Adaptive Control and Identification Using One Neural Network for a Class of Plants with Uncertainties

Toshio Tsuji, Bing Hong Xu, and Makoto Kaneko

***Abstract*—This paper proposes a new neural adaptive control method that can perform adaptive control and identification for a class of controlled plants with linear and nonlinear uncertainties. This method uses a single neural network for both control and identification, and a sufficient condition of the local asymptotic stability is derived. Then, in order to illustrate the applicability of the proposed method, it is applied to the torque control of a flexible beam that includes linear and nonlinear structural uncertainties.**

I. Introduction

In recent years, applications of the neural network to adaptive control have been intensively conducted. For example, Narendra and Parthasarathy [1] introduced multilayer neural networks for identification and adaptive control of nonlinear systems. A number of studies such as [2]–[5] for adaptive control of unknown feedback linearizable systems and [6]–[8] for achieving guaranteed performance of the neuralnet controller have been reported. This is due to the fact that the neural network has excellent capabilities of nonlinear mapping, learning ability, and parallel computations. Most of the proposed adaptive control methods using single neural network can be roughly classified into four types: the direct neural adaptive control [9], [10], the parallel neural adaptive control [11], [12], the feedforward neural adaptive control [13], and the self-tuning neural adaptive control [14], [15].

Yabuta and Yamada [9] proposed the direct neural adaptive control that replaces a conventional feedback controller with a neural network. Also, they discussed the stability of the linear discrete-time single-input-single-output (SISO) plant [10]. Although their method is quite simple and can be applied to various feedback control systems, the uncertainty of the controlled plant cannot be identified and some parameters included in the neural network is quite difficult to be set. On the other hand, as the parallel neural adaptive control, Kraft and Campagne [11] and Sadegh [12] presented an adaptive controller based on the neural network arranged in parallel with a conventional feedback controller. Their idea is to compensate the control input computed from the conventional feedback controller for canceling the effect of the plant uncertainties. Also, Carelli *et al.* [13] proposed an adaptive controller using the feedback error learning [16]. The neural network can gradually modify the control input from a conventional feedback controller and can finally take the place of the conventional feedback controller. Moreover, Akhyar and Omatsu [14] and Khalid *et al.* [15] presented a self-tuning controller that uses a set of neural networks for regulating the gains of the conventional feedback controller in order to improve the performance of the control system. Their methods could maintain stability of the adaptive control system through the function of the conventional feedback controller.

For all of the methods presented above using single neural network, however, even if the adaptive controller of the controlled plant can be obtained by neural network learning, the uncertainty included in the controlled plant cannot be expressed explicitly. When the forward model for the controlled plant is necessary, the controlled plant must be identified again by using other identification techniques. Besides, many of these methods may require long learning time or result unstability in some practical applications, where there are uncertainties between the input to the controlled plant and the error signal required for learning.

Another approach to the neural adaptive control is to utilize multiple neural networks [1], [17]–[23]. In this approach, one neural network is dedicated to the forward model for identifying the uncertainties of the controlled plant and the other neural networks may compensate for the effect of the uncertainties based on the trained forward model. However, multiple neural networks must be trained and stability of this approach is quite difficult to be assured.

In this paper, a new neural adaptive control method that can simultaneously perform adaptive control and identification using only one neural network is proposed. In the proposed method, an identification model is composed of a neural network and a linear nominal model which is approximated for the controlled plant. The neural network can identify the uncertainties included in the controlled plant and can adaptively modify the control input computed from a predesigned conventional feedback controller at the same time. The neural network is of the multilayer perceptron, where the weight's updating rule is the error back-propagation using an identification error between the model output and the controlled plant's output.

This paper is organized as follows. In Section II, a formulation of the controlled plant, a working principle of the proposed method, a model of a neural network, and a stability analysis are shown. In Section III, in order to illustrate the effectiveness of the proposed method, computer simulations for plant models with linear and nonlinear uncertainties are carried out. In Section IV, to illustrate the applicability of the proposed method, torque control experiments of a flexible beam are performed. Experimental results using the proposed method are compared with those of other neural adaptive control methods in order to make clear the distinctive feature of the proposed method. Finally, Section V concludes the paper.

