# A NOVEL ADAPTIVE CONTROL VIA SIMPLE RULE(S) USING CHAOTIC DYNAMICS IN A RECURRENT NEURAL NETWORK MODEL AND ITS HARDWARE IMPLEMENTATION

Ryosuke Yoshinaka[1], Masato Kawashima[1], Yuta Takamura[1], Hitoshi Yamaguchi[1]
Naoya Miyahara[1], Kei-ichiro Nabeta[1], Yongtao Li[2] and Shigetoshi Nara[1]
[1]*Graduate School of Natural Science and Technology, Okayama University*
*3-1-1 Tsushima-naka, Kita-ku, Okayama 700-8530, Japan*
[2]*Research Institute for Electronic Science, Hokkaido University, Kita 12-jyo Nishi 7-chome, Kita-ku, Sapporo, Japan*

Keywords: Dynamics, Adaptive control, Recurrent neural network, Hardware implementation, Autonoumous robot, Neuromorphic device, Brainmorphic device.

Abstract: A novel idea of adaptive control via simple rule(s) using chaotic dynamics in a recurrent neural network model is proposed. Since chaos in brain was discovered, an important question, what is the functional role of chaos in brain, has been arising. Standing on a functional viewpoint of chaos, the authors have been proposing that chaos has complex functional potentialities and have been showing computer experiments to solve many kinds of "ill-posed problems", such as memory search and so on. The key idea is to harness the onset of complex nonlinear dynamics in dynamical systems. More specifically, attractor dynamics and chaotic dynamics in a recurrent neural network model are introduced via changing a system parameter, "connectivity", and adaptive switching between attractor regime and chaotic regime depending surrounding situations is applied to realizing complex functions via simple rule(s). In this report, we will show (1)Global outline of our idea, (2)Several computer experiments to solve 2-dimensional maze by an autonomous robot having a neural network, where the robot can recognize only rough directions of target with uncertainty and the robot has no pre-knowledge about the configuration of obstacles (ill-posed setting), (3)Hardware implementations of the computer experiments using two-wheel or two-legs robots driven by our neuro chaos simulator. Successful results are shown not only in computer experiments but also in practical experiments, (4)Making pseudo-neuron device using semiconductor and opto-electronic technologies, where the device is called "dynamic self-electro optical effect devices (DSEED)". They could be "neuromorphic devices" or even "brainmorphic devices".

## 1 INTRODUCTION

Since chaos is discovered in many natural phenomena, particularly, chaotic dynamics observed in biological systems including brain suggest us to consider whether there are certain important relations between chaotic dynamics and their excellent functions in both information processing and well-regulated functioning or controlling. (Skarda and Freeman, 1987)(Tsuda, 2001)(Fujii et al., 1996)(Tokuda et al., 1997).

On the other hand, the rapid progress of robotics have brought various robots into our life and industry, however, it is still quite difficult for the robots to perform tasks adaptively in various environments, whereas biological systems have excellent functions in both information processing and well-regulated functioning in various environments. Conventional methodologies (decomposing a system into parts and elements) often fall into difficulties to face enormous complexity originating from dynamics in systems with large but finite degrees of freedom.

Under these situations, biological information and control processing has suggested that they could work under novel *dynamical mechanism* that cause excellent functions in information processing and/or controlling. Therefore, many dynamical models have been proposed for approaching the mechanisms by means of large-scale simulation or heuristic methods. Our work originates from a novel idea to harness the onset of complex nonlinear dynamics in information processing or control systems, and mainly study chaotic dynamics in neural networks from the functional viewpoint. First, Nara & Davis intro-

Yoshinaka R., Kawashima M., Takamura Y., Yamaguchi H., Miyahara N., Nabeta K., Li Y. and Nara S..
A NOVEL ADAPTIVE CONTROL VIA SIMPLE RULE(S) USING CHAOTIC DYNAMICS IN A RECURRENT NEURAL NETWORK MODEL AND ITS HARDWARE IMPLEMENTATION .
DOI: 10.5220/0003058301450155
In *Proceedings of the International Conference on Fuzzy Computation and 2nd International Conference on Neural Computation* (ICNC-2010), pages 145-155
ISBN: 978-989-8425-32-4

duced chaotic dynamics into a recurrent neural network model (RNNM) by adjusting only one system parameter (connectivity among neurons), and they proposed that constrained chaos could be potentially useful dynamics to solve complex problem, such as ill-posed problems (Nara and Davis, 1992). As one of functional experiments, chaotic dynamics was applied to solving a memory search task or image synthesis which is set in an ill-posed context (Nara et al., 1993)(Nara et al., 1995)(Nara, 2003). Furthermore, the idea is extended to challenging application of chaotic dynamics to control. Chaotic dynamics introduced in a recurrent neural network model was applied to control tasks that an object is requested to solve a two-dimensional maze for catching a target (Suemitsu and Nara, 2004), or to capture a target moving along different trajectories (Li and Nara, 2008). From the results of computer experiments, we consider that complex dynamics/chaotic dynamics could be useful not only in solving ill-posed problems but also in controlling of systems with large but finite degrees of freedom.

