카오틱 신경망을 이용한 적응제어에 관한 연구

A study on the Adaptive Neuro Controller with Chaotic Neural Networks

김 상 희*, 박 원 우*, 안 희 욱*
Sang Hee Kim, Won Woo Park, Hee Wook Ahn

요 약

본 논문은 개선된 카오틱 신경망을 이용한 비선형 시스템의 적응제어에 관한 것이다. 개선된 카오틱 신경망은 기존의 카오틱 신경망을 간략화하며 동적 특성을 강화하기 위하여 제안하였다. 또한 새로운 동적 역전파 학습방법을 개발하였다. 제안된 신경회로망은 다변수 시스템의 시스템식별과 신경망 적응제어 시스템에 적용하였다. 제안된 신경망은 비선형 동적시스템에 우수한 적응성을 가지므로 시뮬레이션 결과는 우수한 성능을 보였다.

Abstract

This paper presents an indirect adaptive neuro controller using modified chaotic neural networks(MCNN) for nonlinear dynamic system. A modified chaotic neural networks model is presented for simplifying the traditional chaotic neural networks and enforcing dynamic characteristics. A new Dynamic Backpropagation learning method is also developed. The proposed MCNN paradigm is applied to the system identification of a MIMO system and the indirect adaptive neuro controller. The simulation results show good performances, since the MCNN has robust adaptability to nonlinear dynamic system.

Key words : Chaotic Neural Network, Adaptive Control, Dynamic Backpropagation, System Identification

I. Introduction

Recently, the chaotic neural networks(CNNs) having chaotic characteristics like biological neuron have been studied in application to nonlinear dynamic systems because of it's highly nonlinear dynamic characteristics. Biological neurons generally have chaotic characteristics permanently or transiently. The chaotic responses of biological neurons have been modeled quantitatively by many researchers. The primitive model was the Hodgkin Huxley equation. Caianiello and Nagumo Sato modified this model to make chaotic neural networks.[1,2] Aihara et al. proposed a discrete time model with continuous output, and applied this model to chaotic neural networks.[3] They showed that the neural networks could be applied to solve optimization problems such as traveling salesman problem(TSP). The effects of chaotic response have not verified yet by analytical methods. The chaotic characteristics of neuron model generally gives adverse effects on optimization problems, but the transient chaos of neuron model could be beneficial to overcome the local minimum problem. Aihara proposed that the transient chaotic characteristics of neuron could be helpful for global optimization.[4] Even though some modifications on chaotic neuron model, those previously proposed chaotic neuron models are still complicate, and need more dynamic characteristics in neuron itself and learning algorithm.

Nguyen and Widrow[5], Narendra and Parthasarathy[6] proposed indirect control method using two neural networks. Neural Networks identifier in those papers

*금오공과대학교 전자공학부
접수 일자 : 2003. 4. 16　　수정 완료 : 2003. 7. 18
논문 번호 : 2003-2-9
※본 연구는 2001년도 금오공과대학교 학술연구비에 의하여 연구된 논문임

requires pretaining as an inverse model of the system. Those neural networks controllers are not adapted immediately due to uncertainties or disturbances of the system parameters. To overcome these limitations, we proposed a direct adaptive control strategies using chaotic neural networks.[7] The structure of the paper consisted of the fixed PD controller and the chaotic neural networks.

Since conventional CNNs' structure and learning rule does not proper to system identification and control application, a modified chaotic neuron model is presented to simplify the model and to enforce dynamic characteristics. We also modify the learning rule with dynamic backpropagation algorithm for the proposed MCNN. This structure is very compatible with highly nonlinear dynamic system in the neural network structure and learning rule. Chaotic neural networks could be substituted with feed forward neural networks and recurrent neural networks because of complex nonlinearity in chaotic neurons. In this paper, the proposed MCNN paradigm is applied to the system identification of a MIMO system and an indirect adaptive neuro controller. The simulation results show good performances, since the MCNN has the robust adaptability to nonlinear dynamic system.

II. Modified Chaotic Dynamic neural networks

The conventional chaotic neuron model, suggested by Nagumo and Sato[3], has two different type of inputs: one from same layer and the other from extraneous layer, and also has a refractory term, which is self feedback. The refractory term performs effectively dynamic characteristics through repeated signal controlling as one of three terms, which affect output of the chaotic neuron. The neuron model is shown in Fig. 1.