II. Adaptive Control and Identification

One of the keys to developing the adaptive control using the neural network is a way to deal with the unknown nonlinear properties included in the plant using the neural network. In our approach, the nonlinear properties are, first, linearized using one of the conventional approximation technique, and then the nonlinear modeling error is identified by using the neural network. The same neural network is also used to cancel out the effects of the nonlinear modeling error on the controlled variable, so that the single neural network can achieve not only the identification of the uncertainties but the control of the plant. First, we derive the proposed method for a class of plants with linear uncertainty. Then we show that the proposed method is also effective for the plant with nonlinear uncertainty.

A. Plant Formulation

Let us consider a controlled plant with a multiplicative uncertainty described by [24]

$$y(k) = H(z^{-1})u(k) \tag{1}$$

$$H(z^{-1}) = H_n(z^{-1})[1 + \Delta_H(z^{-1})] \tag{2}$$

$$H_n(z^{-1}) = \frac{B_n(z^{-1})}{A_n(z^{-1})} \tag{3}$$

$$\Delta_H(z^{-1}) = \frac{\Delta_B(z^{-1})}{\Delta_A(z^{-1})} \tag{4}$$

Manuscript received January 6, 1997; revised November 8, 1997.

T. Tsuji and M. Kaneko are with Industrial and Systems Engineering, Hiroshima University, Higashi-Hiroshima 739, Japan.

B. H. Xu is with Yamamoto Electric Corporation, Sukagawa 962, Japan.

Publisher Item Identifier S 1083-4427(98)04350-1.

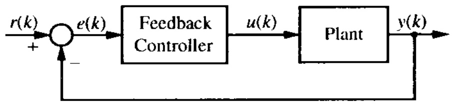


Fig. 1. Block diagram of feedback control system.

where $y(k), u(k), H(z^{-1})$, and $\Delta_H(z^{-1})$ are respectively the output, the input, the controlled plant model, and the multiplicative uncertainty. $H_n(z^{-1})$ is the known, controllable nominal model and $H_n(z^{-1}) \in RH_\infty$ is proper and stable [25]. Also, z^{-1} is the delay operator, and the polynomials $A_n(z^{-1}), B_n(z^{-1}), \Delta_A(z^{-1}), \Delta_B(z^{-1})$ are respectively given as

$$A_n(z^{-1}) = 1 + \sum_{j=1}^{n} a_j z^{-j} \tag{5}$$

$$B_n(z^{-1}) = \sum_{i=0}^{m} b_i z^{-i} \quad (n \geq m) \tag{6}$$

$$\Delta_A(z^{-1}) = 1 + \sum_{j=1}^{h} \alpha_j z^{-j} \tag{7}$$

$$\Delta_B(z^{-1}) = \sum_{i=0}^{l} \beta_i z^{-i} \quad (h \geq l). \tag{8}$$

Here, α_j, β_i are unknown coefficients and $l(\leq n), h(\leq m)$ are unknown orders of the polynomials $\Delta_A(z^{-1}), \Delta_B(z^{-1})$.

The general block diagram of the feedback control system is shown in Fig. 1, where $r(k)$ is the reference signal and $e(k) = r(k) - y(k)$ is the error between the reference signal and the plant output.

First, consider a special case in which there is no uncertainty in the plant (1), that is $\Delta_H(z^{-1}) = 0$. The conventional feedback controller $G_n(z^{-1})$ for the nominal model $H_n(z^{-1})$ can be predesigned to produce a desirable response. The closed loop transfer function $F_n(z^{-1})$ is described by [26]

$$F_n(z^{-1}) = \frac{y(k)}{r(k)} = \frac{G_n(z^{-1})H_n(z^{-1})}{1 + G_n(z^{-1})H_n(z^{-1})}. \tag{9}$$

