls fast oscillation for the equilibrium point. We figure out the dynamics of the coupled system between bursting oscillators through our concept.

References


Spectral analysis of the propagating pulse wave in 6 coupled bistable oscillators

Kuniyasu Shimizu†, Motomasa Komuro‡ and Tetsuro Endo¶

†Dept. of Electrical, Electronics and Computer Engineering, Chiba Institute of Technology
2-17-1, Tsudanuma, Narashino, Chiba, 275-0016 Japan
‡Dept. of Occupational Therapy, Teikyo University of Science and Technology, Japan
¶Dept. of Electronics and Bioinformatics, Meiji University
1-1-1, Higashi-mita, Tama-ku, Kawasaki, 214-8571 Japan
Email: kuniyasu.shimizu@it-chiba.ac.jp

Abstract—Frequency components of the propagating pulse wave observed in 6 coupled bistable oscillators are investigated. It is demonstrated that the propagating pulse wave shows a multimode oscillation. These frequency components are clearly changed, as the degree of nonlinearity becomes weak. For weak nonlinear case, the spectrum of the solution shows four dominant peaks, which attracts controversy based on the past theoretical works.

1. Introduction

It is frequently observed in various fields that coupled multiple elements which have autonomous dynamics become to have a valuable function as a unit. Propagating wave phenomena, which may be utilized modeling of biological information processing[1], correspond to one of them. Various propagating wave phenomena in several systems such as chaotic pulse, propagation of phase states, etc. have been investigated and have attracted constant interest in various areas[2, 3].

In our previous work, we demonstrated that there exists the propagating pulse wave in an inductor-coupled bistable oscillator array even in a practical setting[4]. The propagating pulse wave consists of several adjacent oscillators oscillating with large amplitude, and the part of large amplitude oscillation in the array propagates with a constant speed.

In the meanwhile, to provide a theoretical framework for understanding this system, the analysis of oscillation modes based on the averaging method or the perturbation method for weakly nonlinear case were extensively performed in [5, 6]. One of the notable results is an out-of-phase synchronization of envelopes (the synchronized multimode waves)[6]. This solution can be considered as a kind of propagating wave phenomena, and seems to be similar to the propagating pulse wave in our case. In order to make understanding of the propagating pulse wave forward, the differences between the results in our case and the similar solution should be appreciated.

The propagating pulse wave shows an almost periodic oscillation. In order to demonstrate the property, it is necessary to investigate the dominant oscillation frequencies.

In addition, because the dominant frequency of the solution in [6] agrees well with the theoretical value, we can draw a comparison between two solutions for weak nonlinearity.

In this paper, we investigate spectral property of the propagating pulse wave observed in 6 coupled bistable oscillators. Firstly, we show that how frequency components of the propagating pulse wave are included for strong nonlinear case. Then, we pay attention to how its frequency components are changed as the degree of nonlinearity becomes weak. In addition, we compare the results with the similar solution, and discuss the differences between them.

2. Fundamental equation

The system equation of a ring of inductor-coupled bistable oscillators can be written in the following with reference to [4]:

\[
\begin{align*}
\dot{x}_1 &= y_1, \\
\dot{y}_1 &= -\epsilon(1-\beta C_1 + x_1^2)y_1 - K_N x_1 \\
&\quad \text{(}t = d/dt),
\end{align*}
\]

where \( x = [x_1, x_2, \ldots, x_N]^T \), \( y = [y_1, y_2, \ldots, y_N]^T \), \( x_C = [x_1^2, x_2^2, \ldots, x_N^2]^T \), and \( x_f = [x_1^4, x_2^4, \ldots, x_N^4]^T \). The matrix \( K_N \) is defined as

\[
K_N = \begin{bmatrix}
1 + \alpha & -\alpha & \cdots & -\alpha \\
-\alpha & 1 + \alpha & \cdots & 0 \\
\vdots & \ddots & \ddots & \vdots \\
0 & \cdots & -\alpha & 1 + \alpha 
\end{bmatrix}.
\]

\( N \) is the number of oscillators. The \( x_0, i = 1, 2, \ldots, N \) denotes the normalized output voltage of the \( i \)-th oscillator, \( y \) denotes its derivative. The parameter \( \epsilon (> 0) \) shows the degree of nonlinearity. The parameter \( \alpha (0 \leq \alpha \leq 1) \) is a coupling factor; namely, \( \alpha = 1 \) means maximum coupling, and \( \alpha = 0 \) means no coupling. The parameter \( \beta \) controls amplitude of oscillation. Each isolated oscillator has two steady-states, namely, no oscillation and periodic oscillation depending on the initial condition. Hereafter, we investigate spectral property for \( N = 6 \) case, and the parameter \( \beta \) is fixed to 3.2 for simplicity.
3. Propagating pulse wave and its frequency components

There exists the propagating pulse wave in the system (1). In the following, we review the solution for strong nonlinear case, and then we investigate its spectral property as a first step. Next, frequency component variations with respect to the degree of nonlinearity is investigated. Then, we will discuss the results based on the perturbation method, and compare with the similar solution for weak nonlinear case.