Therefore, in the present paper, we develope our idea and propose a quasi-layered RNNM consisting of sensing neurons(upper layer) and driving neurons(lower layer). In both layers, chaotic dynamics are used. This idea is based on the work of Mikami and Nara who found that chaos has a sensitive response property to external input (Mikami and Nara, 2003). Their idea is applied to practical functional experiments to solve 2-dimensional mazes, as shown in the later sections. We can find a corresponding example in biological behaviors. For instance, auditory behavior of cricket gives a typical ill-posed problem in biological systems (Huber and Thorson, 1985). Further developments are shown about the following topics. They are:

**(a)** to apply our idea to a roving robot with two legs;

**(b)** to apply our idea to an arm robot;

**(c)** to propose a hardware device of pseudo-neuron and a network of them, and to evaluate them by computer experiments;

**(c)** to make an actual hardware device using semiconductor and opto-electronic technologies.

## 2 CONTROL SYSTEM

### 2.1 Construction of Control System

The control system mainly consists of a roving robot with a micro processor unit (MPU), sensing systems of sound signal from target and of detecting obstacles, a neural chaos simulator, Bluetooth interface between the robot and the neural chaos simulator, and a target emitting a specified sound signal, which is like a singing male cricket, shown as Fig.1. The

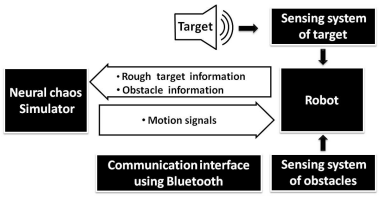



Figure 1: Block diagram of control system.

robot with sensors is shown as Fig.2. It has two driving wheels and one castor (2DW1C). The robot has six sensors which can be divided into two parts. One is the sensing system of detecting obstacles that consists of two ultrasonic sensors which give the robot the ability to detect whether an obstacle does exist in front of the robot without actually touching it. The other is the sensing system of sound signal from target that consists of four sets of directional microphone circuits, which functions as *ears* of the robot. Four microphones are set with directing to the front, the back, the left and the right of the robot, which is shown as Fig.2(right). In our study, a loud speaker is employed as the target, and is emitting 3.6KHz sound signal like a singing cricket. This sound signal is picked up by these four *ears*(microphone) with $\pi/2$ detecting angle oriented to four directions. Among them, a sound signal coming from one side might have a strongest intensity, or be loudest. These four sound signals are amplified, rectified , digitalized, and transferred to MPU, respectively. At the preliminary stage, these signals are used to compare which direction is the strongest one. In near future, we will try to input them into sensing neurons (upper layer) in quasi-layered RNNM, but in the present study, we must emphasize the two points. One is that, the sensing system of sound signal from target does not give accurate directional information of target, but rough directional information of target with uncertainty. The other is that, these signals inputted from four microphones are not processed with complex techniques or methods. These are quite important differences between our work and other conventional robotics.

Now, the problem is how these sensing signals are sent to the neural chaos simulator, which works as the neural network system to make adaptive behaviors of the robot. In our study, a computer with good performance is chosen as the neural chaos simulator, which is programmed in C language and works in Vine Linux. A wireless communication interface

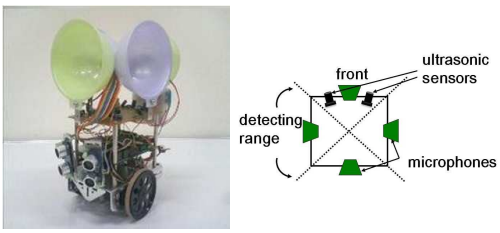



Figure 2: The roving robot with sensors including two ultrasonic sensors and four microphones: A photo picture (left) and a sketch map (right).

between the MPU of the robot and the neural chaos simulator has been built using a pair of Bluetooth serial adapter and a group of communication protocol. After the sensing signals was sent to the neural chaos simulator, the simulator performs neural information processing, produces adaptive motion signals, and sends them back to the robot again. And then, the robot moves one step. At a new position, sensing information from sensors are checked again. The above process is repeated. Therefore, this is a close-loop control system. After the sensing systems and communication interface have worked well, it is the key point whether the neural simulator enables the robot to produce adaptive motions in various environment, such as mazes or moving targets. In next section, we will introduce it in detail.

## 2.2 Context Setting of Solving Mazes

In the present study, the context setting of solving mazes is shown as follows.

1) *Set a typical ill-posed problem of solving two-dimensional mazes.*

2) *Set obstacles unknown by the robot.*

3) *Set a target emitting a sound signal.*

4) *Acquire information for reaching the target.*

• Check whether obstacles to prevent the robot from forward moving exist or not, by ultrasonic sensors for detection.

• Obtain rough direction of the target by four microphones.

5) *Calculate the motion increments at every time step of updating neural network activity.*

# 3 NEURAL CHAOS SIMULATOR

The neural chaos simulator is utilized to simulate dynamical activities of neural network, and works like the ”brain” of the robot. In various unknown environment or mazes, chaotic dynamics generated in the neural network makes the robot generate complex motions to adapt environment or avoid obstacles. In our study, we start from a simple RNNM, and develop it to a quasi-layered RNNM so as to approach the mechanism of brain nearer.

## 3.1 Recurrent Neural Network Model

Our study works with a fully interconnected RNNM consisting of $N$ binary neurons, and the updating rule is defined by

$$S_i(t+1) = \mathrm{sgn}\Big( \sum_{j \in G_i(r)} W_{ij} S_j(t) \Big) \qquad (1)$$

$$\mathrm{sgn}(u) = \begin{cases} +1 & u \geq 0; \\ -1 & u < 0. \end{cases}$$

- $S_i(t) = \pm 1 (i = 1 \sim N)$ : the firing state of a neuron specified by index $i$ at time $t$.
- $W_{ij}$ : connection weight from the neuron $S_j$ to the neuron $S_i$, $W_{ii}$ is taken to be 0.
- $r (0 < r < N)$: fan-in number for neuron $S_i$, named connectivity
- $G_i(r)$: spatial configuration set of connectivity $r$ for neuron $S_i$

