A LEARNING RULE TO CONTROL THE FREQUENCY OF THE OSCILLATORY NEURAL NETWORK

Non-member Hiroaki Kurokawa (Keio University)

Non-member Chun Ying Ho (City University of Hong Kong)

Member Shinsaku Mori (Nippon Institute of technology)

This paper proposes the learning methods to control the frequency of an oscillatory neural network. The learning rules are applied to the neural oscillator that comprises two excitatory neurons, in which only one neuron has a positive feedback weight. Since it is assumed that only the feedback parameter has plasticity, the proposed learning rule can be realized with a high simplicity. By defining the *phase* of the neural oscillation, a mathematical model is conceived so as to conjure up of the blurred vision of phase trajectories in the system. Successful examples of the frequency learning of the sinusoidal function is shown by the computer simulation. With the proposed learning methods, the frequency of the oscillatory neural network can be adjusted to that of any desired value.

Key words: Oscillatory Neural Network, Frequency, Learning, Information Processing

1. INTRODUCTION

The oscillatory phenomenon in the artificial neural networks is one of the most remarkable issues on the study of neural networks and has been studied by many researchers⁽¹⁾⁻⁽²⁾. Detailed analysis of the dynamics of oscillatory neural networks (neural oscillator) have reported in the works⁽³⁾⁻⁽⁴⁾. These oscillatory phenomenon in the neural networks are applied to information processing and are very promising⁽⁵⁾⁻⁽⁷⁾. Since information is expressed as the state of phase or frequency in the information processing using oscillatory neural network, control method of its oscillation (phase or frequency) is necessary to achieve information processing. However only few studies have taken note of the control methods of its frequency. Especially, since the simple neural networks which consists of two neurons are considered as a module in the information processing system, it is worth to show control methods for such a simple neural network.

On the other hand, the learning of the moving pattern is observed in some biological model. Concerning with the phenomenon in the biological models, learning the rhythm of moving patterns can be considered that neural network learns the frequency, and it must be recallable. Studies concerning the learning moving patterns, which also means the maintenance of the synchronization between neural oscillator and a target oscillation have been discussed in the papers⁽⁸⁾⁻⁽¹¹⁾. In these studies, the learning rules, which control the oscillation of the neural oscillator, use the difference of the outputs of each oscillation (activities error).

These algorithms update the weight matrix of the neural oscillator in accordance with such an activity error. The amplitude of the oscillation of the inner state of the neural oscillator changes along with the updates of its weight matrix, and this in turn increases the distortion of the output of the neural oscillator. In many case, this distortion leads a nonzero activity error. Especially for a small size neural oscilla-

tor, such as a 2-neuron oscillator, since the oscillator has few number of weights and the system has low degree of freedom, it is difficult to complete the learning of the target oscillation. Since such a small neural oscillator has high availability to implement to the information processing due to its simpleness, it is also worth to show the learning rule to learn the target frequency.

This paper proposes a learning rule that uses only a plasticity of the self feedback parameter to learn the frequency of the oscillation in the neural network to that of target oscillation. The proposed learning rule is a kind of error feedback learning and the *phase error* rather than the activity error is used to train the neural oscillator in the learning rule. The phase error is defined as the phase lag or lead of the neural oscillation during exactly one period of the target oscillation. In fact, by using this phase error as the training parameter, the phase error in the final stage of the learning will be zero because the period (i.e. frequency) is constant even if the oscillation amplitude is distorted. By the convergence to zero phase error, the converged parameter in the system never change especially in such a error feedback learning, while the update of the parameter may not be zero in the learning rule using activity error. Thus it can be considered that the proposed learning rules achieve higher stability and recallability of the learning than the case using activity error. Most importantly, the neural oscillator has learned parameters that will never change including the learned frequency, such a stability of the proposed learning is strongly desired for the control of the neural oscillator.

In this paper, the learning rule is applied to a neural oscillator that consists of two neurons, in which only one neuron has the positive feedback weight. This means that we can establish a learning method with a high simplicity. Furthermore, since the neural network that has learned parameters will never change its learned frequency, it can be considered that the learned frequency is recallable. In other words, it realizes that the learning and memory of rhythm pattern which is

Fig. 1. An oscillatory neural network

commonly observed in biological models. The objective of this study is to implement the frequency learning of the neural oscillator while phase synchronization is not considered. Successful examples of the frequency learning of the sinusoidal function is shown by the computer simulation in section 5. With the proposed learning methods, the frequency of the oscillatory neural network can be adjusted to any desired value.