3.1. Strong nonlinear case

A typical propagating pulse wave for $\varepsilon = 0.36$ and $\alpha = 0.1$ is depicted in Fig.1. It is clear that the part of large amplitude oscillation in the array propagates with a constant speed. It should be noted that the solution shows an almost periodic oscillation. The corresponding envelope appears in the form of almost periodic oscillation as a corollary.

To detect the dominant component frequencies of $x_i$, $i = 1, 2, \cdots, 6$, Fast-Fourier Transform (FFT) is applied. The power spectrum of $x_1$ associated with the propagating pulse wave in Fig.1 is illustrated in Fig.2. The solution has several dominant peaks, namely this is a multimode oscillation. In the following section, variations of these frequency components with decreasing $\varepsilon$ are investigated.

3.2. Variations with the degree of nonlinearity

The propagating pulse wave appears via some kind of global bifurcation[7]. The bifurcation occurs at the point around the pitch-fork (PF) bifurcation point ($\alpha_{PF}$) of a certain kind of periodic solution. Figure 3 shows a PF bifurcation set of the periodic solution in the $\varepsilon - \alpha$ plane. In neighborhood of the right side of the bifurcation set, the propagating pulse wave appears. That is, at least for $0.01 \leq \varepsilon \leq 0.36$, there exists the propagating pulse wave near the bifurcation set, which can be seen by direct computer simulation of Eq.(1).

It is depicted in Fig.4 that the frequency components of the propagating pulse wave are changed with the value of $\varepsilon$. The dominant frequency components are indicated by the sharp line. As an example, Fig.2 shows nearly comparable result for $\varepsilon = 0.36$. It should be noted that all dominant frequency components approach to $\omega = 1.0 \text{ rad/sec}$, as $\varepsilon$ becomes small.

---

1. All numerical integrations are carried out by 4th order Runge-Kutta method with a step size of 0.01.
2. Power spectra in this paper are obtained from $2^{13}$ sampling points with $48\text{KHz}$ resolution.
3. To be exact, there is a slight difference with the value of $\alpha$. $\alpha = 0.1$ in Fig.2, and $\alpha = 0.101577$ in Fig.4 for $\varepsilon = 0.36$. The reason is found in the caption in Fig.4.
3.3. Weak nonlinear case

According to [6], the multimode (M modes, M=1, 2, · · ·) waves can occur for weak nonlinear case. These stabilities entirely depend on the values of M and β (−g3 in the literature). For β = 3.2, the stabilities of M modes oscillations are calculated in Table 1 referring to the literature, where \( p_{\omega}^{i,m} \), \( i, m = 1, \cdots, M \) correspond to the approximates of stationary amplitudes of equilibrium point in the perturbation method, and the corresponding eigenvalues decide the stability of each equilibrium point. It should be noted that it is indicated that the solutions for M= 1, 2 (with the bold strokes in Table 1) exists stably. On the other hand, there is no solution which is stable in this case for 3 \( \leq M \leq 100 \) (we only show the results up to M = 5 because of space limitations).

In [6], the quasi-periodic oscillations are investigated, some of which (especially for M = 2) have very interesting synchronous property on envelope of each time waveform. Then, we investigate what kind of multimode oscillation (almost periodic oscillation) is resulted when \( \varepsilon \) becomes small enough so that the perturbation method is effective.

Figure 5 shows an upper envelopes (env \( (x_i) > 0 \)) of the time waveforms of \( x_i, i = 1, \cdots, 6 \) for \( \varepsilon = 0.01 \) and \( \alpha = 0.01 \). Being different from [6], the shape of these envelopes are different, and simple phase synchronization of them cannot be recognized. That is, the envelopes appears in the form of almost periodic oscillation similarly to them for strong nonlinear case. The power spectrum of \( x_1 \) for this case is depicted in Fig.6. The spectrum shows four dominant peaks, therefore, this is a four-fold multimode oscillation.

Interestingly, the dominant peaks are nearly equal to angular frequencies of resonance modes \( (\omega_i, i = 1, \cdots, N) \) with \( \varepsilon = 0 \) in Eq.(1). \( \omega_i \) can be explicitly expressed by the square root of the eigenvalues of \( K_N \) in Eq.(1) in the following[6].

\[
\omega_i = \sqrt{1 + \alpha - 2\alpha \cos \frac{2\pi(i-1)}{N}}
\]

(2)

Then, the number of different modes becomes 1 + \([N/2]\) \((\cdot)\) is Gauss’s symbol). Therefore, for \( N = 6 \) there are...
In addition, $\omega^1, \omega^2 (= \omega^6), \omega^3 (= \omega^5)$ and $\omega^4$ in Eq.(2) are shown.

Table 1: Equilibrium points and their stabilities for $\beta = 3.2$ obtained by the perturbation method[6]. The bracketed number indicates the multiplicity of eigenvalues.

