2014 International Symposium on Nonlinear Theory and its Applications
NOLTA2014, Luzern, Switzerland, September 14-18, 2014

Modeling Interacting Dynamical Systems with Wireless Active Networks

A. S. Dmitriev, M. Y. Gerasimov, R. Yu. Emelyanov, Yu. V. Andreyev

Kotelnikov Institute of Radio Engineering and Electronics, Russian Academy of Sciences,
Morhovaya St. 11-7, 125009, Moscow, Russia

Moscow Institute of Physics and Technology,
Institutskii st. 9, Dolgoprudnyi, 141700, Moscow region, Russia
chaos@cplire.ru

Abstract– Generalized wireless sensor networks – wireless active networks (WAN) – are considered as distributed multiple-unit processor platform for modeling behavior of interacting dynamic systems. During simulation, each sensor network node represents a dynamic system, whose equations are solved by a special computing unit, e.g., a microcontroller (MCU) placed on each node. Interaction between the dynamic systems is understood as transmission of the system state data between the nodes of active network via radio channel. This approach is investigated here on example of an ensemble of coupled logistic maps. The ensemble is implemented as a wireless network composed of ultra wideband direct chaotic transceivers PPS-43 [7] with special actuator boards attached. Actuator boards contain an MCU to iterate the maps and a display to visualize information (color LED). The modeling technique, experimental results and analysis are described.

1. Introduction

Usually, wireless sensor network (WSN) is described as a set of nodes equipped with sensors and combined into an ensemble by means of interaction via radio channels [1]. However, there is an increasing number of examples of wireless sensor networks whose nodes (along with transceivers and sensors) include actuators, i.e., devices which affect the environment; or visual information displays (LED, LCD); or information processing devices (MCU, processor) [2]. Below, for such generalized wireless sensor networks we use the term **wireless active networks** (WAN). Wide range of equipment used in the nodes along with communication capabilities allows us to consider WANs as not only means of data acquisition in particular area, but as a powerful platform for solving wide classes of problems related with multiple-unit interacting systems.

2. Previous Results

One of the research directions closely connected with the theme of this work is cellular neural networks (CNN) [3-4], which rely on two basic concepts: local connectivity and analog circuit dynamics. After their appearance, the theory and applications of cellular neural networks were developed rapidly, gradually addressing still wider set of problems and challenges. Then, a generalization of cellular neural networks on cellular nonlinear networks (CNN) as a universal machine and supercomputer was proposed [5]. The idea was to create algorithmically programmable analog cellular computer that had the power of a supercomputer operating in real time and on a single chip. Thus, unlike cellular neural networks, CNN and universal machine are programmable and also have analog memory (global and distributed) and logics; complex cells coupled at high data rates by means of electromagnetic waves, which can be used for modeling wide class of partial differential equations. A wide range of other potential applications is described, including “programmable physics”, “programmable chemistry”, “programmable bionics” [5].

Analysis of relations between wireless networks and CNN is given in [6]. As is noted in [6], in large-scale wireless networks due to energy and interference constraints the access domain of a network node includes only nearest neighbors, and control algorithms are distributed and local. Obviously, this structure is similar to cellular neural networks.

3. Problem Statement and Simulation Scheme

Assume, for instance, an ensemble of interacting dynamic systems. Each system is described with a system of evolution equations, e.g., maps or nonlinear ODEs. The task is to implement this ensemble in WAN and to use this WAN to observe and study the ensemble dynamics.

Ensemble implementation with WAN includes (fig. 1):

- programming equations of each ensemble element for WAN node processor, corresponding to this element;
- determining and establishing links between WAN nodes, according to the links between ensemble elements;
- implementing the links with radio channels.

Ensemble equations and interactions are programmed in three stages:

- coding and debugging program in high-level programming language (C++);
- compilation to machine code;
- downloading the code in processor.

The ability to realize links between network nodes via radio channel is the main moment in building models of interacting dynamic systems with WAN. In general, with radio channels one can simulate ensembles of any link topology.