2. OSCILLATORY NEURAL NETWORK

Figure 1 shows the neural oscillator model which is composed of two neurons. The interconnections between the two neurons can be either positive or negative while only the neuron N_1 has a positive feedback. Dynamics of the internal states of the neurons N_1 and N_2 in this neural oscillator is governed by,

$$\begin{cases} \tau_1 \frac{dU_1}{dt} = F_1 = -U_1 + w_{12}g(U_2) + T_1g(U_1) + S_1 \\ \tau_2 \frac{dU_2}{dt} = F_2 = -U_2 + w_{21}g(U_1) + S_2 \end{cases}, \quad (1)$$

where U_1 and U_2 are the internal state of the neuron, τ_1 and τ_2 the time constant, and w_{ij} the synaptic weight from neuron N_j to N_i . The Parameter T_1 denotes the coefficient of the positive feedback of N_1 , and S_1 and S_2 are the external inputs to N_1 and N_2 , respectively. Note that these external inputs, S_1 and S_2 are constant, and are not objective functions to be synchronized. The function $g(U_k)$ is the activation function of the neuron, and is assumed to take on the following sigmoidal function,

$$g(U_k) = 1/(1 + \exp(-\lambda U_k)), \quad (2)$$

where λ is the coefficient that control the gradient of the function.

It is well known that neural oscillator exhibits many attractive dynamics such as monostable, bistable or oscillation. As many researchers have shown, the internal states of the neurons in this neural oscillator U_1 and U_2 are oscillatory if the synaptic weights and feedback weight are appropriately adjusted. The dynamics of neural oscillator is strongly depending on the parameters in the oscillator, such as w_{ij} , S_k and T_k . This behavior of the neural oscillator is concerned with the stability of the equilibrium points of the neural oscillator. Details of these derivations can be found in the papers^{(3),(4)}. In the next section, We shall demonstrate how the learning rule which can be applied to such *neural oscillator* as an update rule of the feedback parameter T_1 .

Fig. 2. The characteristics of a period versus feedback parameter T_1

3. LERANING RULE

In this study, we shall present the learning methods to adjust the frequency of a neural oscillator that is comprises of two neurons. The objective of this learning is to adjust the frequency of the neural oscillator to a target one by varying the feedback parameter T_1 . In order to establish a simple learning method, we assume that only the feedback parameter T_1 has plasticity.

Here we shall describe the change of the frequency of the neural state with respect to the feedback parameter T_1 . Figure 2 shows the computer simulation result of the characteristics between the period of the neural oscillator and its feedback parameter T_1 . In this simulation, the parameters are defined as, $\tau_1 = \tau_2 = 1.0$, $w_{12} = -25.0$, $w_{21} = 10.0$, $S_1 = 5.0$, $S_2 = -5.0$, and $\lambda = 2.0$. It can be noted that the period (frequency) of oscillation shows a monotonic increasing (decreasing) behavior with respect to T_1 , and there exist a bounded region within which the neural oscillator is oscillatory. Hence, it can be reckoned that the feedback parameter T_1 must be set to within this region for proper operation.

Let us assume that the target oscillation to learn is the periodic sinusoidal function $I(t)$ which is given by,

$$I(t) = A \sin(2\pi ft) + B \quad (3)$$

Without loss of generality, the angular frequency ω is supposed to be 1.0 throughout this section.

The phase φ_1 , which is derived from the inner states of the neurons, in the neural oscillator can be defined as follows.

$$\varphi_1 = \begin{cases} \cos^{-1} \left(U_1 / \sqrt{(U_1^2 + U_2^2)} \right) & (U_2 \geq 0) \\ -\cos^{-1} \left(U_1 / \sqrt{(U_1^2 + U_2^2)} \right) & (U_2 < 0) \end{cases} \quad (4)$$

Here $\cos^{-1}(x)$ is defined as $(0.0, \pi)$ for $-1 \leq x \leq 1$. In fact, φ_1 is an implicit function of the variable t , and it can be considered that the parameter φ_1 represents the phase of the neural oscillator. By the definition of φ_1 in Eq.(4), we can compare the state of the neuron with that of the target oscillation in the same system of coordinates. In the following, we shall propose the learning methods that rely on the parameter φ_1 for the update the feedback parameter T_1 .