<table>
<thead>
<tr>
<th>$M$</th>
<th>Equilibrium points</th>
<th>Eigenvalues</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$r^0_1 = 1.305$</td>
<td>0.637</td>
</tr>
<tr>
<td></td>
<td>$r^0_1 = 2.167$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$r^0_1 = 0.782$</td>
<td>0.533, -0.302</td>
</tr>
<tr>
<td>2</td>
<td>$r^0_1 = r^0_2 = 1.144$</td>
<td>-0.191, -1.141</td>
</tr>
<tr>
<td></td>
<td>$r^0_1 = 1.673, r^0_2 = 0.632$</td>
<td>0.287, -1.407</td>
</tr>
<tr>
<td>3</td>
<td>$r^0_1 = 1.575, r^0_m = 0.424$</td>
<td>0.221 (2), -1.112</td>
</tr>
<tr>
<td>4</td>
<td>$r^0_1 = 1.550, r^0_m = 0.338$</td>
<td>0.192 (3), -1.124</td>
</tr>
<tr>
<td>5</td>
<td>$r^0_1 = 1.539, r^0_m = 0.288$</td>
<td>0.174 (4), -1.186</td>
</tr>
</tbody>
</table>

four different angular frequencies ($\omega_1, \omega_2, \omega_3$ and $\omega_4$). For $\alpha = 0.01$ these values are shown in Fig.6, which correspond to the four dominant frequencies of the propagating pulse wave for $\varepsilon = 0.01$. The result may suggest that theoretical analysis can be applied to this kind of propagating wave.

4. Conclusions

It is demonstrated that the propagating pulse wave shows the multimode oscillation. These frequency components approach to 1.0 rad/sec, as the degree of nonlinearity becomes weak. For weak nonlinear case, the spectrum of the solution shows four dominant peaks. In this meaning, this is a different type of multimode oscillation (especially for $M = 2$) from those in [6]. In while, the dominant peaks agree with $\omega_i$ in Eq.(2), which is remained to be investigated in detail in the future.

Acknowledgments

The authors appreciate the developers of software “BUNKI” for our usage to calculate the bifurcation set in Fig.3. One of the authors (Shimizu) is thankful to Prof. Y. Haga in Chiba institute of technology, Japan for his continued supports and encouragements.

References

Bifurcation analysis of two coupled Izhikevich Oscillators

Daisuke Ito† and Tetsushi Ueta †† and Kazuyuki Aihara ‡

†Dept. Info. Sci&Intelligent Costs. Tokushima University
Email: d-ito@is.tokushima-u.ac.jp
††Center for Adv. Info. Tech. Tokushima University
Email: tetsushi@is.tokushima-u.ac.jp
‡Institute of Industrial Science. University of Tokyo
Email: aihara@sat.t.u-tokyo.ac.jp

Abstract—A simple oscillator of spiking neurons is proposed by Izhikevich. By some numerical experiments, all firing patterns which have been observed in the brain are confirmed at the origin of the mathematical model. Although, a detailed bifurcation analysis has been given by the author, no investigation on its coupling system has been done. In this paper, we consider two Izhikevich neurons coupled by a gap junction. By choosing an appropriate Poincaré section, we can compute bifurcation set for limit cycles. As a result, period-doubling bifurcation and its cascade to chaos is observed by changing the coupling coefficient. We show bifurcation diagrams and numerical simulation results.

1. Introduction

Non-smooth dynamical systems are often derived from actual physical objects, biological activities, electrical systems, and so on. In the mathematical model, there are at least one point which is not differentiable by the state so that it is difficult to treat by using the conventional bifurcation theory. Some theoretical approaches to address this difficulties are available [1, 2], but we also develop a numerical algorithm [3] to maintain non-smooth dynamics by using numerical integration of piecewise differentiable variational equations, i.e., the differential values of the Poincaré mapping required for solving bifurcation problem are evaluated by the solutions of variational equations without any discontinuities. Thus any system including impulses, hysteresis and jumping can be analyzed easily.

A simple oscillator of spiking neurons is proposed by Izhikevich [4]. The biologically plausibility of the oscillator is shown by some numerical experiments. Firing patterns of all known types of neurons are illustrated by changing parameters. The bifurcation sets of this model have been already analyzed [5].

Practical neurons are supposed to be connected any other neurons, however the bifurcation analysis of a junctional Izhikevich oscillator has not been investigated in detail. In this paper, the target is two Izhikevich neuronal oscillators coupled by a gap junction. We investigate bifurcation structures of these models.

2. The method for analysis

If one wants to consider stability or bifurcation problems on the Izhikevich oscillator, calculation becomes complicated because of their non-smoothness. However, analyzing method for the model with nonsmooth function is already provided [3]. It is confirmed that can be calculated by considering switching threshold Poincaré mapping.