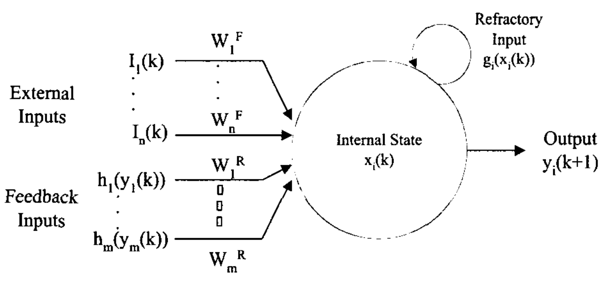


Fig. 1. Chaotic neuron unit

Generally, the dynamics of the i th chaotic neuron in networks at discrete time $k+1$ is

$$y_i(k+1) = f_N[\sum_{j=1}^{n} w_{ij}^F \sum_{r=0}^{k} K_s^r I_j(k-r) + \sum_{j=1}^{m} w_{ij}^R \sum_{r=0}^{k} K_m^r h_j(y_i(k-r)) - \alpha \cdot \sum_{r=0}^{k} K_r^r g_i(x_i(k-r)) - \theta_i] \quad (1)$$

where $f_N(\cdot)$ is a sigmoid function, w_{ij}^F and w_{ij}^R are *coupling coefficients(weights) from the* j th external neuron and the j th feedback neuron to the i th neuron, respectively. $I_j(k-r)$ is the strength of the j th externally applied input at time $k-r$, $h_j(y_i(k-r))$ is a transfer function of the axon connected on the j th chaotic neuron, and $g_i(x_i(k-r))$ is a refractory function of the i th chaotic neuron at time $k-r$, usually an identity function. The n and m are the numbers of external and feedback inputs applied to the chaotic neuron. The decay parameters, K_s^r, K_m^r, and K_r^r are the damping factors of the external, feedback, and refractoriness, respectively. In this paper, we assumed the same values of decay parameters, K. The θ_i is the threshold of the i th chaotic neuron. The constant α is a positive decay parameter of refractory. ($0 \le \alpha$)

$$x_i(k+1) = K \cdot x_i(k) + \sum_{j=1}^{n} w_{ij}^F I_j(k) + \sum_{j=1}^{m} w_{ij}^R h_j(f_N(x_i(k))) - \alpha g_i(f_N(x_i(k))) - \theta_i(1-K) \quad (2)$$

In order to apply the continuous Hopfield neural network structure to the recurrent inputs, Aihara et al. defined the symmetric structure of recurrent weights as $w_{ij}^R = w_{ji}^R,\ w_{ii}^R = 0$. This neural network uses two kinds of learning rules in the same network. Since the structure of that model decreases the efficiency of learning and the dynamic characteristics of networks, this model is not appropriate for modeling dynamic systems.

Since the chaotic neuron model, which was proposed by Aihara, is too complicate, simplification is needed for reducing the computation time. This paper presents a chaotic dynamic neuron unit with same chaotic characteristics. The $\alpha g_i(f_N(x_i(k)))$ term in equation (2) is overlapped in the term of $\sum_{j=1}^{m} w_{ij}^R h_j(f_N(x_i(k)))$ when $i=j$, so the $\alpha g_i(f_N(y_i(k)))$ is abbreviated in this modified model. For more simplification, the threshold term, $\theta_i(1-K)$, is defined as zero, and the nonlinear function, $h_j(\cdot)$, is defined 1 . Fig. 2 shows the modified chaotic neuron unit.