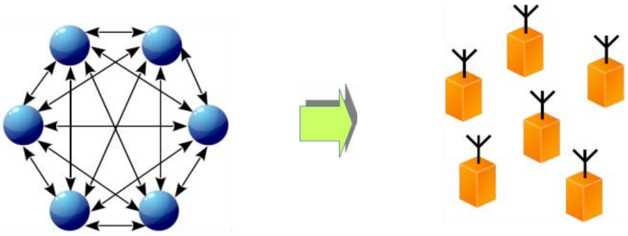

Fig. 1. From globally connected dynamic ensemble to WAN maps

To be sure, assume that all WAN nodes directly radio visible to each other. In this case, to transmit data from node *i* to other nodes, a broadcast packet with source node address can be used. All nodes receive this packet. The nodes, which must be coupled with source node *i* (according to model), use information in the packet from source *i* to form their state on the next step. And the nodes, which must not be linked with node *i*, according to the model, just ignore this packet.

So, according to the model:

- nodes are arranged in space in such a way that each node can “see” any other node of the network. Each node can send and receive data to/from the nodes it must be linked with;
- equations of an ensemble element are integrated (iterated) on CPU of the corresponding node;
- each node operates independently, making iteration steps in regular time intervals Δt;
- in the beginning of each time interval each node transmits information of the states of its variables and then goes to listen mode during the rest of the time interval;
- as a result, during its “own” time interval Δt each node of the network transmits information about its state to all nodes which it is linked with and receives the state information from them;
- the state of node variable is visualized by LED color.

4. Ultrawideband Active Node and Actuator for Simulation

An active network node consists of a wireless transceiver and a special actuator board (fig. 2). Ultrawideband wireless transceiver PPS-43 [7] is used as a communication part. The actuator is based on a microcontroller STM32L which is used as a computing unit to solve equations of an ensemble element and color LED as visual indicator for one of the variables of the ensemble element state.

MCU STM32L has RISC architecture; its clock frequency can be changed in the range 1 to 32 MHz. It has good computing capabilities. It is extraordinary small and energy efficient.

An important feature of this MCU is the ability to emulate floating point operations, so it can be used as full functional device to simulate dynamic systems. The equations are programmed in C++ and then compiled to MCU machine code.

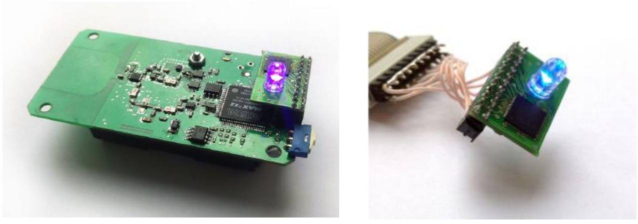

Fig 2. Ultrawideband direct chaotic transceiver PPS-43 and actuator board with color LED

The processor board indication element is a RGB LED, whose color is related to the value of an element variable of the ensemble. LEDs give a visual dynamic picture of cooperative behavior of the network. For example, this color picture allows us to clearly register such phenomena as synchronization and clustering of ensemble elements. In fact, visualization of variable state with LED color is a hardware implementation of a widely used computer method of representing the state of dynamic systems.

5. Simulation Procedure

Simulation of collective behavior of discrete dynamic systems includes the following stages:

- Network creation: at this stage each network node declares itself and establishes connection with each other network node. For each node and for the network as a whole the process ends when each node detects all the other nodes and each node has the “knowledge” that all other nodes have received information from it.
- Synchronization of node clocks.
- Start of collective iteration and evolution.
- Brake of collective behavior in case one or more links are broken.
- Restoring collective iterations.

An important feature of described network is its autonomy, so there is no need in control center (sink).

5.1. Network Creation

Consider a network of n nodes. Nodes are turned on one by one at moments $t_1, t_2, \dots t_n$. Each node listens during time interval T and then transmits information about itself for period δT. Then the operation repeats.

We assume that all nodes are located within the area of direct visibility of each other and $\delta T \ll T$. This allows to neglect collisions between packets.

How long does it take to guarantee that all nodes have information about the presence of the other nodes?

Let node k be the last node turned on and assume that this event happened at moment t_k. In general, for all other $n - 1$ nodes the starting time points are to the left of point t_k. Cycles of all nodes end within time interval $(t_k, t_k + T)$.

Therefore node k will receive information about all nodes before the moment it sends information about itself. If some node doesn't get the information from another of $n - 2$ remaining nodes, it will get it on time interval (t_k, $t_k + T$) in the same way as node k.

Thus, at time T after turning on the last node, each node in the network has got information about the presence of all nodes in the network and is ready to the next stage.