Let us consider $m$-tuple of autonomous differential equations.

$$\frac{dx}{dt} = f(x, \lambda, \alpha), \quad k = 0, 1, \ldots, m - 1$$ (1)

where $t \in \mathbb{R}, x \in \mathbb{R}^n, \lambda \in \mathbb{R}^r$ is an invariant parameter for $f_0, f_1, \ldots, f_{m-1}$ and $\alpha \in \mathbb{R}^s$ are parameters depending only on $f_k$, $r$ and $s$ are integers. We call these equations piecewise-defined differential equations. Assume that $f_k$ are $C^\infty$-class map for all variables and parameters and every equations in Eq.(1) has a solution with an appropriate initial value $x_{k0}$.

If the orbit is periodic, the following a $k$-tuple of local mappings are obtained as follows:

$$T_0 : \quad \Pi_0 \rightarrow \Pi_1$$
$$\quad x_0 \mapsto x_1 = \varphi_0(t_0, x_0)$$

$$T_1 : \quad \Pi_1 \rightarrow \Pi_2$$
$$\quad x_1 \mapsto x_2 = \varphi_1(t_1, x_1)$$

$$\quad \vdots$$

$$T_{k-1} : \quad \Pi_{k-1} \rightarrow \Pi_0$$
$$\quad x_{k-1} \mapsto x_0 = \varphi_{k-1}(t_{k-1}, x_{k-1})$$

(2)

where $\tau = \sum_{j=0}^{k-1} t_j$ is the period of the limit cycle. Then we define the following composite mapping as the Poincaré mapping correlated with Eq.(2).

$$T = T_{m-1} \circ \cdots \circ T_1 \circ T_0$$ (3)

When $x_0$ which started from local section $\Pi_0$ passes local section $\Pi_0$ again and follow Eq(4), $x_0$ is called a fixed point.

$$x_0 = x_m = T(x_0)$$ (4)

The derivative with the initial value of the Poincaré map is given by

$$\frac{\partial T}{\partial x_0} \bigg|_{x_0=x} = \prod_{k=m-1}^{0} \frac{\partial T_k}{\partial x_k} \bigg|_{x_0=x}$$ (5)
Each Jacobian matrix can be written from \[3\] as follows:

\[
\frac{\partial T_k}{\partial x_k} = \left[ I_n - \frac{1}{f_k} \frac{\partial q_{k+1}}{\partial x} \right] \frac{\partial \phi_k}{\partial x_k}
\]

where \(I_n\) is an \(n \times n\) identity matrix. \(\frac{\partial \phi_k}{\partial x_k}\) can be obtained by solving the following differential equation:

\[
\frac{d}{dt} \left( \frac{\partial \phi_k}{\partial x_k} \right) = \frac{f_k}{\partial x} \frac{\partial \phi_k}{\partial x_k} \quad \text{with}
\]

\[
\left[ \frac{\partial \phi_k}{\partial x_k} \right]_{l=0} = I_n, \quad k = 0, 1, 2, \ldots, m - 1
\]

Now, we define a local coordinate \(u \in \Sigma \subset \mathbb{R}^{n-1}\) corresponding to \(\Pi_0\) by using a projection \(p\) and embedding map \(p^{-1}\)

\[
p^{-1} : \Sigma \rightarrow \Pi_0, \quad p : \Pi_0 \rightarrow \Sigma
\]

Accordingly, the Poincaré mapping on the local coordinate is obtained as

\[
T_1 : \Sigma \rightarrow \Sigma
\]

\[
\gamma \rightarrow p \circ T \circ p^{-1}(u)
\]

The Jacobian matrix is given by

\[
\frac{\partial T_1}{\partial u_0} = DT_1(u_0) = \frac{\partial p}{\partial x} \frac{\partial T}{\partial x} \frac{\partial p^{-1}}{\partial u}
\]

The characteristic equation is given by

\[
\chi(\mu) = [DT_1 - \mu I_{n-1}] = 0
\]

The roots of Eq.(11) \(\mu_1, \mu_2, \ldots, \mu_{m-1}\) give multipliers of the fixed points. The derivative with the parameter value of the Poincaré Map is given by

\[
\frac{\partial T}{\partial \lambda} = \frac{\partial T_{m-1}}{\partial x_{m-1}} \frac{\partial T_{m-2}}{\partial x_{m-2}} + \frac{\partial T_{m-1}}{\partial x_{m-1}}
\]

Each Jacobian matrix can be written as follows:

\[
\frac{\partial T_1}{\partial \lambda} = \frac{\partial T_{m-1}}{\partial x_{m-1}} \frac{\partial T_{m-2}}{\partial x_{m-2}} + \frac{\partial T_{m-1}}{\partial x_{m-1}}
\]

\[
\frac{\partial T_0}{\partial \lambda} = \frac{\partial T_0}{\partial \lambda}
\]

We used the numerical differentiation as substitute for the second variational because the calculation becomes more complex, and it is enough.

### 3. Junctional Izhikevich oscillator

A simple oscillator of spiking neurons is proposed by Izhikevich [4]. There are two advantages in this oscillator: first, it does not cost the calculation more than Hodgkin-Huxley-type oscillator. Second, many firing patterns measured in biological activities are reproduced by this model.