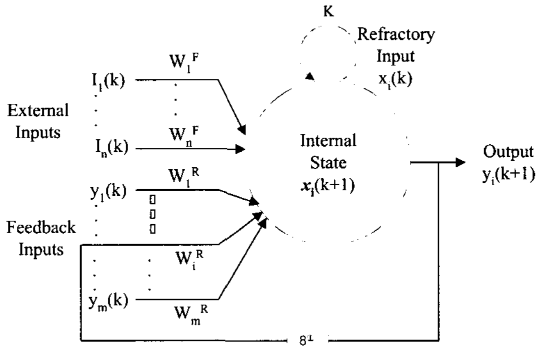


Fig. 2. Modified Chaotic Dynamic Neuron Unit

Then the reduced form of Eq. (2) is

$$x_i(k+1) = K \cdot x_i(k) + \sum_{j=1}^{n} w_{ij}^F \cdot I_j(k) + \sum_{j=1}^{m} w_{ij}^R \cdot y_i(k) \tag{3}$$

$$y_i(k+1) = f_N[x_i(k+1)] \tag{4}$$

$$f_N[x_i(k+1)] = \frac{1}{1+e^{-x_i(k+1)/\varepsilon}} \tag{5}$$

where ε is slope of sigmoid function.

To increase dynamic characteristics, the nonsymmetric weights are applied to recurrent inputs such as $w_{ij}^R \neq w_{ji}^R,\ w_{ii}^R \neq 0$. The chaotic neuron sums three inputs; the refractoriness, $K \cdot x_i(k)$, the activation, $\sum^{n} w_{ij}^F \cdot I_j(k)$, and the recurrent input, $\sum^{m} w_{ij}^R \cdot f_N(x_i(k))$. The summation passes through the nonlinear sigmoid function.

III. Dynamic Backpropagation Learning

Although the chaotic neuron model inherently has robust dynamic characteristics, the traditional chaotic neural networks(CNN), proposed by Aihara et al. decrease the dynamic characteristics in the structure and the learning rules. They used the backpropagation learning rule for the forward inputs between layer and used the time progressing learning rule(the continuous Hopfield learning algorithm) for the recurrent inputs in inter layer. These learning rules may be appropriate to the static patterns but not to the dynamic system applications such as forecasting, identifications, signal processing and dynamic system control. In this paper, the structure of CNN is modified, and the new learning rule is proposed for enhancing the dynamic characteristics.

Modified chaotic neural network is a globally coupled neural networks. Each chaotic neuron unit is globally coupled with present and past outputs of chaotic neuron units. Modified chaotic neural networks in Fig. 3 have two different coupling coefficients (weights) for both directions among the neurons of interlayer, and forward direction between layers. This connection weights in interlayer is defined as nonsymmetric form, $w_{ij}^{OR} \neq w_{ji}^{OR},\ w_{ii}^{OR} \neq 0$ and $w_{ij}^{IR} \neq w_{ji}^{IR},\ w_{ii}^{IR} \neq 0$.

This structure is similar with fully recurrent neural networks.

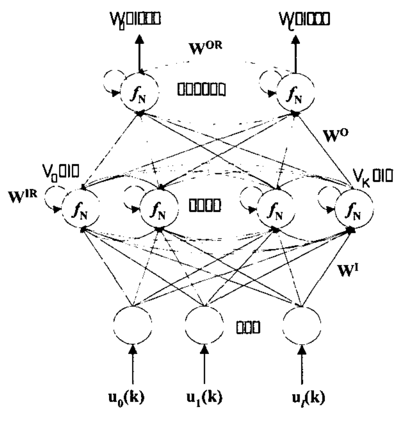


Fig. 3. Modified Chaotic Neural Network Structure

Consider fig. 3, $u_i(k)$ is the i th input for each discrete time k, $S_j^H(k)$ is the weighted sum of inputs and refractory input to j th neuron in hidden layer, $x_j(k)$ is the output of j th neuron in hidden layer, K is refractory parameter of chaotic neuron, and $f_N(\cdot)$ is nonlinear sigmoid function. $\mathbf{W}^I$ and $\mathbf{W}^{IR}$ represent the weight vector between input and hidden layer and inter connecting weight vector in the hidden layer. The weighted sum of j th neurons in hidden layer is as follows:

$$S_j^H(k) = \sum_{i=1}^{l} w_{ij}^I u_i(k) + \sum_{q=1}^{m} w_{qj}^{IR} x_q(k-1) + K \cdot S_j^H(k-1) \tag{6}$$