Next, consider the general case of $\Delta_H(z^{-1}) \neq 0$ with the controller $G(z^{-1})$ for the plant $H(z^{-1})$ defined as

$$G(z^{-1}) = G_n(z^{-1})[1 + \Delta_G(z^{-1})] \tag{10}$$

where $\Delta_G(z^{-1})$ represents the modification of the controller $G(z^{-1})$. Thus, the closed loop transfer function $F(z^{-1})$ that consists of $H(z^{-1})$ and $G(z^{-1})$ can be given as

$$F(z^{-1}) = \frac{G(z^{-1})H(z^{-1})}{1 + G(z^{-1})H(z^{-1})}. \tag{11}$$

If (9) and (10) are equivalent, the response of $F(z^{-1})$ using $H(z^{-1})$ and $G(z^{-1})$ can agree with the desirable response [27]. Carrying out an operation using (4) and (9)–(11), we can obtain the following transformation for this equivalence:

$$\Delta_G(z^{-1}) = -\frac{\Delta_H(z^{-1})}{1 + \Delta_H(z^{-1})}. \tag{12}$$

However, since $\Delta_H(z^{-1})$ is unknown, the modified value $\Delta_G(z^{-1})$ cannot be computed by (12). When $\Delta_H(z^{-1})$ is over the admissible error's range of the feedback controller $G_n(z^{-1})$, the control system performance decreases, or yields a steady-state error, or even turns into unstable performance. In order to solve this control problem, in the next subsection we propose a new method that can identify the uncertainty $\Delta_H(z^{-1})$ using a neural network and adaptively modify the control input from the feedback controller $G_n(z^{-1})$.

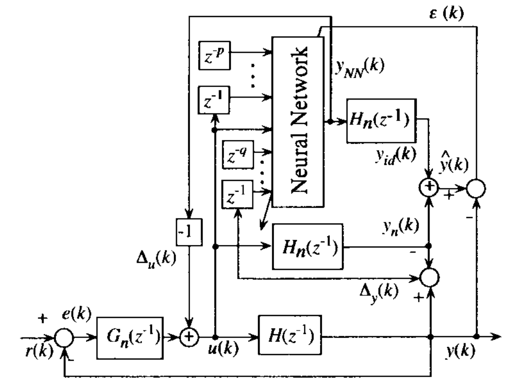


Fig. 2. Block diagram of the proposed method using one neural network.

B. Proposed Method

Fig. 2 shows the block diagram of the proposed method in this paper. The output $\hat{y}(k)$ of the identification model is a sum of the output of the nominal model $y_n(k)$ and the identified output $y_{id}(k)$ that is the output of the neural network $y_{NN}(k)$ passed through the nominal model $H_n(z^{-1})$. The neural network is trained using the identified error $\epsilon(k)$ between the model's output $\hat{y}(k)$ and the plant's output $y(k)$

$$\epsilon(k) = \hat{y}(k) - y(k). \tag{13}$$

The output of the neural network $y_{NN}(k)$ modifies the control input as

$$\Delta_u(k) = -y_{NN}(k). \tag{14}$$

Next, the working principle of the proposed method is explained. By using Fig. 1 and (10), the control input $u(k)$ can be represented as

$$\begin{aligned} u(k) &= G_n(z^{-1})[1 + \Delta_G(z^{-1})]e(k) \\ &= u_n(k) + \Delta_u(k) \end{aligned} \tag{15}$$

$$u_n(k) = G_n(z^{-1})e(k) \tag{16}$$

$$\Delta_u(k) = \Delta_G(z^{-1})u_n(k) \tag{17}$$

where $u_n(k)$ and $\Delta_u(k)$ are the nominal control input and the modification, respectively.

Also, from (1) and (2) the output $y(k)$ becomes

$$\begin{aligned} y(k) &= H_n(z^{-1})[1 + \Delta_H(z^{-1})]u(k) \\ &= y_n(k) + H_n(z^{-1})\Delta_y(k) \end{aligned} \tag{18}$$

$$y_n(k) = H_n(z^{-1})u(k) \tag{19}$$

$$\Delta_y(k) = \Delta_H(z^{-1})u(k) \tag{20}$$

where $\Delta_y(k)$ is the uncertain output via the uncertainty $\Delta_H(z^{-1})$.