The equation sets are as follows:

\[
\begin{align*}
\psi' & = 0.04 \psi^2 + 5 \psi + 140 - u + I \\
u' & = a(bv - u)
\end{align*}
\]

with the auxiliary after-spike resetting

\[
\begin{align*}
\text{if } \psi \geq 30 \text{mV, then} \quad \begin{cases}
\psi & \leftarrow c \\
u & \leftarrow u + d
\end{cases}
\end{align*}
\]

where, the state variables \(\psi\) and \(u\) correspond to the membrane potential of the neuron and membrane recovery variable, respectively. \(a, b, c, d\) and \(I\) are parameters.

Next, We consider two Izhikevich neurons coupled by a gap junction. The equation set are as follows:

\[
\begin{align*}
\psi_0' = 0.04 \psi_0^2 + 5 \psi_0 + 140 - u_0 + I + \delta (\psi_1 - \psi_0) \\
u_0' = a(bv_0 - u_0)
\end{align*}
\]

\[
\begin{align*}
\psi_1' = 0.04 \psi_1^2 + 5 \psi_1 + 140 - u_1 + I + \delta (\psi_0 - \psi_1) \\
u_1' = a(bv_1 - u_1)
\end{align*}
\]

with the auxiliary after-spike resetting

\[
\begin{align*}
\text{if } \psi_0 \geq 30 \text{mV, then} \quad \begin{cases}
\psi_0 & \leftarrow c \\
u_0 & \leftarrow u_0 + d
\end{cases}
\end{align*}
\]

\[
\begin{align*}
\text{if } \psi_1 \geq 30 \text{mV, then} \quad \begin{cases}
\psi_1 & \leftarrow c \\
u_1 & \leftarrow u_1 + d
\end{cases}
\end{align*}
\]

where, the parameter \(\delta\) is coupling coefficient. When \(\delta \geq 1\), each other are connected strongly. By prechecking for this system, not much impressive dynamical behavior with \(\delta > 0\), we allow all values for \(\delta\) without biological assumption.

In the following, we place local section defined by the reset operation:

\[
\Pi_0 = [x = (v_0, u_0, v_1, u_1)^T \in \mathbb{R}^4 | q_0(x) = v_0 - 30 = 0] \\
\Pi_1 = [x \in \mathbb{R}^4 | q_1(x) = v_1 - 30 = 0]
\]

Local mappings are defined as follows:

\[
\begin{align*}
T_{00} : \Pi_0 & \rightarrow \Pi_0 \\
T_{10} & : \Pi_1 \rightarrow \Pi_0 \\
T_{01} & : \Pi_0 \rightarrow \Pi_1 \\
T_{11} & : \Pi_1 \rightarrow \Pi_1
\end{align*}
\]

\[
\begin{align*}
v_0 & \leftarrow c \\
u_0 & \leftarrow \varphi_1(t, x_0, A) + d \\
v_1 & \leftarrow \varphi_2(t, x_0, A) \\
u_1 & \leftarrow \varphi_3(t, x_0, A)
\end{align*}
\]

\[
\begin{align*}
v_0 & \leftarrow \varphi_0(t, x_0, A) \\
u_0 & \leftarrow \varphi_1(t, x_0, A) \\
v_1 & \leftarrow c \\
u_1 & \leftarrow \varphi_3(t, x_0, A) + d
\end{align*}
\]

Thus, comparison local mapping is defined as follows:

\[
T = T_{10} \circ T_{11} \circ \cdots \circ T_{00}
\]

We choose the projection and embedding as follows:

\[
\begin{align*}
h : \Pi_0 & \rightarrow \Sigma \\
x & = (v_0, u_0, v_1, u_1)^T \mapsto w = (u_0, v_1, u_1)^T
\end{align*}
\]

\[
\begin{align*}
h^{-1} : \Sigma & \rightarrow \Pi_0 \\
w & \mapsto x
\end{align*}
\]
4. Bifurcation of fixed point

We compute the bifurcation diagrams in $\delta-a$ plane. Figure 1 shows diagrams whose parameters are fixed to $b = 0.2$, $c = −50$, $d = 2$, $I = 10$. A simple Izhikevich oscillator with this parameters shows a period one attractor. However, in the case of two Izhikevich oscillators coupled by a gap junction, the bifurcation phenomena is occurred by changing the parameter $\delta$. In fig.1, the limit cycle bifurcate into chaos attractor by a PD cascade in the right side of chaos region(i). Figure 2 (a) and (b) show a period two and a period four attractors, respectively. Via the PD cascade, a chaotic attractor appears(Fig.2(c)). Figure 3 depicts the one-dimension bifurcation diagram and the maximum Lyapunov exponent value with $a = 0.2$. It is confirmed that a chaos attractor is appeared by a PD cascade.