The j th neuron's output of hidden layer is as follow:

$$x_j(k) = f_N[S_j^H(k)] \tag{7}$$

Consider Fig. 3, $y_p(k)$ is p th output of output neuron for each discrete time k, $S_p^O(k)$ is the weighted sum of inputs and refractory input to p th output neuron in output layer, $x_j(k)$ is the output of j th neuron in hidden layer, K is refractory parameter of chaotic neuron, and $f_N(\cdot)$ is nonlinear sigmoid function. $\mathbf{W}^O$ and $\mathbf{W}^{OR}$ represent the weight vector between hidden and output and inter connecting weight vector in the output layer. The weighted sum of p th neurons in hidden layer is as follows:

$$S_p^O(k) = \sum_{j=1}^{m} w_{jp}^O x_j(k) + \sum_{r=1}^{n} w_{rp}^{OR} y_r(k-1) + KS_p^O(k-1) \tag{8}$$

$$y_p(k) = f_N[S_p^O(k)] \tag{9}$$

Using Eq. (6)(7), the weighted sum of neuron in output layer(Eq. 8) can define as follows:

$$S_p^O(k) = \sum_{j=1}^{m} w_{jp}^O f_N[\sum_{i=1}^{l} w_{ij}^I u_i(k) + \sum_{q=1}^{m} w_{qj}^{IR} x_q(k-1) + KS_j^H(k-1)] + \sum_{r=1}^{n} w_{rp}^{OR} y_r(k-1) + KS_p^O(k-1) \tag{10}$$

$$= \sum_{j=1}^{m} w_{jp}^O f_N[\sum_{i=1}^{l} w_{ij}^I u_i(k) + \sum_{q=1}^{m} w_{qj}^{IR} f_N[\sum_{i'=1}^{l} u_{i'}(k-1) + \sum_{j=1}^{m} w_{jq}^{IR} x_j(k-2) + KS_q^H(k-2)]] \tag{11}$$

$$+ \sum_{r=1}^{n} w_{rp}^{OR} f_N[\sum_{j'=1}^{m} w_{j'r}^O x_{j'}(k-1) + \sum_{r'=1}^{n} w_{r'r}^{OR} y_{r'}(k-2) + kS_r^O(k-2)] + KS_p^O(k-1)$$

$$O_p(k) = NF(u(l), x(l), y(l);\ l \le k) \tag{12}$$

where $O_p(k)$ is the p th output of chaotic neural network, $NF(\cdot)$ is a nonlinear function which represents a nonlinear dynamic mapping chaotic neural networks.

This neural network model in Eq. (12) is a globally coupled with present and past inputs and outputs of all neurons. Therefore, this model could simulate any complex nonlinear dynamic system.

The dynamic learning process may be formulated as:

$$\boldsymbol{W}(k+1) = \boldsymbol{W}(k) - \eta \cdot \nabla_w \boldsymbol{E}(k) \tag{13}$$

where $W(k)$ is an estimated weight vector at time k and η is a step size parameter, which affects the rate of convergence of the weights during learning.

The error index $E(k)$ should be defined as

$$\boldsymbol{E}(k) = \frac{1}{2} \sum_{i=1}^{n} [y_i^d(k) - y_i^m(k)]^2 = \frac{1}{2} \sum_{i=1}^{n} e_i^2(k) \tag{14}$$

where $e_i(k) = y_i^d(k) - y_i^m(k)$ is a learning error of i th neuron between the desired and network output at time k.

The gradient of error index with respect to an arbitrary weight vector W is represented by

$$\nabla_w \boldsymbol{E}(k) = -e(k) \nabla_w \boldsymbol{y}^m(k) = -e(k) \nabla_w \boldsymbol{O}(k) \tag{15}$$

where $\boldsymbol{O}(k)$ is output vector of neural network, and $\boldsymbol{y}^m(k) = \boldsymbol{O}(k)$ in case simple identification task. The output gradient $\nabla_w \boldsymbol{O}(k)$ with respect to output weights, interconnecting of output, interconnecting of hidden, and input weight in Eq. (15) are given by

$$\frac{\partial \boldsymbol{O}(k)}{\partial w_{ij}^O} = f_N'[S_j^O(k)] A_{ij}^O(k) \tag{16}$$

$$\frac{\partial \boldsymbol{O}(k)}{\partial w_{ij}^{OR}} = f_N'[S_j^O(k)] A_{ij}^{OR}(k) \tag{17}$$