Substituting (15), (17) into (20), we have

$$\begin{aligned} \Delta_y(k) &= \Delta_H(z^{-1})[u_n(k) + \Delta_u(k)] \\ &= \Delta_H(z^{-1})[1 + \Delta_G(z^{-1})]u_n(k). \end{aligned} \tag{21}$$

By (17) and (22), the modification $\Delta_u(k)$ can be rewritten as

$$\Delta_u(k) = \frac{\Delta_G(z^{-1})}{\Delta_H(z^{-1})[1 + \Delta_G(z^{-1})]}\,\Delta_y(k). \tag{22}$$

Substituting (12) into (22), we obtain the following relation between $\Delta_u(k)$ and the uncertain output $\Delta_y(k)$:

$$\Delta_u(k) = -\Delta_y(k). \tag{23}$$

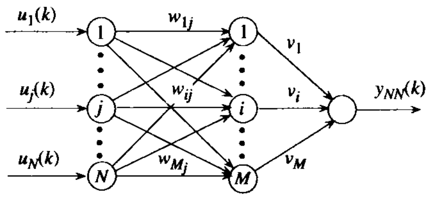


Fig. 3. Neural network used in the proposed method.

On the other hand, by Fig. 2, (13) and (18), the identified error $\epsilon(k)$ can be given as

$$\begin{aligned}\epsilon(k) &= [H_n(z^{-1})y_{NN}(k) - y_n(k)] \\ &\quad - [H_n(z^{-1})\Delta_y(k) + y_n(k)] \\ &= H_n(z^{-1})[y_{NN}(k) - \Delta_y(k)]. \end{aligned} \tag{24}$$

If the neural network is well trained, we can expect that $\epsilon(k)$ finally becomes zero in (24). Since $H_n(z^{-1})$ is the nominal model and is not identically zero, we can have

$$y_{NN}(k) = \Delta_y(k). \tag{25}$$

We can see that the output of the plant under the proposed method can agree with the desirable response using (14). As a result, the proposed method can adaptively control a class of plants with linear uncertainty given as (1)–(8) using the neural network. The proposed control system is designed to cancel out the effects of the second term in the right side of (18). Therefore, it should be noted that if the output $y(k)$ can be decomposed as (18) and the uncertainty $\Delta_y(k)$ can be identified by learning of the neural network, the proposed method is also valid for nonlinear uncertainties included in $\Delta_y(k)$. The effectiveness of the proposed method to the nonlinear uncertainties included in the plant will be verified in Sections III and IV. The next subsection explains how the modification $\Delta_u(k)$ in (14) can be realized using the neural network.

C. Neural Network Model

The multilayer neural network used in this paper is shown in Fig. 3. The numbers of units in the input layer and the hidden layer are N and M, respectively. The number of units of the output layer is one. In Fig. 3, $w_{ij}(k)$ represents the weight that connects the unit j in the input layer and the unit i in the hidden layer; $v_i(k)$ represents the weight that connects the unit i in the hidden layer and the output unit; $\boldsymbol{W}(k) \in \Re^{M\times N}, V(k) \in \Re^{M\times 1}$ are the weight matrix of the hidden layer and the weight vector of the output layer, respectively. From Fig. 2, the input vector to the neural network $U_{IN}(k) \in \Re^{N\times 1}$ is defined as

$$\begin{aligned}\boldsymbol{U}_{IN}^T(k) = [&u(k), u(k-1), \cdots, u(k-q) \\ &\Delta_y(k-1), \cdots, \Delta_y(k-p)]\end{aligned} \tag{26}$$

where $p \geq h, q \geq l, N = p + q + 1$.

Let the unit j's output of the input layer be denoted as $I_j = u_j(k)\ (j = 1, \cdots, N)$ and the unit i's output of the hidden layer be denoted as $H_i = \sigma(s_i)$, where $s_i = \Sigma_{j=1}^{N}\ w_{ij}I_j$ and $\sigma(x)$ is the sigmoid function defined as

$$\sigma(x) \equiv \frac{1}{\gamma}\ \tanh(\gamma x). \tag{27}$$

The positive parameter γ is related with the shape of the sigmoid function.

Fig. 4 shows the input–output relation of the sigmoid function. When $\gamma \leq 0.1, \sigma(x)$ can be approximated by a linear function. On the other hand, when $\gamma \geq 1, \sigma(x)$ has the form of the tanh function.