In $a < 0.173$, a chaos region(ii) exists between period three and period six attractors. The period three attractor disappears by a tangent bifurcation and alternatively a chaotic attractor is shown. Figure 4 shows the one-dimension bifurcation diagram and the maximum Lyapunov exponent value with $a = 0.17$. It is confirmed that a chaos region exists between a period three and a period six regions.

Figure 5 shows bifurcation sets with $b = 0.15$, $c = −50$, $d = 2$, $I = 10$. In this parameter, The bifurcation structure is shape similar to fig.1 but the chaos region(ii) is disappeared. Moreover, in the left side of fig.5, a period one region exists. Only the one neuron spikes with period one in this region. When the switching thresholds are changed, a similar bifurcation structure to fig.1 and 5 are also obtained.

Figure 1: Bifurcation Diagram in the $\delta-a$ plane with $b = 0.2$, $c = −50$, $d = 2$, $I = 10$.

Figure 2: Phase portraits with $a = 0.2$, $b = 0.2$, $c = −50$, $d = 2$, $I = 10$. (a) Period 2, $\delta = −0.115$. (b) Period 4, $\delta = −0.12$. (c) Chaos, $\delta = −0.125$.

Figure 3: One dimension bifurcation diagram and Lyapunov exponent value with $a = 0.2$, $b = 0.2$, $c = −50$, $d = 2$, $I = 10$. 
Lyapunov exponent $\delta$

Figure 4: One dimension bifurcation diagram and Lyapunov exponent value with $a = 0.16, b = 0.2, c = -50, d = 2, I = 10$.

Figure 5: Bifurcation Diagram in the $\delta-a$ plane with $b = 0.15, c = -50, d = 2, I = 10$.

Figure 6: Bifurcation Diagram in the $TH_0-TH_1$ plane with $a = 0.2, b = 0.2, c = -50, d = 2, I = 10, \delta = -0.1$

In the future, we want to compute bifurcation sets at other parameters and investigate the model connected more neurons.

Acknowledgments

This research is supported by the Japan Society for the Promotion of Science (JSPS) through its “Funding Program for World-Leading Innovative R&D on Science and Technology (FIRST Program).”

References


Amplitude death in a pair of time-delayed chaotic oscillators coupled by a static connection

Luan Ba Le, Keiji Konishi, Hideki Kokame and Naoyuki Hara

Department of Electrical and Information Systems
Graduate School of Engineering, Osaka Prefecture University
1-1 Gakuen-cho, Nakaku, Sakai, Osaka, 599-8531 Japan
Email: konishi@eis.osakafu-u.ac.jp

Abstract—Amplitude death, the coupling induced stabilization of unstable fixed points, is observed in a pair of time-delayed chaotic oscillators coupled by the static connection. The cluster treatment of characteristic roots methodology allows us to estimate the boundary curves of the death region in a parametric space. This region can be utilized to streamline the design of the long delay times appropriately for inducing amplitude death. The analytical results are verified by electronic circuit experiments.

1. Introduction

Amplitude death, a diffusive connection induced stabilization of unstable fixed points in coupled oscillators, has been a subject of extensive investigation in the last 15 years [1, 2]. Aronson et al. analytically investigated death phenomenon for two coupled nonlinear oscillators [2]. They reported that death never occurs in coupled identical oscillators. Reddy, Sen, and Johnston showed that a time-delayed coupling effect, which exists owing to the finite speed of data propagation, is able to induce the amplitude death even in coupled identical oscillators [3]. Their report has considerably intrigued in the field of nonlinear physics [4]. Atay showed that the distributed time-delay connections facilitate amplitude death [5]. Konishi et al. reported that the multiple delay [6] and time-varying delay [7] connections can also facilitate amplitude death in coupled oscillators.

In order to use death in practical situations, one has to design a mutual connection that induces death. However, one inevitably confronts two problems for such a design. The first problem is how to select the type of mutual interactions and how to determine the coupling parameters. The second one is how to deal with high-dimensional oscillators. The previous paper provided a solution for the above problems [8] and, in particular, focused on the amplitude death in a pair of time-delayed chaotic oscillators [9] coupled by three types of mutual interactions: static connections, dynamic connections, and delayed connections.

Since time delay systems have been widely employed to model undesirable nonlinear phenomena such as chatter and chaos in metal cutting tools [10, 11], the stabilization of time-delay systems has been an important issue for engineering applications. The previous paper concluded that the amplitude death can occur with the dynamics and delayed connections, but cannot with the static one [8]. Although the dynamic and delayed connections are not easy to implement in experimental situations due to their complicated structure, the static connection has a simple structure. Therefore, the static one is the best solution from a cost standpoint.

The present paper theoretically and analytically considers the stability of a pair of non-identical time-delayed oscillators coupled by the static connection. The traditional procedure for the stability analysis is difficult to employ for deriving the boundary curves of death, since its characteristic equation includes a cross-talk term between the non-identical delays. To overcome this difficulty, the present paper applies the methodology proposed by Sipahi and Olgac [12] to the characteristic equation. This methodology allows us to derive the boundary curves of death which are useful to design the delay times without trial-and-error testing. Furthermore, our theoretical results are verified by electronic circuit experiments.

