2009 International Symposium on Nonlinear Theory and its Applications
NOLTA'09, Sapporo, Japan, October 18-21, 2009

Chaotic Synchronization and De-synchronization for a Token-based Computational Architecture

Hiroyasu Ando*[1], Ferdinand Peper*[2], and Kazuyuki Aihara*[3]

[1] RIKEN Brain Science Institute, Saitama, 351–0198 Japan
[2] National Institute of Information and Communications Technology, Nano ICT Group, Kobe, 651–2492 Japan
[3] Institute of Industrial Science, the University of Tokyo, Tokyo, 153–8505 Japan
Email: hiroyasu_ando@brain.riken.jp

Abstract—Synchronization in chaotic systems has been of much importance in the field of nonlinear dynamics, not only for mathematical aspects but also for their applications such as computation, etc. In this paper, we apply chaotic synchronization and de-synchronization to a token-based computational architecture in which the tokens are driven by chaotic fluctuations. It is expected (though not strictly proved here) that the resulting scheme is computationally universal, since a similar scheme, driven by stochastic fluctuations, also is. Simulations confirm that the chaotic scheme behaves in similar ways as the equivalent token-based stochastic scheme.

1. Introduction

Chaotically fluctuating orbits sensitively depend on initial conditions and thus it seems difficult to control two identical chaotic systems such that they become synchronized. However, it has been shown that chaotic dynamical systems are controllable due to their inherent determinism, and that, moreover, two or more chaotic elements are synchronizable (reviewed in [1]). Since chaotic dynamics provides a rich dynamical behavior, synchronization phenomena in chaotic systems could be applied to computation [2] for instance.

The concept to exploit stochastic fluctuation has been proposed recently in applications ranging from computation [3] to secure communication [4]. Among these stochasticity-based computing schemes, a token-based circuit driven by stochastic noise, called Brownian circuit, has been proposed that is computationally universal [5, 6]. The Brownian circuits are searched by noise-driven tokens, whereas computational operations are achieved as the outcome of this search process.

Since thermal noise can be used as a (freely available) noise source, Brownian circuits have an interesting potential for implementations at nanometer scales [3]. There is a catch, though, and that is that stochastic search is less efficient than conventional schemes, even though some local optimization of search efficiency can be accomplished through the placement of *ratchets*, which are diode-like devices that restrict search to one direction. Another way to increase efficiency of search is to take control of the fluctuations, in a way that preserves a token's ability to explore the state space inside a circuitry, while decreasing the number of trials that fail to carry the computation forward. The use of chaotic fluctuations is a promising avenue in this context, given that chaotic search has been shown to be more efficient than stochastic search for some optimization problems [7] and for solving a maze [8].

In this paper, we consider the above mentioned token-based circuits in terms of deterministic chaotic dynamics. Towards investigation of the efficiency of the circuits by chaos, as a first step, we show the possibility to implement the token-based circuits by chaotic dynamics, particularly by chaotic synchronization.

2. Brownian circuits

Brownian circuits are circuits in which signals are encoded by tokens that are allowed to fluctuate on wires [5, 6]. Like in traditional logic circuits, there is a set of primitive modules for Brownian circuits from which any arbitrary circuit can be constructed. Called *Brownian Circuit Primitives* (BCPs), this set is quite simple, as compared to conventional token-based circuits. This simplicity is due to the power inherent in the stochastic search process that tokens (inadvertently) conduct.

Fig. 1(a) shows the *Hub* primitive, which is nothing more than three wires knot together at one point. The Hub can contain at most one token at a time on its wires, and this token is allowed to fluctuate between the three wires. Fig. 1(b) shows the *CJoin* primitive, which is equipped with switching ability: a token on a wire of a Cjoin can only be switched when another token on the opposite wire is present, and as the result of a switching event both tokens are transferred to the CJoin's other pair of (orthogonal) wires. The tokens are allowed to fluctuate, like with tokens on the wires of a Hub, and the CJoin can switch forward and backward. Fig. 1(c) shows an example of a simple Brownian circuit, which is analyzed in more detail in

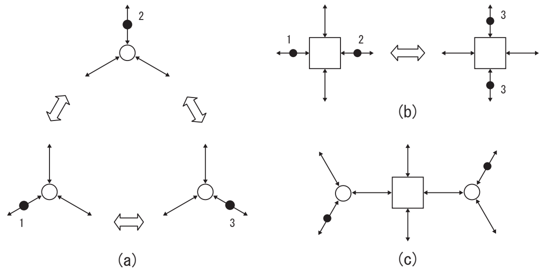


Figure 1: (a) Hub primitive, and (b) CJoin primitive and their possible transitions. The labels 1, 2, and 3 on the wires denote states described in detail in Section 3. (c) Brownian circuit consisting of one CJoin connected to two Hubs.

this paper.