At a certain time $t$, the state of neurons in the network can be represented as a $N$-dimensional state vector $\boldsymbol{S}(t)$, called as *state pattern*. The updating rule shows that time development of state pattern $\boldsymbol{S}(t)$ depends on the connection weight matrix $W_{ij}$ and connectivity $r$. Therefore, when full connectivity $r = N-1$, by means of a kind of orthogonalized learning method (Nara et al., 1995) , appropriately determining $W_{ij}$ could embed a group of $N$ dimensional state pattern as cyclic memory attractors in $N$ dimensional state space. Attractor patterns consists of ($K$ patterns per cycle)$\times$ $L$ cycles, and each patterns has $N$ neurons. For example, in Fig.3, we take $K = 6$, $L = 4$,and $N = 400$. In this case, the firing states of $N = 20 \times 20 = 400$ neurons are represented by black pixel or white pixel. As the network evolves with the updating rule for enough time steps, randomly initial state pattern will converge into one of embedded cyclic attractors. Now we

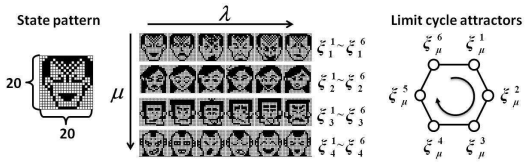



Figure 3: One example of embedded attractor patterns: when connectivity $r = N-1$, if $S(t)$ is $\xi_1^1$, then the output sequence is $\xi_1^2$, $\xi_1^3$ ,..., $\xi_1^6$, $\xi_1^1$,....

reduce connectivity $r$ by blocking signal transfer from other neurons, attractors gradually becomes unstable, and the network state changes from attractor dynamics to chaotic dynamics. In order to analyze the destabilizing process, we have calculated a bifurcation diagram of overlap (Fig.4) where overlap means one-dimensional projection of state pattern $S(t)$ to a certain reference pattern. Therefore, an overlap $m(t)$ is defined by

$$m(t) = \frac{1}{N} S(0) \cdot S(t) \qquad (2)$$

$$t = Kp + t_0 \qquad (p = 1, 2, \ldots) \qquad (3)$$

where $S(0)$ is an initial pattern(reference pattern) and $S(t)$ is the state pattern at time step $t$. Because $m(t)$ is a normalized inner product, $-1 \leq m(t) \leq 1$. $m(t) = 1$ means that the present state pattern and the reference pattern is same. In other words, the reference pattern repeatedly appears every K=6 steps. In our study, for the upper layer and the lower layer, we have calculated the overlap $m(t)$ with respect state pattern $S(t)$ when it evolves for long time.

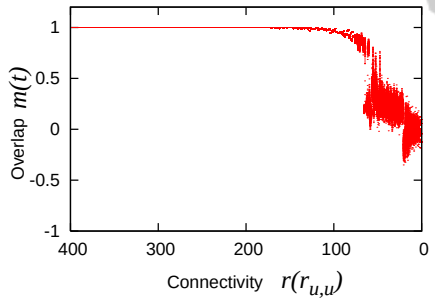



Figure 4: The long-time behaviours of overlap $m(t)$ at $K$-step mappings. The horizontal axis is $r(r_{u,u})$ (0-399).

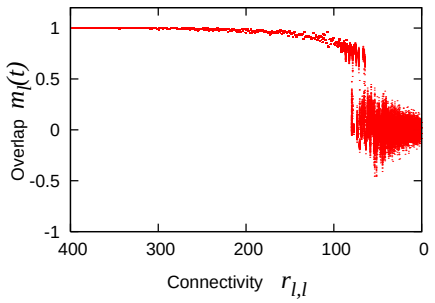



Figure 5: The long-time behaviors of overlap $m_l(t)$ at $K$-step mappings ($r_{u,l} = 0$). The horizontal axis is $r_{l,l}$ (0-399).

## 3.2 Quasi-layered Recurrent Neural Network Model

A quasi-layered RNNM consists of a upper layer with $N$ neurons and a lower layer with $N$ neurons. The upper layer is updated by only self-recurrence. On the other hand, the lower layer are updated not only by self-recurrence but also by recurrent outputs of the upper layer. State pattern $S(t) = [x(t), y(t)]$, where $x(t) = \{x_i(t) = \pm 1 \mid i = 1, 2, \cdots, N\}$ and $y(t) = \{y_i(t) = \pm 1 \mid i = 1, 2, \cdots, N\}$. The updating rules of two layers are defined by

$$x_i(t+1) = \mathrm{sgn}\left( \sum_{j \in G_u(r_u)} W_{ij}^{u-u} x_j(t) \right) \qquad (4)$$

$$y_i(t+1) = \mathrm{sgn}\left( \sum_{j \in G_l(r_l)} [W_{ij}^{l-l} y_j(t) + W_{ij}^{u-l} x_j(t)] \right), \qquad (5)$$

where $u$ means upper layer, $l$, lower layer, respectively. $W_{ij}^{u-u}$ is connection weight from neuron $x_j$ of upper layer to neuron $x_i$ of upper layer, $W_{ij}^{u-l}$ and $W_{ij}^{l-l}$ are defined similarly.

In quasi-layered RNNM, if we take sufficiently large connectivity $r$ ($r_{u,u} \simeq N, r_{u,l} \simeq N, r_{l,l} \simeq N$), by appropriately determining connection weight $W_{ij}$, a group of arbitrarily designed state patterns can be embedded as cyclic memory attractors. Certain cyclic memory attractors are embedded in each layer. Since three connectivities $r_{uu}$, $r_{ul}$, $r_{ll}$ affect the development of state pattern, for the upper layer and the lower layer, we have calculated the overlap $m(t)$ with respect state pattern $S(t)$ when it evolves for long time. The overlap $m_u(t)$ of the upper layer as a function of connectivity $r_{u,u}$ is same to Fig.4.