3.1 learning rule 1 First we shall present one of the simple learning rules. This learning rule uses the phase error between the target oscillator and the neural oscillator to determine whether the phase of the neural oscillator is leading or lagging. This can be achieved by taking the difference of the phase errors at the beginning and at the end of one cycle of the target oscillation. In order to illustrate the learning

Fig. 3. Phase differences

rule 1, we first define $\Delta\varphi_n$ as follows:

$$\Delta\varphi_n = \varphi(t + \frac{1}{f}) - \varphi(t) \quad (n = 0, 1, 2, \dots) \quad (5)$$

Here $\varphi_n(t)$ is the phase of the neural oscillator at time t during the n -th cycle of the target frequency, and $\Delta\varphi_n$ is the phase difference between the neural oscillator and the n -th period of the target oscillation. If $\Delta\varphi$ is less than zero, then the neural oscillator is lagging as shown in Fig. 3, and its frequency must be higher to learn the target frequency. On the other hand, if $\Delta\varphi$ is greater than zero, the neural oscillator is leading, and its frequency must be lower. From Fig. 2 that depicts the variation of the frequency of the neural oscillator with respect to the feedback parameter T_1 , the update rule of the feedback parameter can be derived as,

$$\Delta T_1 = \varepsilon \Delta\varphi. \quad (6)$$

Where ε is the learning rate which is positive and constant. The learning is realized by following update rule of $T_1(t)$.

$$T_1(t + t_{step}) = T_1(t) + \Delta T_1 = T_1(t) + \varepsilon \Delta\varphi. \quad (7)$$

Where t_{step} is the time step of the update of feedback parameter T_1 . Here t_{step} is defined as the time long enough thereafter the oscillation of neural oscillator is stable. Since the characteristics of the frequency versus T_1 is monotonic decreasing, it is considered that $\varepsilon\Delta\varphi$ converges to 0.0 upon the update of T_1 . And the frequency of the neural oscillator will be adjusted to k -times of the target frequency, where k is an integer.

The reason behind is that the learning rule 1 algorithm simply compares the 2 phases of neural oscillator at the beginning and the end of the period of the target oscillation, and the phase error can be accumulated to the nearest 2π , i.e., one cycle. Hence this learning rule 1 will be effective only if the initial frequency of the neural oscillator is close to that of the target oscillation or it is required or allowed to adjust the learning frequency to k -times of the target frequency.

3.2 learning rule 2 To achieve the exact frequency learning, we propose the learning rule 2. This learning rule also makes use of the phase error between the neural oscillator and the target oscillator for the frequency learning. However, different from the learning rule 1, the phase error in the learning rule 2 is defined as the accumulated phase change of the neural oscillator within one cycle of the target oscillation. Since phase error will be nonzero if the frequency of the neural oscillation is not equal to that of the target oscillation, the exact frequency learning can be accomplished by means of learning rule 2. As the learned oscillation amplitude may not be symmetric, it is

Fig. 4. The changes of trajectory in phase plane

difficult, if not impossible, to define the relative amplitude difference in one time step. On the contrary, this kind of learning can be readily implemented by using the phase error rather than the activity error.

Similarly to learning rule 1, we define the $\Delta\sigma$ as follows:

$$\Delta\sigma = \sum_{t=t}^{t+1/f} \Delta\varphi(t) - 2\pi \quad (n = 0, 1, 2, \dots) \quad (8)$$

Where $\Delta\varphi(t)$ is given by,

$$\Delta\varphi(t) = \varphi(t + \Delta t) - \varphi(t). \quad (9)$$

The parameter Δt is supposed to be very small comparing to the period of the target oscillation,

$$\Delta t \ll \text{period}. \quad (10)$$

The parameter $\Delta\sigma$ is the phase difference of the neural oscillation from that of the target oscillation. Then the changes of feedback parameter are given by,

$$\Delta T_1 = \varepsilon \Delta\sigma. \quad (11)$$

Where ε is the learning rate which is positive and constant. Assuming that the time step of the update is t_{step} and it is long enough for the oscillation of neural oscillator to be stable oscillation, the update rule of $T_1(t)$ can be written by,

$$T_1(t + t_{step}) = T_1(t) + \Delta T_1 = T_1(t) + \varepsilon \Delta\sigma. \quad (12)$$

Similarly to 2π -point learning, $\Delta\sigma$ converges to 0.0 upon the update of T_1 . By applying this learning rule 2, the frequency of the neural oscillator can be adjusted to the exact target frequency

4. PROPERTY OF THE LEARNING RULE

The property of the proposed learning rule can be easily clarified from a mathematical standpoint. The dynamics of the oscillation in the neural oscillator is shown as the trajectory in the phase plane. Figure 4 shows the trajectory of the neural oscillator (thin line) and that of the target oscillation (thick line). Here adjustment of the frequency of the neural oscillator corresponds to the correction of trajectory of the neural oscillator to that of the target oscillator. In order to clarify the property of the proposed learning rule, we shall consider the behavior of the trajectory when the parameter T_1 is updated.