3. Root Module by dynamical systems

To simplify the model of the Brownian circuits primitives, we unify their representation. That is, the Hub and the CJoin are represented by a single Root module, the behavior of which is defined by a nonlinear dynamical system. The Root module has 4 wires. These wires may contain tokens in certain combinations, and these combinations encode the presence or absence of tokens on certain wires of a Hub or a CJoin. The wires of a Root module are labeled by the numbers $n = 0, 1, 2, 3$. Each wire is represented by two uncoupled oscillators, called the x-oscillator and the y-oscillator, which are denoted by $x^{(n)}$ and $y^{(n)}$, whereby n is the label of the wire. These oscillators play an important role in the representation of tokens. Depending on whether two oscillators are synchronized or de-synchronized, there will be no or one token, respectively, on the corresponding wire. How are the oscillators implemented? We use an autonomous continuous-time chaotic system, and to this end we need at least 3-dimensions per oscillator because of the Poincaré-Bendixson theorem. We associate wire n with m-dimensional oscillator $X^{(n)}$, with $m \geq 2 \times 3 = 6$. The x- and y-oscillator are extracted from $X^{(n)}$ through the projection operators $\mathbf{P}_x, \mathbf{P}_y$.

Since the Root module has four wires, we use four pairs of uncoupled 3-dimensional nonlinear dynamical systems to represent it. Each pair is driven by the rest of the pairs through coupling constants. The dynamics is described by the following equation.

$$\dot{X}^{(n)} = \mathbf{F}(X^{(n)}, \mathbf{a}) + \mathbf{g}(\mathbf{P}_x X^{(m)}, \mathbf{P}_y X^{(m)}, \alpha_{(n,m)}), \quad (1)$$

where $X^{(n)}$, $n = 0, 1, 2, 3$, are vectors, and $X_i^{(n)}, i = 0, 1, \ldots, 5$, are the elements. The function $\mathbf{F}$ governs the basic dynamics of the uncoupled system. In this paper, the basic dynamics follows the Lorentz system: $\dot{X}_{0,3} = -\sigma(X_{0,3} - X_{1,4})$, $\dot{X}_{1,4} = -X_{0,3}(X_{2,5} - r) - X_{1,4}$, $\dot{X}_{2,5} = X_{0,3}X_{1,4} - bX_{2,5}$ with the parameters $\sigma = 10, r = 8/3, b = 28$. The function $\mathbf{g}$ is a coupling function with the coupling constant $\alpha_{(n,m)}$, $m = 0, 1, 2, 3$, which is from the wire m to n. In particular, if $i = 1, 4$, then $\mathbf{g}_i(x^{(m)}, y^{(m)}, \alpha_{(n,m)}) = \sum_m \alpha_{(n,m)} |x^{(m)} - y^{(m)}|$. Otherwise $\mathbf{g}_i(x^{(m)}, y^{(m)}, \alpha_{(n,m)}) = 0$. P_x and P_y are projections on X_0 and X_3, respectively.

Next, we explain how to encode the Hub and the CJoin by the Root module (RM). The encodings of the token states in the Hub and the CJoin are summarized in Table 1, together with the coupling constants conditions of the RM, which are shown at the right side of the table. The state of a Hub or a CJoin thus corresponds with a set of coupling constants satisfying a certain condition. Changing a condition for a set of coupling constants results in a change in which oscillators in a Root module are synchronized or de-synchronized, and this in turn effects a change in a Hub or a CJoin.

Entry no. 1 in Table 1 refers to the absence of tokens on any wire of a Hub or a CJoin. This is encoded as the presence of a token on the wire no. 0 of the RM, so the two oscillators $x^{(0)}$ and $y^{(0)}$ will be de-synchronized and the other three pairs of oscillators will be synchronized.