Next, let us show examples about long time behaviors of $m_l(t)$ for a few cases of connectivities, $r_{uu}$, $r_{ul}$, $r_{ll}$. They are shown in Fig.5 ($r_{l,l}$ dependence when $r_{u,l} = 0$), Fig.6 ($r_{l,l} + r_{u,l}$ dependence when $r_{u,l} \neq 0$). About the latter case, let us show an example of time development of $m_u(t)$ and $m_l(t)$ in Fig.7 when $r_{u,u} = 40$, $r_{u,l} = 400$, $r_{l,l} = 399$, where the lower layer sensitively responds to the upper layer depending on what trajectories of the upper layer pass through state points near the embedded attractors or far from them.

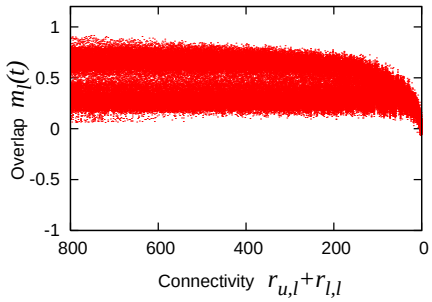



Figure 6: The long-time behaviors of overlap $m_l(t)$ at $K$-step mappings ($r_{u,u} = 40$, $r_{u,l} = 400$). The horizontal axis represents the reduced input connectivity to the lower layer, 0-799.

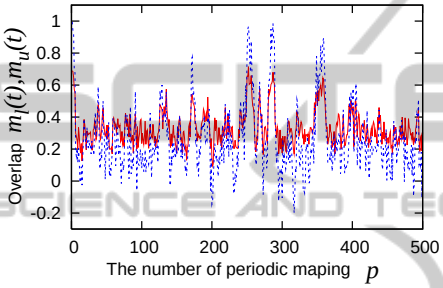



Figure 7: The overlap $m_u(t)$(Red solid line) and $m_l(t)$(Green broken line) along time axis. The horizontal axis represents the $p$-th $K$-steps, where $t = Kp + t_0$ ($K = 6$ and $t_0 = 1200$).

# 4 DESIGNING ATTRACTORS FOR CONTROLLING

## 4.1 Motion Functions

At a certain time $t$, the robot is assumed to be at the present origin and orientates 0 (rad), that is, at any time $t$, the robot has a local coordinates, which is shown in Fig.8. Since the robot has two driving wheels and one castor wheel, when the two driving wheels rotate with same velocity and reverse direction, its rotation radius can be regarded as zero if we do not consider the slippage of the wheels. Therefore, the motion of the robot at each step includes two actions. First, the robot rotates with an angle $\theta(t)$ around the present origin. Next, it moves forward for an distance $L(t)$. In other words, two-dimensional motions of the robot depend these two time variable $\theta(t)$ and $L(t)$. Therefore, in order to realize 2-dimensional motion of the robot using dynamical behavior of the neural network, four hundred dimensional state pattern $S(t)$ is transformed into the rotation angle $\theta(S(t))$ and the moving distance $L(S(t))$ by a simple coding, which are called as *motion functions* and defined by

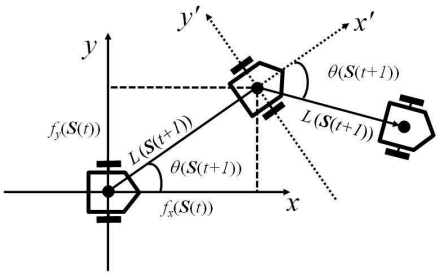



Figure 8: The motion of the robot: at each new position, the robot has a local coordinates in which $x$ axis always orients to the front of the robot.

$$\theta(S(t)) = \tan^{-1}\frac{f_y(S(t))}{f_x(S(t))} \quad (6)$$

$$L(S(t)) = \pi d\sqrt{f_x^2(S(t)) + f_y^2(S(t))} \quad (7)$$

where d is the diameter of the driving wheels, and $f_x(\boldsymbol{S}(t))$, $f_y(\boldsymbol{S}(t))$ are the $x$-axis increment and $y$-axis increment in the local coordinates at time $t$, and are defined by

$$f_x(\boldsymbol{S}(t)) = \frac{4}{N}\boldsymbol{A}\cdot\boldsymbol{C} \qquad f_y(\boldsymbol{S}(t)) = \frac{4}{N}\boldsymbol{B}\cdot\boldsymbol{D} \quad (8)$$

where $f_x(S(t))$ and $f_y(S(t))$ are four $N/4$ dimensional sub-space vectors of state pattern $S(t)$, which is shown in Fig.10. The inner products of $A\cdot B$ and $C\cdot D$ are normalized by $4/N = 100$, so $f_x(S(t))$ and $f_y(S(t))$ ranges from -1 to +1.Therefore, the rotation angle $\theta(S(t))$ takes value from $-\pi$ to $\pi$, and the moving distance $L(S(t))$, from 0 to $\sqrt{2}\pi$d.

## 4.2 Attractors for Controlling

### 4.2.1 Upper Layer

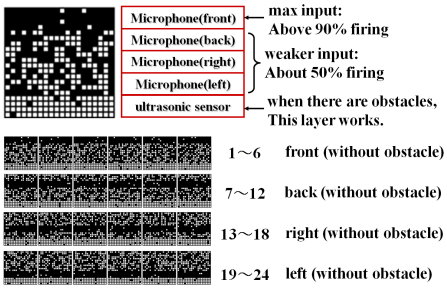



Figure 9: One of four embedded attractors in upper layer.

The upper layer for sensing consists of 5 independent sub-space vectors that corresponds to 5 sensors—four microphones for detecting target direction and a couple of ultrasonic sensors for detecting obstacles, shown in Fig.9. Attractors are embedded in the case that the direction of the maximum signal intensity received by microphones is front, back, right, or left without obstacles, respectively. As the robot is moving, the signal intensity of four microphones changes. Correspondingly, firing state of sensing neurons in the upper layer sensitively responds to external signal input and produce adaptive dynamics to act driving neurons. On the other hand, if there are obstacles to prevent the robot from moving forward, sensing neurons corresponding to ultrasonic sensors are activated, and cause strong chaotic dynamics in sensing neurons to act driving neurons so as to enable the robot to perform complex motions.