From Eq.(1), we can easily obtain following equation which represents the derivative of trajectory in the phase plane.

$$\eta(U_1, U_2, T_1) = \frac{dU_2}{dU_1} = \frac{-U_2 + w_{21}f(U_1) + S_2}{-U_1 + w_{12}f(U_2) + T_1f(U_1) + S_1} \quad (13)$$

Hereafter we assume that (U_{11}, U_{12}) is a point on the trajectory of the learning oscillation, and T_1 updates its value to $T_1 + \Delta T_1$ then. Now,

Fig. 5. Relationships between the period and the U_1 at $U_2 = 0$, $U_1 > 0$ with the variations of feedback parameter T_1

let us consider the two types of oscillation; the frequency of a neural oscillator is higher than the target frequency and vice versa.

Taking into account of the proposed learning rule, when the frequency of neural oscillator is higher than the target frequency, ΔT_1 of Eq.(13) can be calculated as greater than 0. Otherwise ΔT_1 will be less than 0. From Eq.(13), we can know that a differential coefficient of the trajectory of the neural oscillator changes as shown in followings in each quadrant in the phase plane.

- (1) (U_{11}, U_{12}) is in the quadrant I $\rightarrow$ the denominator of Eq.(13) is less than 0, the numerator of Eq.(13) is greater than 0
 - (a) $\Delta T_1 > 0 \rightarrow \eta(U_{11}, U_{12}, T_1 + \Delta T_1) < \eta(U_{11}, U_{12}, T_1)$
 - (b) $\Delta T_1 < 0 \rightarrow \eta(U_{11}, U_{12}, T_1 + \Delta T_1) > \eta(U_{11}, U_{12}, T_1)$
- (2) (U_{11}, U_{12}) is in the quadrant II $\rightarrow$ the denominator of Eq.(13) is less than 0, the numerator of Eq.(13) is less than 0
 - (a) $\Delta T_1 > 0 \rightarrow \eta(U_{11}, U_{12}, T_1 + \Delta T_1) > \eta(U_{11}, U_{12}, T_1)$
 - (b) $\Delta T_1 < 0 \rightarrow \eta(U_{11}, U_{12}, T_1 + \Delta T_1) < \eta(U_{11}, U_{12}, T_1)$
- (3) (U_{11}, U_{12}) is in the quadrant III $\rightarrow$ the denominator of Eq.(13) is greater than 0, the numerator of Eq.(13) is less than 0
 - (a) $\Delta T_1 > 0 \rightarrow \eta(U_{11}, U_{12}, T_1 + \Delta T_1) > \eta(U_{11}, U_{12}, T_1)$
 - (b) $\Delta T_1 < 0 \rightarrow \eta(U_{11}, U_{12}, T_1 + \Delta T_1) < \eta(U_{11}, U_{12}, T_1)$
- (4) (U_{11}, U_{12}) is in the quadrant IV $\rightarrow$ the denominator of Eq.(13) is greater than 0, the numerator of Eq.(13) is greater than 0
 - (a) $\Delta T_1 > 0 \rightarrow \eta(U_{11}, U_{12}, T_1 + \Delta T_1) < \eta(U_{11}, U_{12}, T_1)$
 - (b) $\Delta T_1 < 0 \rightarrow \eta(U_{11}, U_{12}, T_1 + \Delta T_1) > \eta(U_{11}, U_{12}, T_1)$

The trajectory of the neural oscillator changes depending on the update of the feedback parameter T_1 , and the differential coefficient of the trajectory changes as shown in previous. Thus, in the case that T_1 increases the radius of the trajectory becomes larger, and in the case that T_1 decreases the trajectory becomes smaller. Under the assumption that only T_1 has plasticity, the higher the frequency of the oscillator, the smaller its trajectory in the phase plane as shown in Fig. 5. In short, ΔT_1 of Eq.(13) increases, the frequency of the neural oscillator can be lower, and the frequency will be higher otherwise. In Fig. 5, parameters in the neural oscillator are assumed to take on the values of $\tau_1 = \tau_2 = 1$, $w_{21} = 10$, $w_{12} = -25$, $S_1 = 5$, $S_2 = -5$ and $\lambda = 2$.