2. Time-delayed chaotic coupled oscillators

Let us consider the pair of non-identical scalar time-delayed chaotic oscillators (see Fig. 1) [8]:

\[
\begin{align*}
\dot{x}_1 &= f(x_{1\tau_1}) - \alpha x_1 + u_1, \\
\dot{x}_2 &= f(x_{2\tau_2}) - \alpha x_2 + u_2,
\end{align*}
\]

Figure 1: Block diagram of a pair of oscillators (1) coupled by connection (2).

where \(x_{1,2} \in \mathbb{R}\) and \(u_{1,2} \in \mathbb{R}\) are the system states and coupling signals, respectively. \(x_{1\tau_1,2\tau_2} := x_{1,2}(t - \tau_{1,2})\) are the...
delayed states, $\tau_{1,2} \geq 0$ are the delay times and $\alpha > 0$ is a parameter. $f : \mathbb{R} \to \mathbb{R}$ represents a nonlinear function. The symbol $\mathbb{R}$ denotes the set of real numbers. These oscillators are coupled by the static connection described as

$$u_{1,2} = k(x_{1,2} - x_{2,1}),$$

(2)

where $k \in \mathbb{R}$ is the coupling strength.

Each individual oscillator without coupling (i.e., $u_{1,2} \equiv 0$) has the fixed points

$$x^* : 0 = f(x^*) - \alpha x^*.$$  

(3)

Throughout this paper, it is assumed that there is one unstable fixed point. The location of the fixed point $x^*$ never changes even by coupling; in other words, the static connection changes only the stability of the point. The characteristic equation of linearized system at the fixed point $x^*$ can be rewritten as

$$a_0(\lambda) + a_1(\lambda)e^{-\lambda \tau_1} + a_2(\lambda)e^{-\lambda \tau_2} + a_3e^{-\lambda(\tau_1 + \tau_2)} = 0,$$

(4)

where

$$a_0(\lambda) := x^2 + 2(\alpha - k)\lambda + (\alpha - k)^2 - k^2,$$

(5)

$$a_1(\lambda) = -\beta \lambda,$$

(6)

$$a_2(\lambda) := -\beta \lambda - \beta(\alpha - k),$$

(7)

$$\beta := \frac{|d f(x)/dx|_{x=x^*}}{2}.$$  

(8)

The nonlinear function $f$ is given by

$$f(x) = \begin{cases} 0 & \text{if } x \leq -4/3 \\ -1.8x - 2.4 & \text{if } -4/3 < x \leq -0.8 \\ 1.2x & \text{if } -0.8 < x \leq 0.8 \\ -1.8x + 2.4 & \text{if } 0.8 < x \leq 4/3 \\ 0 & \text{if } x > 4/3 \end{cases},$$

(9)

as a typical case study [13]. The three fixed points of individual oscillator located at the intersection of $f(x)$ and $ax$: $x^* = \pm 6/7$ and 0. The slopes of $f(x)$ at $x^*$ can be estimated as $\beta(\pm 6/7) = -1.8$ and $\beta(0) = 1.2$. The parameters are fixed at $\alpha = 1$ and $k = -2$. Additionally, characteristic equation (4) in the absence of delays (i.e., $\tau_1 = \tau_2 = 0$) has two roots -6.79 and -2.80, thus the number of unstable roots is zero.

3. Death region and boundary curves

The cluster treatment of characteristic roots (CTCR) paradigm is capable of deriving the boundary curves of a stability region in $(\tau_1, \tau_2)$ space [12]. To simplify the stability analysis, we only identify the roots lied on the imaginary axis: $\lambda = j \lambda_{\text{lim}},$ where the symbol $j$ is denoted as $j = \sqrt{-1}$. The stability posture in $(\tau_1, \tau_2)$ space is shown in Fig. 2, where the amplitude death region is symbolized by $\Omega$. Every point on the boundary curves correspond to the purely imaginary root. The numbers of unstable roots are stated in several regions.

Figures 3 and 4 illustrate the time series data of the two non-identical oscillators for the two parameter sets: (A) $\tau_1 = 4.4, \tau_2 = 6.8$; (B) $\tau_1 = 3.4, \tau_2 = 6.8$. These sets are indicated in Fig. 2. For the parameter set (A), the two identical oscillators without coupling behave chaotically until $t = 500$, and become periodical after coupling. On the other hand for the parameter set (B), the two individual oscillators before coupling has the chaotic motion. After coupling the states $x_{1,2}(t)$ converge on $x^*_{1,2}$.

From the death region on a wide scale as shown in Fig. 5, it seems that one may choose $(\tau_1, \tau_2)$ arbitrarily large to obtain death, if they are within the strips (see the dotted lines in Fig. 5), $\tau_{1,2} = 2\tau_{1,2} + \xi$, $0 \leq \xi \leq 2$. The static connection can induce death over a long delay-time region $\Omega$. However, we do not still have the proof for this fact.
Figure 4: Time series data $x_{1,2}(t)$ of coupled oscillators (numerical simulation): (B) $\tau_1 = 3.4$, $\tau_2 = 6.8$.