### 4.2.2 Lower Layer

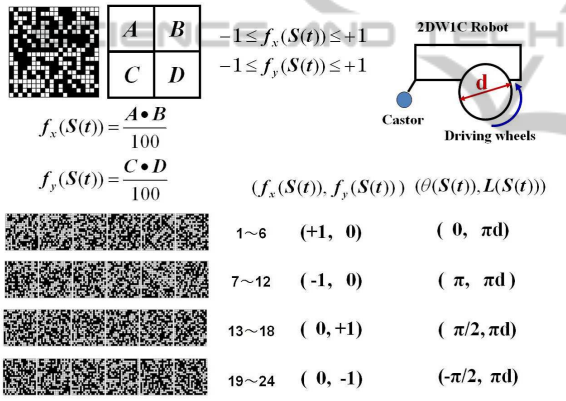



Figure 10: Attractor patterns designed for motion control: Each cyclic memory corresponds to a prototypical simple motion.

The lower layer for driving consists of four groups of attractor patterns, shown in Fig.10. Each attractor pattern consists of four random sub-space vectors $A, B, C$ and $D$. $A$ and $B$ are independent random patterns. A group of specified intra-pattern structure is given, such as $A = C$ or $-C$ and $B = D$ or $-D$ in the present study. By the coding of motion functions, each group of attractor pattern corresponds to one of stationary motions in two-dimensional space.

## 5 CONTROL ALGORITHM

The study on RNNM tells that sufficiently large or quite small connectivity $r$ enables the neural network to generate chaotic dynamics or attractor dynamics (Nara and Davis, 1992). In quasi-layered RNNM, a group of connectivity ($r_{u,u}$, $r_{u,l}$, $r_{l,l}$) should be considered. Here, when $r_{l,l}$ is set as 60, and different $r_{u,u}$ causes chaotic dynamics with different dynamical properties. Correspondingly, by the coding of motion functions, the robot shows weak (localized) chaotic motions ( Fig.11(left)), or strong chaotic motions ( Fig.11(right)). On the other hand, by virtue

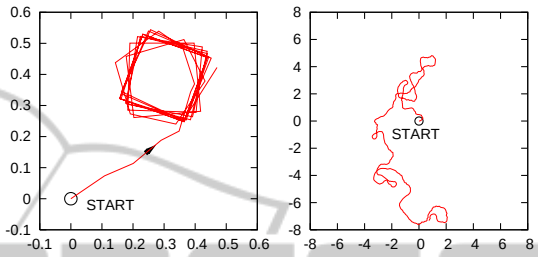



Figure 11: Examples of motion control ($r_{u,l} = 400$ and $r_{l,l} = 60$): (left) larger $r_{u,u} = 60$ generates weak chaotic dynamics with localized property, (right) smaller $r_{u,u} = 20$ causes strong chaotic dynamics.

of the sensitivity of chaos in quasi-RNNM, once the best optimized connectivities are appropriately determined, it will work well even when obstacle information is given to sensing neurons. However, at the present stage, only rough target directional information is given to sensing neurons, and adaptive switching of connectivity $r_{u,u}$ is utilized to control the robot in two-dimensional mazes. The control algorithm is shown in Fig.12. Depending on the sound

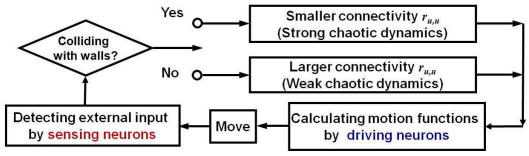
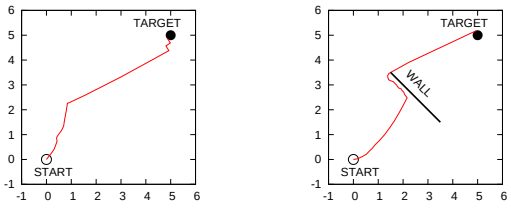



Figure 12: Control algorithm of solving 2-dimensional mazes.

signal intensity from the target caught by four microphones, sensing neurons in the upper layer sensitively response it. Correspondingly, driving neurons in the lower layer sensitively responds to sensing neurons and the robot adaptively turn toward the strongest intensity direction quickly due to sensitivity of chaotic dynamics. When there are no obstacles in the range of ultrasonic sensor, the robot moves with larger connectivity $r_{u,u}$. When there is an obstacle, it moves chaotically with smaller connectivity $r_{u,u}$ and trys to find detour to avoid obstacles. Several examples of computer experiment are shown in Fig.13. In near future, switching of connectivity $r_{u,u}$ will be replaced

with only sensing neurons responding to external input adaptively.

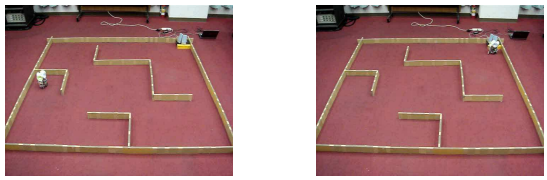



(a) (b)

Figure 13: Examples of Computer simulation: (a) No obstacle, $r_{uu}$ stays at larger one; (b) When obstacles prevent the robot from moving forward, smaller $r_{uu}$ is used to update the network.

# 6 EXPERIMENT OF HARDWARE IMPLEMENTATION

In the present experiment, a loud speaker is set as a target, which is emitting a specified sound signal like calling song of a male cricket. When the robot is moving like a female cricket in two-dimensional space, it can catch only rough directional information from four microphones, which are attached to the front, the back, the left and the right of the robot. According to the control algorithm shown in Fig.12, several kinds of typical 2-dimensional mazes have been constructed. Using chaotic dynamics, the robot successfully avoids obstacles and reaches the target. Fig.14 shows some video snapshots of the robot solving the above 2-dimensional maze. Now we are devoting ourself to implement sensing neurons into the robot, and will report them in near future.