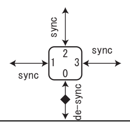
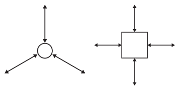
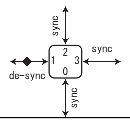
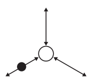
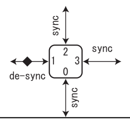
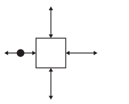
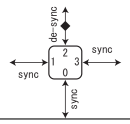
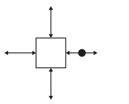
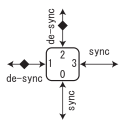
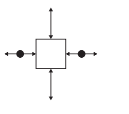
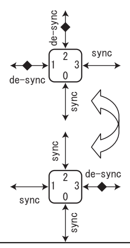
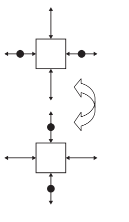
Entry no. 2 in Table 1 refers to the presence of one token on one of the wires of a Hub. This corresponds to the de-synchronization of the oscillator pair associated with one of the three wires $n = 1, 2, 3$ in the Root module, whereas the other pairs of oscillators will be synchronized.

Entries no. 3 and 4 in the Table refer to the presence of one token on the left, resp., right input wire of a CJoin, and they follow the same philosophy of synchronization and de-synchronization. Once a second token is input to a CJoin, we enter in a state described by entry no. 5 of Table 1. This is a transitional state, which is entered only temporarily, before a CJoin enters in the state described by entry no. 6. The latter state allows the tokens at the CJoin to fluctuate pair-wise between the input-side of the CJoin and its output side.

4. Connecting the root modules

A Brownian circuit is constructed by connecting BCPs to each other according to a desired circuit topology. This requires that dynamical systems corresponding to individual RMs be combined into a larger, more global, dynamic system. A wire is divided in two *branches*, one branch for each of the two modules at both ends of the wire. To smoothly connect two branches, their states need to be made compatible. Two branches connected to each other can contain at most one token at a time (*multi-token prohibition*), be-

Table 1: Encoding of the Hub and the CJoin by the Root module (RM). The conditions governing parameter $\alpha_{(n,m)}$ are shown in the right-most columun.

No.	RM	BCPs	Description
1			**No token**: $\alpha_{(0,1)} \ll \alpha_{(1,0)}$, $\alpha_{(0,3)} \ll \alpha_{(3,0)}$, $\alpha_{(0,2)} \ll \alpha_{(2,0)}$.
2			**1 token**: At any wires in Hub, $\alpha_{(1,0)} < \alpha_{(0,1)}$.
3			**1 token**: At Input wire-1 in CJoin, $\alpha_{(1,3)} \ll \alpha_{(3,1)}$, $\alpha_{(1,0)} \ll \alpha_{(0,1)}$, $\alpha_{(1,2)} \ll \alpha_{(2,1)}$.
4			**1 token**: At Input wire-2 in CJoin, $\alpha_{(3,1)} \ll \alpha_{(1,3)}$, $\alpha_{(2,1)} \ll \alpha_{(1,2)}$, $\alpha_{(2,3)} \ll \alpha_{(3,2)}$.
5			**2 Tokens**: At Input wires in CJoin, $\alpha_{(1,3)} \ll \alpha_{(3,1)}$, $\alpha_{(1,0)} \ll \alpha_{(0,1)}$, $\alpha_{(2,0)} \ll \alpha_{(0,2)}$, $\alpha_{(2,3)} \ll \alpha_{(3,2)}$, $\alpha_{(1,2)} = \alpha_{(2,1)} = 0$,.
6			**2 Tokens**: At Input or Output-wires in CJoin, $\alpha_{(3,1)} < \alpha_{(1,3)}$, $\alpha_{(1,0)} \ll \alpha_{(0,1)}$, $\alpha_{(2,0)} \ll \alpha_{(0,2)}$, $\alpha_{(3,2)} < \alpha_{(2,3)}$, $\alpha_{(1,2)} = \alpha_{(2,1)} = 0$,.

cause there can be only one token on a wire at a time. The implementation of the multi-token prohibition requires a closer look at the synchronizing behavior of pairs of oscillators. Given that a token is represented as the de-synchronization of two oscillators, the dynamics of these oscillators needs to be used to synchronize the other pairs of oscillators that represent branches at which no tokens are allowed to appear due to the multi-token prohibition. It has been shown that uncoupled Lorenz systems can be synchronized by common white Gaussian noise [9], and by interpreting our signals from the de-synchronized oscillators as a common noise source, we can use a similar mechanism to synchronize other pairs of oscillators.