Packet reception from the last node in the network list indicates the end of the network creation process.

5.2. Node Clock Synchronization

Clock synchronization is a well-known problem for wireless sensor networks that work in synchronous mode. Certain approaches to synchronization use methods of nonlinear dynamics [8, 9].

In this report, time synchronization is solved as follows. First, time intervals for emission are defined. The devices negotiate the order in which they transmit. The order is determined by means of sorting ID list (from 1 to n); position in this list is the device ID number.

Device number 1 emits first. The emission moment is known to the other elements of the ensemble, because it is determined from the moment of reception of the previous signal from the device with the minimum ID. Other nodes adjust their clocks with this packet and get actually synchronized.

To increase LED glow time and to compensate clock discrepancy between devices, the second element transmits in time ΔT after the first element, the third element ΔT after the second, and so on.

Hence time interval for intercommunication between nodes is equal to $\Delta T(n - 1)$ which is set significantly smaller than the period of synchronization cycle T.

5.3. Start of Collective Iteration and Evolution

Let there be WAN with n nodes. Each node is composed of a wireless transceiver and an actuator. i-th node map is described by expression

$$x_i(k+1) = f(\alpha_{ii}x_i(k) + \sum_{i\neq j}\alpha_{ij}x_j(k)) \qquad (1)$$

where k is iteration step. i-th node processing unit iterates map (1) with parameters corresponding to a chaotic mode, with random initial conditions. Each node is assigned an ID number (1 to n). Node variable value is displayed with RGB LED color. After clock synchronization and setting order, nodes communicate with each other and exchange data about their state via radio links. The received values of the variables are used to iterate maps (1). Weights α_{ij} (coupling strengths) are set such as to provide map synchronization through collective iteration. For certainty, in the experiments all the coupling strengths of all the maps are equal ($\alpha_{ij} = \alpha_{ij} = \alpha$, $\forall i, j, i \neq j$).

Conditions for global synchronization, i.e., the range of α for synchronous mode, were obtained in [10].

In the experiments, 1D map was represented by the logistic map in chaotic mode with parameter μ = 7,71 (Lyapunov exponent λ = 0,349) and coupling coefficients between the ensemble elements α = 0,157.

After the start of collective iterations, in several steps the ensemble converges to stable synchronous mode state, in which the map variables oscillate chaotically but synchronously with each other.

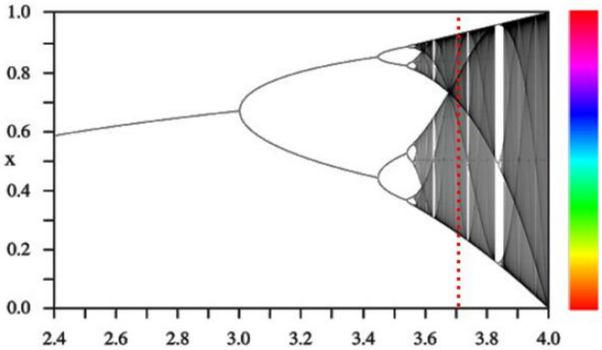

Fig. 3. Bifurcation diagram for the logistic map and color mapping to HSV color model (fixed saturation and value).

5.4. Break of Collective Behavior

When one or several links of the map is broken, the node algorithm reacts as follows. If on time interval T comes no information from at least one of the other n–1 nodes, data on the other node states is discarded and the node transfers to autonomous operation mode. In some time, this leads to desynchronization, first, of chaotic states of the nodes, and then, in some other time, to mistiming of the node clocks. The latter can be observed visually as asynchronous alternation of LED colors.

5.5. Reinstatement of Collective Iterations

Even after the collective behavior is broken, node algorithm constantly tracks the presence of other network nodes. If it finds them, and as soon as all the nodes from the list are present in the network, the algorithm first restores node clock synchronization, and then the process of collective iterations. As a result, synchronous collective iterations are self-restored.

6. The Experiment

The sensor network that was used in the experiment consisted of 4 nodes (fig. 4). Each node contained a direct chaotic transceiver and an actuator board, on which logistic map was iterated in accordance with the above algorithm.