(a) $t = 0$s (b) $t = 408$s

Figure 14: Video snapshots of the roving robot walking on a horizontal floor where there are obstacles between the robot and the target. Around the starting point, because the distance is a little far from the target, the robot shows chaotic motion (a), chaotically walks for finding appropriate detour and reaches the target (b).

# 7 DEVELOPING HARDWARE IMPLEMENTATION & DEVICE FABRICATION

Now, we briefly show our preliminary results about the further developments based on our idea to use functional chaos.

## 7.1 A Roving Humanoid Robot and an Arm Robot Driven by Chaos

Based on our idea, we show the other examples. One is the case that the same method of solving 2-dimensional mazes is applied to a humanoid robot with two legs as shown in Fig.15. This experiment is desinged to develope our idea to 3-dimensional configuration of obstacles, whereas the previous example is to solve 2-dimensional mazes. Thus, the behaviors necessarily include the actions not only of straddling or to step over obstacles but also of climbing up and down obstacles, if they would have appropriate sizes (see Fig.16). This experment is still under going.

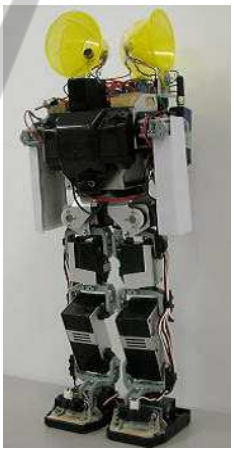


Figure 15: A humanoid robot used in our experiment.

The other experiment is to apply the same idea to arm motions of animals like humanbeings or monkeys. However, it should be noted that we set the situation in which the robot does not have advanced visual information processing ability like manmals but has only poor visual sensing system. The set situation is that the sensors can detect only rough direction of target with including uncertainty. The four infrared light sensors are attached on the arms to realize such ill-posed situations. (see Fig.17) At present, the experiment succeed only in the case without obstacles. (see Fig.18).

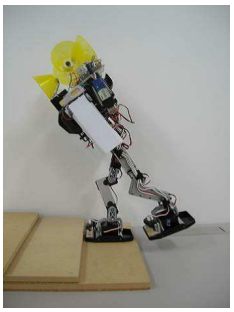


Figure 16: A humanoid robot under the action of climbing upstairs.

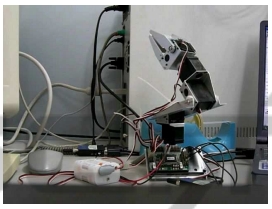


Figure 17: An arm robot used in our experiment.

Figure 18: An arm robot under the action of reaching the target.

## 7.2 A Pseudo-neuron Device and a Diffusively Coupled Network of them

In this section, we report our study about pseudo-neuron device fabricated by Self Electro-optic Effect Device (SEED) and coupled dynamic SEED (D-SEED).

### 7.2.1 Self Electro-optic Effect Device (SEED) & Dynamic SEED

A typical single SEED composed of *p-i-n* semiconductors is shown in Fig. 19. The primary variable of this system is photocarrier density $n$ which is generated by the incident optical power $P_{in}$, where the rate equation for photocarrier density $n$ is given later. The important point is that it indicates a bistable property with respect to incident optical power as shown in Fig.20.

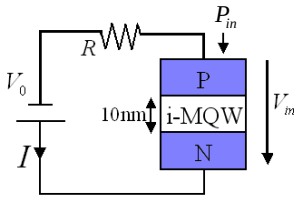



Figure 19: Single SEED.

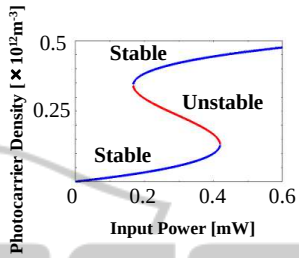



Figure 20: Bistability.

Now let us propose a pseudo-neuron device that consists of two bistable SEED elements optically connected in a series with feedback from one of them to the power of an incident light beam (Fig. 21). We called it "Dynamic SEED (D-SEED)".

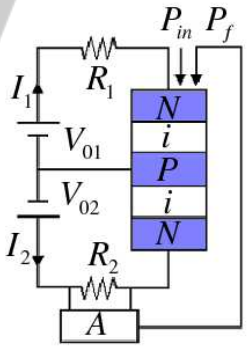



Figure 21: Serially connected SEEDs with feedback.

Coupled rate equations for photocarrier density $n_1$ (upper) and $n_2$ (lower) are represented as

$$\frac{dn_1}{dt} = -\frac{n_1}{\tau_1} + \frac{\alpha_1 (P_{in} + P_f)\Omega_{01}}{\{\omega - \xi_1\}^2 + \left(\frac{\Omega_{01}}{2}\right)^2} \quad (9)$$

$$\frac{dn_2}{dt} = -\frac{n_2}{\tau_2} + \frac{\alpha_2 (P_{in} + P_f - m_1 n_1)\Omega_{02}}{\{\omega - \xi_2\}^2 + \left(\frac{\Omega_{02}}{2}\right)^2} \quad (10)$$

$$\xi_i = \omega_{0i} - \beta (V_{0i} - R_i I_i) \quad (i = 1, 2) \quad (11)$$

$$P_f = A I_2 R_2 \quad (12)$$

where $\tau$, $\alpha$, $\Omega_0$, $\omega$, $\omega_0$, $\beta$, $V_{in}$, $V_0$, $R$, $\eta$ are system parameter. Note that the primary parameter is $P_{in}$ which

causes many kind of bifurcation phenomena. $m$ is absorption parameter and $A$ is feedback gain parameter.