Figure 5: Death region and the boundary curves of stability on a wide scale.

4. Electronic circuit experiments

The two chaotic oscillators coupled by the static connection are sketched in Fig. 6. The delay units employ the bucket brigade delay line MN3011 (Panasonic) to generate the delayed signal [8] and the nonlinear function is implemented by the three opamps and the two diodes.

Here, $x_i$ denotes the voltage of the $i$-th oscillator. The boxes labeled $-\tau_i$ and $f$ are the time delay unit and the nonlinear function unit, respectively. These oscillators are governed by

$$
\begin{align*}
C_n \dot{x}_1(t) &= \frac{1}{R_n} \left[ f(x_1(t-\tau_1)) - x_1(t) \right] + u_1(t), \\
C_n \dot{x}_2(t) &= \frac{1}{R_n} \left[ f(x_2(t-\tau_2)) - x_2(t) \right] + u_2(t),
\end{align*}
$$

(10)

where $R_n, C_n$ are a resistor and capacitor, respectively. The coupling terms $u_{1,2}(t)$ are the currents from the connection circuit to the oscillators. The static connection is described by

$$
u_{1,2}(t) = -\frac{1}{R} \left[ x_{1,2}(t) - x_{2,1}(t) \right].
$$

(11)

Figure 6: Two time-delay chaotic oscillators coupled by a static connection.

Figure 7: Nonlinear function $f(x)$. Horizontal axis: $x$ (1V/div); vertical axis: $f(x)$ (1V/div).

It is sufficient to treat the above circuits as the dimensionless oscillators (1) on the following relations:

$$
\begin{align*}
\bar{\tau}_i &:= \frac{\tau_i}{R_n C_n}, \\
\dot{x}_{1,2} &:= \frac{dx_{1,2}(\bar{t})}{d\bar{t}}, \\
x_{1,2} &:= x_{1,2}(\bar{t}), \\
x_{1,2} &:= x_{1,2}(\bar{t} - \bar{\tau}_{1,2}), \\
u_{1,2} &:= u_{1,2}(\bar{t}), \\
k &:= -\frac{R_n}{R}.
\end{align*}
$$

These relations show that circuit equation (10) is identical to equation (1) with $\alpha = 1$. The input-output characteristic of the nonlinear function unit is shown in Fig. 7. From the characteristic, the fixed point $x^* = 2.2$ V and the slope $\beta(x^*) = -1.8$ are approximately estimated. The circuit parameters and the coupling resistor are fixed at $R_n = 1.0$ k$\Omega$, $C_n = 1.0$ $\mu$F, and $R = 0.5$ k$\Omega$. We consider two parameter sets: (A) $\tau_1 = 4.4$ ms, $\tau_2 = 6.8$ ms; (B) $\tau_1 = 3.4$ ms, $\tau_2 = 6.8$ ms. In dimensionless oscillators (1), $\tilde{\tau}_{1,2}$ for the two parameter sets correspond to (A) $\tilde{\tau}_1 = 4.4$, $\tilde{\tau}_2 = 6.8$; (B) $\tilde{\tau}_1 = 3.4$, $\tilde{\tau}_2 = 6.8$, which indicated as the points A and B on Fig. 2.

Figures 8(a) and 8(b) show that the individual oscillator at the parameter set (A) exhibits chaotic behavior. When the switch S shown in Fig. 6 is closed, the two individual circuits are connected by the coupling resistor. After S is closed, the chaotic behavior changes to a periodic motion, but death is not observed as shown in Fig. 8(c). For the parameter set (B), the individual oscillator exhibits chaotic behavior (see Figs. 9(a) and 9(b)). The oscillators behave chaotically until S is closed. As shown in Fig. 9(c), the states $x_{1,2}(t)$ gradually converge on $x_{1,2}^*$: the am-
Figure 8: Experimental verification at parameter set (A): (a) chaotic behavior of oscillator 1; (b) chaotic behavior of oscillator 2 (horizontal axis: $x(t - \tau)$ (0.5V/div), vertical axis: $x(t)$ (0.5V/div)); (c) time series data $x_{1,2}(t)$ (V) just before and after coupling. Amplitude death occurs, after $S$ is closed. This experimental study confirms that the estimated death region by analytical insight in the preceding section agrees well with the electronic circuit experiments.

5. Conclusion

The present paper investigated the static connection induced amplitude death in the two non-identical time-delayed chaotic oscillators. The precise shape of the death region derived by the CTCR paradigm leads to the conclusion that even the relatively long delay times can induce the amplitude death. This perspective seems to be important, because systems with large delay are frequently met in various fields of applications. Moreover, our analytical results are experimentally confirmed by the real electronic oscillators.

References