As shown in Fig. 4, we can find that the trajectory of neural oscillator will be corrected to the target trajectory, and thus the frequency of neural oscillator will approach to the target frequency. The correction towards the target trajectory continues until the trajectory of neural oscillator coincides with the target trajectory.

The amplitude of update is determined by the phase error and learning rate ϵ , which is shown in Eq.(6) and Eq.(11). Thus the amplitude

Fig. 6. The wave form of the oscillation (learning rule 1); (a) Target oscillation $I(t)$ ($\omega = 0.5$): $5.0 < t < 15.0$, (b) $120.0 < t < 130.0$ (c) Neuron's state U_1 : $5.0 < t < 15.0$ (before learning), (d) $120.0 < t < 130.0$ (after learning)

of the update become smaller as the trajectory of the neural oscillator become close to the target trajectory. Finally, it will become zero after a sufficient times of update. Although the trajectory of neural oscillation never conform to the target oscillation, the update diminishes to zero because the frequency, i.e. period, will be stable and the proposed learning rule only consider frequency adjusting to the target oscillation.

5. SIMULATION RESULTS

In the following, we shall present the computer simulation results of the proposed learning methods. In the simulation, parameters are set to as follows: $\tau_1 = \tau_2 = 1.0$, $w_{12} = -25.0$, $w_{21} = 10.0$, $S_1 = 5.0$, $S_2 = -5.0$, $\lambda = 2.0$. We also assumed that the parameter ω in Eq.(3) is 0.5 and 0.25 in the simulation of learning rule 1 and learning rule 2. In all simulations, differential equations are calculated using the fourth-order runge-kutta method.

5.1 learning rule 1 First we show the simulation results of learning rule 1 in Fig. 6. Figure 6 (a) and (b) show the target oscillation, and (c) and (d) show the oscillation of the neural oscillator. In this simulation the learning of the neural network is assumed to be started from time 20.0 and ω of the target oscillation is assumed to be 0.5. Figure 6 (c) shows the results where learning has not been applied, and (d) shows the simulation results where learning has been applied. Here in Fig. 6(d), the feedback parameter T_1 has converged to the optimum value already. In the simulation, the period of neural oscillator after the learning converged is 2, and this is exactly same as that of the sinusoidal function given by Eq. (3). The results of the simulation showed that the frequency of the neural oscillator is adjusted to the target frequency by using the proposed learning rule 1.

The changes of the feedback parameter T_1 versus time is shown in Fig. 7. We can observe that the convergence of T_1 to the optimum value. From the results in the Fig. 7, we can conclude that the convergence ability does not depend on the learning rate ϵ . However the speed of the convergence does depend on ϵ . From Fig. 7, we can find that the larger the value of learning rate, the higher is the speed of the convergence of the feedback parameter T_1 . However, using a too large value of ϵ hinder a proper convergence of the learning. For example, according to another simulation, in the case that ϵ is around 2.2 or over, learning is not converged under the condition in the simulation as shown in Fig. 8. In Fig. 8, it is shown that the case when ϵ is 4.0. However it is not so difficult to define the parameter ϵ , since the region of ϵ , converge or not, can be divide by the only one threshold.

Fig. 7. Changes of the feedback parameter T_1 (learning rule 1)

Fig. 8. Changes of the feedback parameter T_1 (learning rule 1, $\epsilon = 4.0$)

Fig. 9. The wave form of the oscillation (learning rule 1); (a) Target oscillation $I(t)$ ($\omega = 0.25$): $5.0 < t < 15.0$, (b): $120.0 < t < 130.0$ (c) Neuron's state U_1 : $5.0 < t < 15.0$ (before learning), (d) $120.0 < t < 130.0$ (after learning)

As mentioned in section 3.1, we have explained that the frequency of the neural oscillator may be adjusted to k -times of the target frequency with learning rule 1, where k is an integer. In Fig. 6, the frequency of the neural oscillator is shown to be the exact target frequency. This is the case where $k = 1$. In Fig. 9, we show another simulation results of learning rule 1, in which the frequency is learned to be 2 times of that of target oscillation (e.g. $k = 2$). In the simulation, the period of neural oscillator after the learning converged is 2, while the period of sinusoidal function is 4 ($\omega = 0.25$). The neural oscillating frequency is dictated by the initial condition of the feedback parameter T_1 . To elaborate, the oscillating frequency will converge to the nearest multiple of target frequency depending on the initial frequency before the learning is applied.