The devices were arranged on a table and turned on by operator. At startup, the initial value for each dynamic system is selected using random-number generator. According to the above rules, each device listens for time interval *T* and then transmits a packet with node ID number and state variable value information. After the listening, the step is repeated. It took some time to synchronize the node clocks after turning on the last device. Clock synchronization was visualized by simultaneous actuator color changes. For some time (4-5 iterations), LED colors changed chaotically and remained different on each actuator. After that, the colors were still chaotically changing with time, but the difference of node colors became visually smaller and smaller, and the collective behavior of the ensemble led to synchronous state. This state is stable and it remains until some natural or artificial break.

As an example of such a “break,” one device was turned off during the experiment. The remaining nodes began operating autonomously, without using other node state information. Three remaining powered devices worked as follows: for some time, it looked as if they were synchronous, but after several iterations the difference in the state values increased and LED colors became visually different and finally came to completely uncorrelated behavior. However, time synchronization between the devices remained, therefore moments of color changes were still simultaneous.

After the node clocks lost synchronization (mistiming), the 4th device was turned on again. Clock synchronization process repeated itself. It wasn’t visually apparent because by this time the clock discrepancy in quartz generators was not large enough. Full clock synchronization was restored after time interval of about *T*. Then the process of restoring the synchronous state of the ensemble began. It was indicated by color convergence.

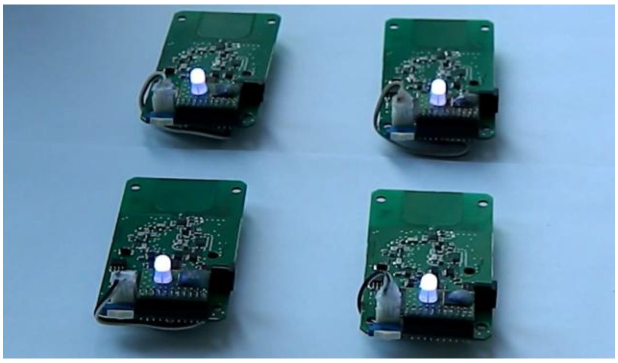

Fig 4. Synchronous mode in the experiment**.**

7. Conclusions

In this report, wireless active networks are used as a technique for investigation of collective behavior of dynamic systems. As an example, this approach is applied in experiment to an ensemble of 1D maps, whose equations are iterated on the nodes using data on the states of the other ensemble elements, that is transmitted via ultra wideband radio channel.

References

[1] Puccinelli, D., Haenggi, M. "Wireless sensor networks: applications and challenges of ubiquitous sensing," Circuits and Systems Magazine, IEEE , vol. 5, no. 3, pp. 19-31, 2005.

[2] Special Issue on Wireless Sensor and Actuator Networks. IEEE Trans. on automatic control, 2011, vol. 57, No 10.

[3] L. Chua and L. Yang. “Cellular Neural Networks: Theory”. IEEE Trans. Circuits and Systems, 1988. vol. 35, pp. 1257-1272.

[4] L. Chua and L. Yang. “Cellular neuralNetworks: Applications”. IEEE Trans. Circuits and Systems, 1988, vol. 35, pp. 1273-1290.

[5] T. Roska, L. Chua. “The CNN Universal Machine an Analog Array Computer”. IEEE Trans. on Circuits and Systems. 1993, vol. 40, pp. 163-173.

[6] M. Haenggi. “Distributed sensor networks: a cellular nonlinear networks perspective”. Intern. Journal of Neural Systems, 2003, vol. 13, No 6, pp. 405-414.

[7] Yu. Andreyev, A. Dmitriev, E. Efremova and V. Lazarev. “Ultra Wideband Transceivers Based on Chaotic Pulses and Their Applications to Wireless Body Area Networks” 2013 Proc. International Symposium on Nonlinear Theory and its Applications (NOLTA2013), Santa Fe, USA, pp. 221-224, September 2013.

[8] S. Barbarossa. "Self-organizing sensor networks with information propagation based on mutual coupling of dynamic systems", IWWAN 2005, 2005.

[9] R. Pagliari, A.Scaglione. “Scalable Network synchronization with Pulse-Coupled Oscillators”. IEEE Trans. On Mobile on Mobile Computing, 2011, vol. 10, No. 3, pp. 392-405.

[10] A. Dmitriev, M. Shirokov, S. Starkov. “Chaotic Synchronization in Ensembles of Coupled Maps”, IEEE Transactions on Circuits and Systems, 1997, vol. 44, No. 10, pp. 918-926.