The multi-token prohibition not only applies to two branches connected to each other, but also to branches further away. In the case of Fig. 1(c) the left Hub contains a token on one of its branches, and even if the branch containing the token is not connected to the CJoin, there can be no token on the CJoin's branch connected to the Hub. The reason is that a token on a wire of the Hub can eventually move to the branch of the CJoin, and this would violate the multi-token prohibition.

To connect the RMs into Brownian circuits such that the multi-token prohibition is satisfied, it is necessary to switch adaptively between the coding states of the module. This is implemented by changing the coupling constants according to the conditions in the right column of Table 1. To add this adaptive functionality to the RMs, we use so-called *Angel operators*, which are functions that observe branches that should satisfy the multi-token prohibition. The Angel operators conduct their tasks on a local scale, such as not to inadvertently pick up information on the behavior of tokens at more remote locations.

The multi-token prohibition is implemented by the following equation.

$$\alpha_{(n,m)} \quad = \quad \beta_{(n,m)} + \mathbf{\Delta}_{(n,m)} \cdot \mathbf{G}(\mathbf{Y}) \qquad (2)$$

where $\mathbf{Y} \in \{0,1\}^{N_B}$ is a token indicator vector determined by Angel operator, N_B is the number of branches in a BCP. The element of $\mathbf{Y}$, denoted by Y_j, represents the information from the branch-j of a Hub ($j = 1,2,3$) or a CJoin ($j = 1,2,3,4$). If there is a token in the observed (connected) branch, Y_j goes to 1, otherwise $Y_j = 0$. The constants $\beta_{(n,m)}$ represent default states of RMs. The $\mathbf{\Delta}_{(n,m)} \in \mathbf{R}^M$ is a constant vector that determines a kind of learning rate by which the variables α can change in a certain time interval. This constant vector is set such that the time scale of the dynamics of equation (2) is slower than the dynamics $X^{(n)}$ of the RMs. Failing to set this constant vector appropriately would cause the dynamics to become unstable, due to inadvertent feed-back effects. The set-theoretical (Boolean) function $\mathbf{G} : \mathbf{R}^{N_B} \to \mathbf{R}^M$ is explained through an example in the next section. The second term in the RHS of equation (2) is an inner product of vectors.

5. Simulations

To exemplify the coding of Brownian circuits by the connected Root modules, we simulate the circuit in Fig. 1(c). First, we define equations that indicate the presence of tokens. For the left and right Hub, respectively, we use the variables Y_1 and Y_2 as token indicators. $Y_1 = f(|X_0^{(1)} - X_3^{(1)}| + |X_0^{(3)} - X_3^{(3)}|)$, $Y_2 = f(|X_0^{(2)} - X_3^{(2)}| + |X_0^{(3)} - X_3^{(3)}|)$, The function $f(\cdot)$ is a sigmoidal function for which $f(x,a) = (1 + \exp(-1000(x-a)))^{-1}$, whereby a is a threshold parameter. So, when $Y_1 = 1$ there is a token in one of the branches of the left Hub. The token indicators are used as input to the Angel operators. An

Angel operator at the point connecting a Hub to the CJoin observes not only tokens on the input wires of the CJoin, but also tokens on the output wires of the CJoin, because tokens on these output wires can move back through the CJoin to its input wires, and from there move back to the Hub.

The indicators for the left resp. right side of the CJoin are Y_1 and Y_2, which are defined as $Y_j = f(|X_0^{(1)_j} - X_3^{(1)_j}| + |X_0^{(2)_j} - X_3^{(2)_j}| + |X_0^{(3)_j} - X_3^{(3)_j}|)$, An angel operator connecting the CJoin to a Hub observes all the branches of the Hub, because a token on any of them can move to the CJoin on a short notice.

The function $\mathbf{G}$ is basically composed of zero and identity components for the Hub, but $\mathbf{G}$ can also be a set theoretical operator of Y_j for the CJoin. This is because a CJoin can assume many token states such as the ones in entries no. 0, 3, 4, 5, 6 in Table 1. The many state transitions that are possible require not only simple binary values of $Y_j = 0, 1$, but also restrictions on combinations of different Y_j's. These are defined by two additional functions. $G_1(Y_1, Y_2) = \sqrt{Y_1 \cdot Y_2}$, $G_2(Y_1, Y_2) = f_2(Y_1 + Y_2, a - N + 2)$, where $f_2(x, y, a) = 1 - f(x + y, a - N + 2)$ with $N = 0, 1, 2$ depending on the number of tokens at the CJoin and the two Hubs.