5.2 learning rule 2 Secondly, we show the simulation results of learning rule 2. Similarly to the results of the learning rule 1, Fig. 10 (a) and (b) show the target oscillation, and (c) and (d) show the oscillation of the neural oscillator. The learning rule 2 is assumed to be starting from time 20.0 and ω of the target oscillation is assumed to be 0.5. In the simulation, the period of neural oscillator after the learning converged is 2, and this is exactly same to the period of sinusoidal function given by Eq. (3). The results of the simulation show that the frequency of the neural oscillator is adjusted to the target frequency by using the proposed learning rule 2.

In this simulation, the frequency of neural oscillator converges to

Fig. 10. The wave form of the oscillation (learning rule 2); (a) Target oscillation $I(t)$ ($\omega = 0.5$): $5.0 < t < 15.0$, (b): $120.0 < t < 130.0$ (c) Neuron's state U_1 : $5.0 < t < 15.0$ (before learning), (d) $120.0 < t < 130.0$ (after learning)

Fig. 11. The wave form of the oscillation (learning rule 2); (a) Target oscillation $I(t)$ ($\omega = 0.25$): $5.0 < t < 15.0$, (b): $120.0 < t < 130.0$ (c) Neuron's state U_1 : $5.0 < t < 15.0$ (before learning), (d) $120.0 < t < 130.0$ (after learning)

Fig. 12. Changes of the feedback parameter T_1 (summation learning)

exactly the same frequency of the target oscillation. This ability of learning does not depend on the initial value of feedback parameter T_1 as mentioned in section 3.2. To clarify this ability of learning rule 2, we show another simulation results of learning rule 2 in Fig. 11. In this simulation, the parameter ω in Eq.(3) is 0.25 and it cannot be realized by means of learning rule 1 as shown in Fig.9. However, in the simulation, the period of neural oscillator after the learning converged is 4, and this is exactly same to the period of sinusoidal function given by Eq. (3). In short, the frequency of neural oscillator is adjusted to exactly the same frequency of the target oscillation by means of learning rule 2.

Similarly to the simulation of learning rule 1, we also show the changes of the feedback parameter T_1 in Fig. 12 where the parameter ω of the target oscillation is 0.5. We can also observe a convergence to the optimum value. Most important, convergence ability does not depend on the learning rate, ϵ . However Fig. 12 also shows that the speed of the convergence depends on the learning rate, ϵ , i.e. the larger the value of learning rate, the faster is the speed of the convergence of feedback parameter T_1 . Similar to the case of learning rule 1, using a too large value of ϵ hinder a proper convergence of the learning. As shown in the Fig. 12, the convergence behavior of the learning rule 2 almost resembles that of the learning rule 1.

The merit of the learning rule 1 is that calculation in the learning

is reduced. On the other hand, comparing to learning rule 1, learning rule 2 achieves a proper frequency learning that does not depend on any initial conditions of the neural oscillator. In other words, the learning rule 2 has an ability to learn wide-range of frequency.

6. CONCLUSIONS

In this paper, we proposed two learning rules to control the frequency of the oscillatory neural network. Since these learning methods require only one plastic connection in the network, it can be considered that the system has a high simplicity. We also depicted the property of our learning rules from a mathematical point of view. Since the neural network that has learned parameters will never change its learned frequency, one can conclude that the model is very useful to implement the learning and memory of rhythm pattern which is commonly observed in biological models. Simulation results have been presented to demonstrate the efficiency of the proposed system.