$$\frac{\partial \boldsymbol{O}(k)}{\partial w_{ij}^{IR}} = [\sum_{r=1}^{n} f_N'(S_r^O(k)) \cdot w_{jr}^O] \cdot f_N'(S_j^H(k)) \cdot A_{ij}^{IR}(k) \tag{18}$$

$$\frac{\partial \boldsymbol{O}(k)}{\partial w_{ij}^{I}} = [\sum_{r=1}^{n} f_N'(S_r^O(k)) \cdot w_{jr}^O] \cdot f_N'(S_j^H(k)) \cdot A_{ij}^{I}(k) \tag{19}$$

where

$$A_{ij}^O(k) = O_i(k) + [w_{jj}^{OR} f_N'(S_j^O(k-1)) + K] A_{ij}^O(k-1),\quad A_{ij}^O(0) = 0, \tag{20}$$

$$A_{ij}^{OR}(k) = O_i(k-1) + [w_{jj}^{OR} f_N'(S_j^O(k-1)) + K] A_{ij}^{OR}(k-1),$$

$$A_{ij}^{OR}(0)=0\,, \tag{21}$$

$$A_{ij}^{IR}(k)=x_i(k-1)+[w_{jj}^{IR}f_N'(S_j^H(k-1))+K]\cdot A_{ij}^{IR}(k-1)\,,$$

$$A_{ij}^{IR}(0)=0\,, \tag{22}$$

$$A_{ij}^{I}(k)=u_i(k)+[w_{jj}^{IR}f_N'(S_j^H(k-1))+K]\cdot A_{ij}^{I}(k-1)\,,$$

$$A_{ij}^{I}(0)=0\,, \tag{23}$$

where $f_N'(\cdot)$ is the differential of $f_N(\cdot)$.

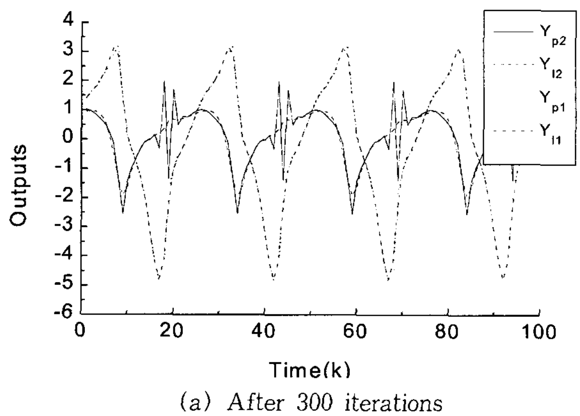
The globally asymptotical stability condition of MCNN is derived using M.Gupta's papers[8 10].

Example 1: This example is identify MIMO plant that is described by the equation

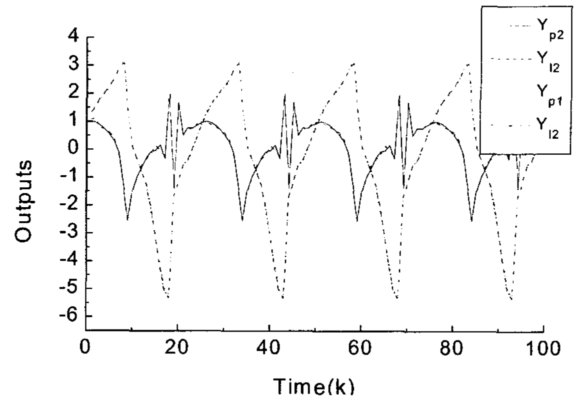
$$\begin{bmatrix} y_{p1}(k+1) \\ y_{p2}(k+1) \end{bmatrix}=\begin{bmatrix} \dfrac{y_{p1}(k)}{1+y_{p2}^2(k)} \\ \dfrac{y_{p1}(k)y_{p2}(k)}{1+y_{p2}^2(k)} \end{bmatrix}+\begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix} \tag{24}$$

where $\begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix}=\begin{bmatrix} \sin(2\pi k/25) \\ \cos(2\pi k/25) \end{bmatrix}$.