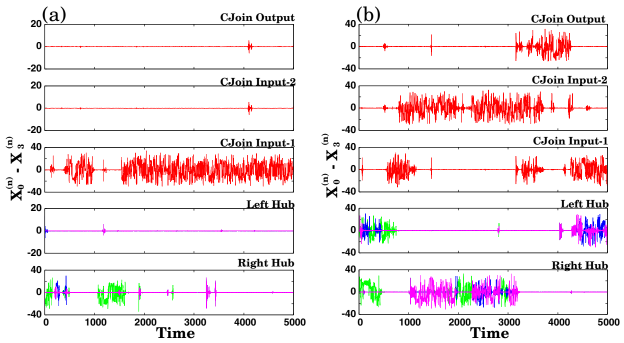


Figure 2: Token representation by de-synchronization in the Root modules for the Brownian circuit in Fig. 1(c). (a) Case with one token in the left side of the circuit. (b) Case with two tokens in both the left and right sides of the circuits. In both cases the panels in the graphs have the following meanings (from top to bottom): Output, Input-2, Input-1, Left Hub, Right Hub. For the Hubs the synchronization states are superimposed.

The default relations for the coupling constants of the two Hubs are: $\beta_{(1,0)} < \beta_{(0,1)}$, $\beta_{(0,3)} = \beta_{(3,0)}$, $\beta_{(0,2)} = \beta_{(2,0)}$. The β's for other pairs (n, m) and (m, n) are the same. The default relations for the coupling constants of the CJoin are: $\beta_{(1,0)} < \beta_{(0,1)}$, $\beta_{(1,2)} = \beta_{(2,1)}$, $\beta_{(2,3)} \ll \beta_{(3,2)}$, $\beta_{(0,3)} = \beta_{(3,0)}$, $\beta_{(2,0)} < \beta_{(0,2)}$, $\beta_{(1,3)} \ll \beta_{(3,1)}$.

In Fig. 2, we show the token moving in the circuit in Fig. 1(c) represented by de-synchronizatoin/synchronization in the wires of the Root modules. The vertical axes represent $X_0^{(n)} - X_3^{(n)}$ in the wires for the two Hubs and the Cjoin. So, a non-zero value means de-synchronization and thus the presence of a token. In Fig 2(a), where there is one token in the left side of the circuit, no significant de-synchronization occurs in the Input-2 and the Output branches of Cjoin and the right Hub. A token is moving between the left Hub and the Input-1 branch of the CJoin. For (b), there are two tokens in both sides of the circuit just like in Fig 1(c). Token transitions in the circuits can be observed through the de-synchronizations of the wires. Note that the token transitions in the figures follow the multi-token prohibition.

6. Discussion and Conclusion

Brownian circuits have been implemented by chaotic synchronization, whereby the presence and movement of tokens are represented by de-synchronized chaotic oscillators. Since the fluctuation used this model follows deterministic rule, we obtain increased control over the dynamics, thus potentially allowing faster system performance. Construction of larger circuits by the proposed scheme will require extension of the multi-token prohibition rule and the Angel operator, such that locality of relations is preserved as much as possible, because that will allow the connection of modules to each other without considerations of the dynamics inside each module—a key requirement for the modular construction of circuits.

This work is partly supported by Aihara Complexity Modeling Project, ERATO, JST.

References

[1] S. Boccaletti *et. al. Phys. Rep.*, vol. 329, pp. 103–197, 2000, *ibid*, vol. 366, pp. 1–101, 2002.

[2] K. Murali and S. Sinha, *Pys. Rev. E*, vol. 75, p. 025201, 2007.

[3] F. Peper, "Exploiting Noise in Computation" in Proc. NOLTA 09.

[4] L. B. Kish, *et. al.*, *SPIE Newsroom*, 2007.

[5] F. Peper, J. Lee, and *et. al.*, "Brownian circuits.Part I: Concept and basic designs" *In preparation*, 2008.

[6] J. Lee, F. Peper, and *et. al.*, "Brownian Circuits . Part II: Efficient Designs and Brownian Cellular Automata" *In preparation*, 2008.

[7] L. Chen and K. Aihara, *Neural Network* vol.8, pp. 915–930, 1995.

[8] Y. Suemitsu and S. Nara, *Neural Comp.* vol.16, pp. 1943–1957, 2004.

[9] C. S. Zhou and J. Kurths, *Phys. Rev. Lett.* vol. 88, p. 230602, 2002.