The rate equations show that for the upper SEED feedback light is added to incident light, whereas the lower SEED received the light decreased by the absorption in upper SEED. When we choose appropriate parameter values, we can obtain various bifurcation phenomena as shown in Fig. 22. Important bifurcations are "Hopf bifurcation" and "Saddle-node bifurcation".(Ohkawa et al., 2005)

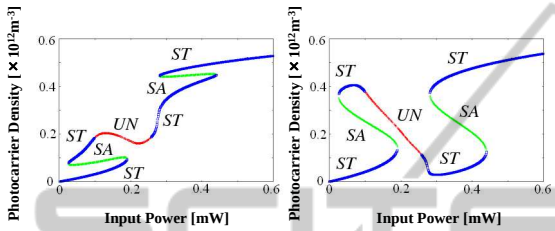



Figure 22: Stationary solutions of upper SEED (left) and lower SEED (right) as a fuction of input light power $P_{in}$. Blue lines, green lines and red lines indicate stable "ST", saddle "SD" and unstable "UN" respectively.

The important point of this case is that a saddle-node bifurcation occurred on the marginal limit cycle, so that, period of limit cycle becomes infinitely long as light power approaches the saddle-node bifurcation point (Fig. 23). Two typical cases of oscillation are shown in Fig. 23. Now we extend our idea to a

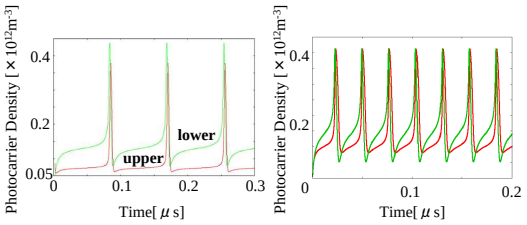



Figure 23: Oscillatory solutions of $n_1$ and $n_2$ for input light power, $P_{in}$ = 0.170 (left), 0.220 (right), respectively. Note that periods become infitely long as they approach edges of saddle-node bifurcation points around unstable regions (Fig. 22).

network of D-SEEDs which are diffusively coupled in two-dimentional array. Coupled rate equations are written include diffusion term from Eq. (13).

$$D\left[\frac{\partial^2 n}{\partial x^2}+\frac{\partial^2 n}{\partial y^2}\right] \cong D\left(n_{i-1,j}+n_{i+1,j}+n_{i,j-1}+n_{i,j+1}-4n_{i,j}\right) \quad (13)$$

Where $(i,\ j)$ means the position of D-SEEDs in arrays located in lattice points in two-dimensions, as shown in Fig. 24 (left-up). Under cetain light power, time-dependence of carrier density shows chaotic dynamics and long time behavior converge into synchronaized periodic oscillation (Fig. 24 (right)). We

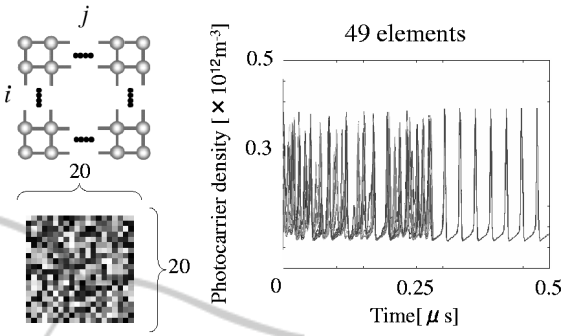



Figure 24: Two-dimensional square lattice D-SEEDs (left-up), snapshot of a network state at a certain time step, where brightness of each cell corresponds to $n_{i,j}$ and is represented in grayscale normalized by maximum value of carrier density (left-low), and Example of time-dependence carrier density (chaotic oscillation).

confirmed it in the numerical simulation for various initial conditions. We obtained global spatio-temporal chaotic dynamics if we sufficiently increase the number of D-SEEDs. By varying light power, we also confirmed multi-stable state. In Fig. 25, we succeeded to obtain switching between two states, chaotic state and multi-stable state, by varying incident light power. Now we try to apply chaos to a novel control by adaptive switching among chaotic state, multi-stable state and synchronized periodic state.

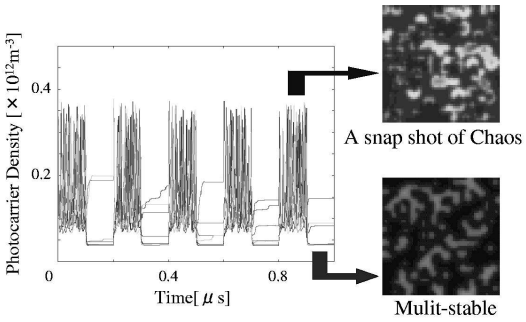



Figure 25: Switching between two states by varying light power and $P_{in}$=0.160 (Chaos) and 0.110 (Multi-stable).

### 7.2.2 Complex Control using Pulsed Neuron Network

In our previous studies, The key idea is adaptive switching between chaotic dynamics and atractor dynamics. Now, it is necessary to develop hardware

implementation of artificial neuron device. As functional examples, let us pick up two cases. One is application to motion control and the other is to memory dynamics using D-SEEDs network (pulsed neuron network). In this paper, only the former is described.

Now let us consider to solve two-dimensional maze as an example of ill-posed problems as shown in the previous sections. We attempt to solve maze by adaptive switching between chaotic dynamics and atractor dynamics with D-SEEDs network. State pattern of the network is represented by 400-dimensional state vectors, while motion in 2-dimensional space is only two-dimensional vectors. Therefore, it is necessary to convert 400-dimensional state to 2-dimensional motion by coding function. We designed 3 different coding methods to prove effectiveness of chaos which is caused various methods. Our coding methods are introduced as follows.

**(a)** Adaptive switching between chaotic state and synchronized periodic state.

**(b)** Adaptive switching between chaotic state and multi-stable state.

Let us consider these two cases. We calculate 2-dimensional motion(Fig.26) from 400-dimensional state by coding function.