The results in Fig 4 show the identification results using the proposed dynamic backpropagation learning method in Eq (16) (23). In this example, one MCNN is used to identify the coupled nonlinear MIMO dynamics. The structure of MCNN consists of two inputs for $u_1(k)$, $u_2(k)$ respectively, 8 neurons in hidden layer, and 2 identified outputs for $y_{I1}(k)$, $y_{I2}(k)$. The learning rate is selected as 0.3, and refractory rate is 0.15.



(a) After 300 iterations



(b) After 2000 iterations

Fig. 4. Identified results of MIMO System

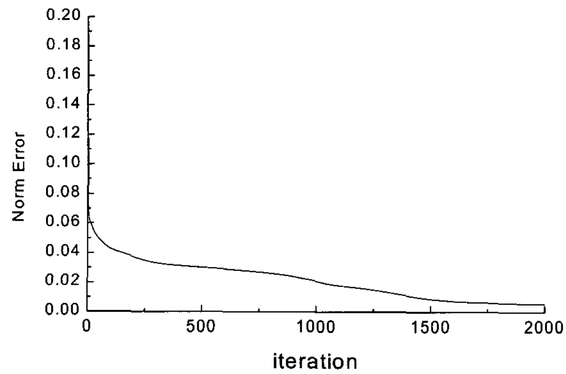


Fig. 5. Normalized Error with iterations

IV. Learning control scheme with MCNN

An approach for indirect adaptive control using modified chaotic neural networks is presented. The indirect adaptive control system consists of system identifier and controller. The system identifier with a chaotic neural network, called chaotic neural network identifier(CNNI), identifies an unknown plant for providing unknown plant information to the controller with a chaotic neural network. Both neural networks identifier and controller use dynamic backpropagation algorithm. In identifier, the generalized dynamic backpropagation algorithm could be adopted for adjusting weights of CNNI. In controller, the relationship between the activation value of plant and the plant output should be constructed for adjusting the weights of a chaotic neural network controller(CNNC) in figure 6.

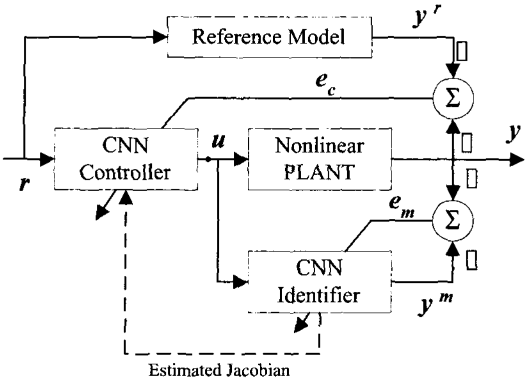


Fig. 6. Structure of Indirect Neuro Adaptive Controller

The gradient of error index with respect to an arbitrary weight vector W of controller should be redefined. The dynamic learning process for CNNC may be formulated as:

$$W(k+1) = W(k) - \eta_c \cdot \nabla_w E_c(k) \tag{25}$$

where $W(k)$ is an estimated weight vector for controller at time k, and η_c is a step size parameter for CNNC which affects the rate of convergence of the weights during learning. The error index $E_c(k)$ should be defined as:

$$E_c(k) = \frac{1}{2}\sum_{i=1}^{n}[y_i^r(k) - y_i(k)]^2 = \frac{1}{2}\sum_{i=1}^{n} e_{ci}^2(k) \tag{26}$$

where n is number of output in plant, and $e_{ci}(k) = y_i^r(k) - y_i(k)$ is a learning error between the reference model and the plant output at time k. The gradient of error index with respect to an arbitrary weight vector W is represented by

$$\nabla_W E_c(k) = -e_c(k)\nabla_W y(k) = -e_c(k)\nabla_{u(k)}y(k)\nabla_W u(k) = -e_c(k)\nabla_{u(k)}y(k)\nabla_W O^c(k) \tag{27}$$

where $e_c(k)$ is learning error vector at time k, and the plant input vector $u(k)$ is defined as the output vector of CNNC $O^c(k)$.

Since the plant is normally unknown, the sensitivity term $\nabla_{u(k)}y(k)$ could not be defined. After sufficient learning procedure, the learning error of CNNI could approximate to zero. Progressing the learning procedure of CNNI, the outputs of CNNI is close to the plant output, i.e., $y(k) \approx y^m(k)$. The sensitivity term could be redefined as

$$\nabla_{u(k)}y(k) \approx \nabla_{u(k)}y^m(k) = \nabla_{u(k)}O(k) \tag{28}$$

where $y^m(k) \equiv O(k)$ and $\nabla_{u(k)}O(k) \equiv \frac{\partial O(k)}{\partial u(k)}$.