(Manuscript received July 25, 97, revised February 23, 98)

References

- (1) C. M. Gray, P. König, A. K. Engel, and W. Singer "Oscillatory responses in cat visual cortex exhibit inter-columnar synchronization which reflects global stimulus properties," *Nature*, vol. 338, pp.334-337, 1989.
- (2) R. Eckhorn, R. Bauer, W. Jordan, M. Brosch, W. Kruse, M. Munk, and H. J. Reitboeck, "Coherent Oscillations: A Mechanism of Feature Linking in the Visual Cortex? -Multiple Electrode and Correlation Analyses in the Cat", *Biol. Cybern.*, vol. 60, pp.121-130, 1988.
- (3) Shun-ichi Amari, "Random Nets Consisting of Excitatory and Inhibitory Neuron-like Elements" *IEICE Transactions* vol.55-D pp.179-185 (1972), (in Japanese)
- (4) Shun-ichi Amari, "Characteristics of Random Nets of Analog Neuron-Like Elements", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-2, no.5 pp. 643-657 (1972)
- (5) DeLiang Wang, "Emergent Synchrony in Locally Coupled Neural Oscillators", *IEEE Transactions on Neural Networks*, vol.6, no.4 pp. 941-948 (1995)
- (6) Ch. von der Malsburg and W. Schneider, "A Neural Cocktail-Party Processor", *Biol. Cybern.*, vol.54, pp. 29-40 (1986)
- (7) Ch. von der Malsburg and Joachim Buhmann, "Sensory segmentation with coupled neural oscillators", *Biol. Cybern.*, vol.67, pp. 233-242 (1992)
- (8) Kenji Doya and Shuji Yoshizawa, "Motor Pattern Memory in Neural Network" *IEICE technical report MBE87-141*(1987), (in Japanese)
- (9) R. Hecht Nielsen, "Neurocomputing", Addison Wesley (1990)
- (10) J.J. Hopfield and D.W. Tank, " "Neural" Computation of Decisions in Optimization Problems", *Biological Cybernetics* vol. 52, pp.141-152(1985)
- (11) Hirofumi Nagashino and Yohsuke Kinouchi, "An Oscillatory Input Makes a Neural Network Nonoscillatory", *Proc. of International Symposium on Nonlinear Theory and Its Application (NOLTA'95)*, pp.1137-1140
- (12) Hiroaki Kurokawa and Shinsaku Mori, "A Learning Method to Control the Frequency of the Oscillatory Neural Network", *Proc. of International Conference on Neural Networks (ICNN'95)*, pp.1521-1524

Hiroaki Kurokawa (Non-member) was born in Hiroshima, Japan, on September

3, 1969. He received the B.E. and M.E. degrees in Electrical Engineering from Keio University, Yokohama, Japan, in 1993 and 1995, respectively. Currently, he is studying toward the Ph.D. degree in the Department of Electrical Engineering, Keio University. He is mainly engaged in research on neural networks. He is a student member of the Institute of

Electrical and Electronics Engineers (IEEE), the International Neural Network Society (INNS), and the Institute of Electronics, Information and Communication Engineers (IEICE).

Chun Ying Ho (Non-member) was born in Hong Kong on January 23, 1966.

He received the B.Sc. and M.Ph. degrees in Electronic Engineering from the Chinese University of Hong Kong in 1987 and 1989, respectively. He joined Keio University, Yokohama, Japan in 1990, and obtained the Ph.D. degree in Electrical Engineering in 1994. He was with SGS Microelectronics (Japan) as an IC design engineer from 1994 to 1996, where he

conducted several projects related to HDD and DVD IC design. In 1996, he returned to Hong Kong and is presently a Research Assistant Professor at the City University of Hong Kong. His research interests include neural networks, genetic algorithm, digital IC design, and image processing. Dr. Ho is a member of the Institute of Electrical and Electronics Engineers (IEEE) and the Institute of Electronics, Information and Communication Engineers (IEICE).

Shinsaku Mori (Member) was born in Kagoshima, Japan, on August 19, 1932.

He received the B.E., M.E., and Ph.D. degrees in Electrical Engineering from Keio University, Yokohama, Japan, in 1957, 1959 and 1965, respectively. 1957-1996, he had been with the Department of Electrical Engineering, Keio University, Japan, where he was a professor. During the academic year 1978-1979 he was on leave from Keio University as a Visiting Pro-

fessor of Electrical Engineering at University of Wisconsin, U.S.A. Since 1997, he has been with the Department of Electrical and Electronics Engineering, Nippon Institute of Technology, Japan, where he is a professor. His research interests include circuit theory, and medical engineering, especially on nonlinear circuits, chaos, digital phase-locked loops, modulation and coding, and hyperthermia. Dr. Mori is a member of the Institute of Electrical Engineers of Japan, the Japan Society for Simulation Technology, the Society of Information Theory and Its Applications, the Institute of Electronics, Information and Communication Engineers (IEICE), and the Institute of Electrical and Electronics Engineers (IEEE).