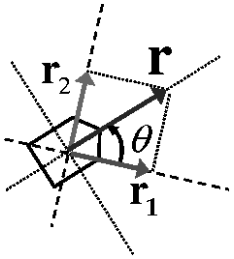



Figure 26: 2-dimensional motion.

In the coding method (a), we introduce the coding but let us discard their detailed description. Next,

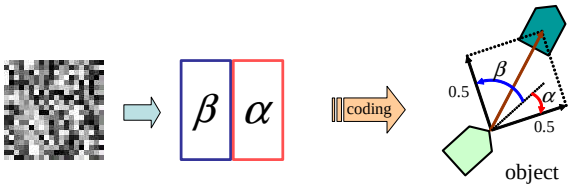



Figure 27: Coding method (b)-1.

in the coding method (b), let us propose the two coding methods. One is shown in Fig.27, and the other is shown in Fig.28, where the detailed definition of motion increments are discarded as well.

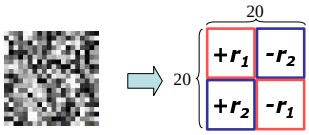



Figure 28: Coding method (b)-2.

So it is possible to obtain 4 monotonic motions which are almost linear motions to 4 quadrants direction and chaotic motion by changing the position. Now, we solve maze by adaptive switching between monotonic motion and chaotic motion with a simple control algorithm shown in Fig. 29. We assume that,

1. An object can acquire only rough direction in which quadrant it exists.
2. The object does not have pre-knowledge about obstacle configulation.

These context settings give an ill-posed problem. According to the control algorithm shown in Fig. 29.

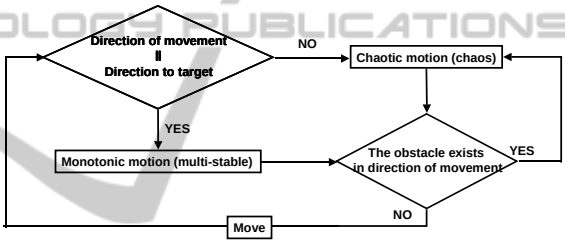



Figure 29: Control algorithm of adaptive switching between chaotic motion and monotonic motion.

The successful results of computer experiments were obtained and will be reported in the conference and in a recent paper as well.

## 8 CONCLUDING REMARKS

Even from the results of the present preliminary experiment, we can conclude that chaotic dynamics is useful to solve complex problems, such as mazes, not only in computer experiments, but also in hardware implementation of the robot systems. Although detail consideration should be done from functional aspects, it is at least an evidence that chaotic dynamics could play important roles in biological systems including brain. Therefore, we are sure that those novel functional aspects of chaotic dynamics could be applied to complex control by simple rule in systems with large but finite degrees, and be useful to engineering application mimicking excellent functions observed in biological systems including brain.

## ACKNOWLEDGEMENTS

This work has been supported by Grants-in-Aid for Scientific Research, #19500191 in Japan Society for the Promotion of Science and #22120509 in the Ministry of Education, Culture, Sports, Science & Technology.

## REFERENCES

Fujii, H., Itoh, H., Aihara, K., Ichinose, N., and Tsukada, M. (1996). Dynamical cell assembly hypothesis-theoretical possibility of spatio-temporal coding in the cortex. In *Neural Networks*. vol. 9, p. 1303.

Huber, F. and Thorson, H. (1985). Cricket auditory communication. In *Sci. Amer.* vol. 253, pp. 60-68.

Li, Y. and Nara, S. (2008). Novel tracking function of moving target using chaotic dynamics in a recurrent neural network model. In *Cognitive Neurodynamics*. vol. 2, pp. 39 - 48.

Mikami, S. and Nara, S. (2003). Dynamical responses of chaotic memory dynamics to weak input in a recurrent neural network model. In *Neural Computing*. vol. 11, pp. 129-136.

Nara, S. (2003). Can potentially useful dynamics to solve complex problems emerge from constrained chaos and/or chaotic itinerancy? In *Chaos*. vol. 13, pp. 1110-1121.

Nara, S. and Davis, P. (1992). Chaotic wandering and search in a cycle memory neural network. In *Progress of Theoretical Physics*. vol. 88, pp. 845-855.

Nara, S., Davis, P., Kawachi, M., and Totuji, H. (1993). Memory search using complex dynamics in a recurrent neural network model. In *Neural Networks*. vol. 6, pp. 963-973.

Nara, S., Davis, P., Kawachi, M., and Totuji, H. (1995). Chaotic memory dynamics in a recurrent neural network with cycle memories embedded by pseudo-inverse method. In *Int. J. Bifurcation and Chaos Appl. Sci. Eng.* vol. 5, pp. 1205-1212.

Ohkawa, Y., Yamamoto, T., Nagaya, T., and Nara, S. (2005). Dynamic behaviors of coulpled self-electro-optic effec devices. In *Appl. Phys. Lett.* vol. 86, p. 111107.

Skarda, C. A. and Freeman, W. J. (1987). How brains make chaos in order to make sense of the world. In *Behavioral and Brain Sciences*. vol. 10, pp. 161-195.

Suemitsu, Y. and Nara, S. (2004). A solution for two-dimensional mazes with use of chaotic dynamics in a recurrent neural network model. In *Neural Computation*. vol. 16, pp. 1943-1957.

Tokuda, I., Nagashima, T., and Aihara, K. (1997). Global bifurcation structure of chaotic neural networks and its application to traveling salesman problems. In *Neural Networks*. vol. 10, pp. 1673-1690.

Tsuda, I. (2001). Towards an interpretaion of dynamic neural activity in terms of chaotic dynamical systems. In *Behavioral and Brain Sciences*. vol. 24, pp. 793-847.