The jacobian matrix could be defined as

$$J_{ij}(k) = \frac{\partial O_i(k)}{\partial u_j(k)} = \frac{\partial O_i(k)}{\partial S_i^O(k)} \sum_{p=1}^{m} \frac{\partial S_i^O(k)}{\partial x_p(k)} \frac{\partial x_p(k)}{\partial S_p^H(k)} \frac{\partial S_p^H(k)}{\partial u_j(k)} \tag{29}$$

where $J_{ij}(k)$ is an element of jacobian matrix which represents the sensitivity of plant output for input. Consider Eq. (6) (9), the partial derivatives can be defined as

$$\frac{\partial S_i^O(k)}{\partial x_p(k)} = w_{pi}^O, \quad \frac{\partial S_p^H(k)}{\partial u_j(k)} = w_{jp}^I.$$

$$J_{ij}(k) = \frac{\partial O_i(k)}{\partial u_j(k)} = f_N'(S_i^O(k)) \sum_{p=1}^{m} w_{pi}^O f_N'(S_p^H(k)) w_{jp}^I \tag{30}$$

Eq. (27) could be redefined as

$$\nabla_{W^c} E_c(k) = -e_c(k) J(k) \nabla_{W^c} O^c(k) \tag{31}$$

Using negative gradient in (31), the weights for CNNC can be adjusted in eq. (25). The equations (16) (23) define the dynamic backpropagation learning algorithms for CNNC.

Example 2: The plant is described by the difference equation[6]

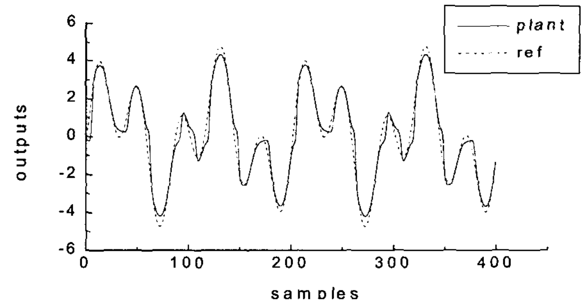
$$y^p(k+1) = \frac{y^P(k)}{1 + y^p(k)^2} + u^3(k) \tag{32}$$

The reference model is described by difference equation

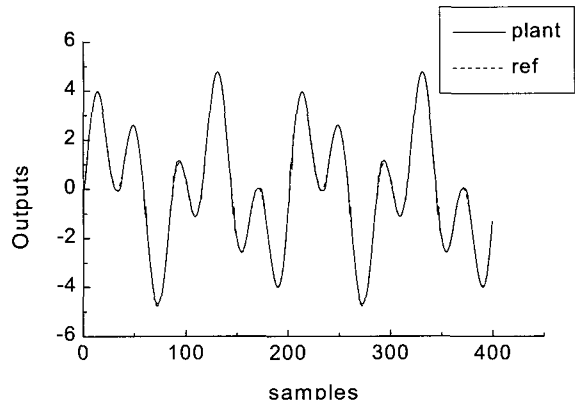
$$y^m(k+1) = 0.6y^m(k) + r(k) \tag{33}$$

where

$r(k) = \sin(2\pi k/25) + \sin(2\pi k/10)$.



(a) After 10 iterations



(b) After 1000 iterations

Fig. 7. Response of Controller

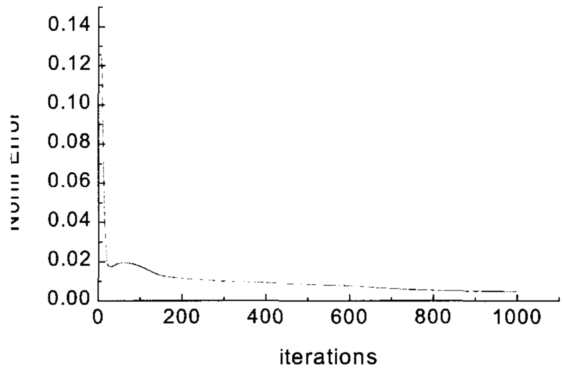


Fig. 8. Normalized Error with iterations

The results in Fig. 7 and 8 show the indirect adaptive neuro control results using the proposed dynamic backpropagation learning method in Eq. (16) (23) for CNNI and Eq.(30) and (31) for CNNC. The structure of MCNNI consists of two inputs, 7 neurons in hidden layer, and 2 outputs, and MCNNC has also same structure. The learning rates are selected as 0.3, and refractory rate is 0.15 for the CNNI and CNNC. Since MCNN has fast adapting characteristics, the CNNI identifies the plant model as on line learning method.

V. Conclusion

In this paper, we presents an indirect neuro adaptive controller which consists of two MCNNs: a MCNNI and a MCNNC. Traditional CNN was modified to simplify the model and to enforce the dynamic characteristics. We also modified the learning rule with dynamic backpropagation algorithm for the proposed MCNN. The performance of MCNN was tested for two examples: one of them was a nonlinear MIMO system identification and, the second was an indirect neuro adaptive controller. The simulation results show good performances, since the MCNN has the robust adaptability to nonlinear dynamic system.

VI. References

[1] Caianiello, E. R., DeLuca, A., "Decision equation for binary system: Application to neuronal behavior", Kybernetik, 3, 33 40, 1966.

[2] J. Nagumo and S. Sato, "On response characteristics of a mathematical neuron model," Kybernetik, vol. 10, pp. 155 164, 1972.

[3] L. Chen and K. Aihara, "Chaotic simulated annealing by a neural network model with transient chaos," Neural Networks, vol. 8, no. 6, pp. 915 930, 1995.

[4] L. Chen and K. Aihara, "Global Searching Ablity of Chaotic Neural Networks", IEEE Trans. Circuits and System I, vol 46, No 8, pp. 974 993, 1999.

[5] D.H.Nguyen and B.Widrow, " Neural Networks for Self Learning Control Systems", IEEE Control systems Magazine. Vol. 10, pp. 18 23. April. 1990

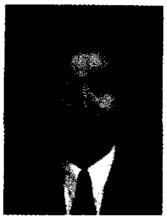
[6] K. S. Narendra and K. Parthasarathy, "Gradient methods for optimization of dynamical systems containing neural networks," IEEE Trans Neural Networks, vol. 2, no. 2, pp 252 262, 1991.

[7] S. H. Kim, C. W. Jang, "Trajectory Control of Robotic Manipulators using Chaotic Neural Networks", ICNN Vol. 3, pp1685 1688, 1997

[8] H. K. Khalil, "Nonlinear System", Prentice Hall.

[9] L. Jin, N. Nikiforuk, and M. Gupta, "Absolute Stability Conditions for Discrete Time Recurrent Neural Networks", IEEE Trans. Neural Networks, Vol. 5, No. 6, pp. 954 964, 1994.

[10] L. Jin and M. Gupta, "Stable Dynamic Backpropagation Learning in Recurrent Neural Networks", IEEE Trans. Neural Networks, Vol. 10, No. 5, pp. 1321 1334, 2000.



김 상 희(Sang Hee Kim)
正會員
1983년 홍익대학교 전기공학과(공학사)
1985년 연세대학교 전기공학과 (공학석사)
1992년 TEXAS A&M Univ (공학박사)
1993년3월～현재 금오공과대학교
전자공학부 부교수
관심분야: 신경회로망, 계측제어, 생체인식

박 원 우(Won Woo Park)
正會員
1773년 서강대학교 전자공학과 (공학사)
1983년 The Univ. of Texas at Austin
전기공학과 (공학석사)
1986년 The Univ. of Texas at Austin
(공학박사)
1993년3월～현재 금오공과대학교 전자공학부 부교수
관심분야: 컴퓨터 구조, 컴퓨터 그래픽스, 인공지능

안 희 욱(Hee Wook Ahn)
正會員
1982년 서울대학교 전기공학과 (공학사)
1991년 한국과학기술원 전기전자공학과
(공학석사)
1997년 - 한국과학기술원 (공학박사)
1982년 ~1998년 산업 기술 시험원(선임연구원)
관심분야: 컨버터 회로 및 제어, 모터제어