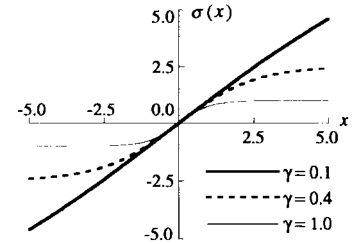


Fig. 4. Sigmoid function used in the neural network.

Moreover, the output of the output unit is denoted as $O_k = \sigma(\kappa)$, where $\kappa = \Sigma_{i=1}^{M}\ v_iH_i$.

Now, the energy function is defined as

$$\begin{aligned}J(k) &= \tfrac{1}{2}\ \epsilon^2(k) \\ &= \tfrac{1}{2}\ \{H_n(z^{-1})[y_{NN}(k) - \Delta_H(z^{-1})u(k)]\}^2. \end{aligned} \tag{28}$$

The energy function is minimized by changing the weights w_{ij} and v_i in the training process. According to the error back-propagation algorithm [28], the weight updating rules at one sampling time can be described as

$$\begin{aligned}\boldsymbol{V}(k+1) &= \boldsymbol{V}(k) - \eta\ \frac{\partial J(k)}{\partial V(k)} \\ &= \boldsymbol{V}(k) - \eta\epsilon(k)H_n(z^{-1})\ \frac{\partial y_{NN}(k)}{\partial \boldsymbol{V}(k)}\end{aligned} \tag{29}$$

and

$$\boldsymbol{W}(k+1) = \boldsymbol{W}(k) - \eta\epsilon(k)H_n(z^{-1})\ \frac{\partial y_{NN}(k)}{\partial \boldsymbol{W}(k)} \tag{30}$$

where $\eta > 0$ is the learning rate.

D. Stability Analysis

This subsection deals with the local asymptotic stability of the proposed method for the plant (1) near the optimal set of the neural network's weights. If the multilayer neural network is used, there exists the optimal set of the weights that results the identified error $\epsilon(k) = 0$ [29], [30].

Near the optimal set of the weights, $y_{NN}(k)$ is linearized by

$$y_{NN}(k) \approx \varrho\boldsymbol{V}^T(k)\boldsymbol{W}(k)\boldsymbol{U}_{IN}(k) \tag{31}$$

where $\varrho > 0$ is the gradient of the sigmoid function.

On the other hand, by (4), (7), (8), and (20), the uncertain output $\Delta_y(k)$ can be written as

$$\begin{aligned}\Delta_y(k) &= \left[\sum_{i=0}^{l}\ \beta_i z^{-i}\right] u(k) - \left[\sum_{j=1}^{h}\ \alpha_j z^{-j}\right] \Delta_y(k) \\ &= \boldsymbol{\theta}^T\boldsymbol{U}_{IN}(k)\end{aligned} \tag{32}$$

where

$$\begin{aligned}\boldsymbol{\theta} = [&\beta_0, \beta_1, \cdots, \beta_l, 0, \cdots, 0, \\ &-\alpha_1, \cdots, -\alpha_h, 0, \cdots, 0]^T \qquad \in \Re^{N\times 1}\end{aligned} \tag{33}$$

is the parameter vector. Thus, from (31), (32), the identified error $\epsilon(k)$ in (24) becomes

$$\epsilon(k) = H_n(z^{-1})\varphi^T(k)\boldsymbol{U}_{IN}(k) \tag{34}$$

where

$$\varphi^T(k) = \varrho\boldsymbol{V}^T(k)\boldsymbol{W}(k) - \boldsymbol{\theta}^T \qquad \in \Re^{1\times N} \tag{35}$$

is defined as the parameter error.

From Fig. 2 and (34), it can be seen that if the identified error $\epsilon(k)$ can be asymptotically stabilized, the asymptotic stability of the proposed method can be also guaranteed. Since the nominal model $H_n(z^{-1})$ is controllable and the control input $\boldsymbol{U}_{IN}(k)$ is bounded, the stability of the parameter error $\varphi(k)$ should be guaranteed in order to assure the stability of the identified error $\epsilon(k)$.

Now, let us consider a Lyapunov function $\Psi(k)$ of the following form:

$$\Psi(k) = \varphi^T(k)\varphi(k). \tag{36}$$

When the difference

$$\Delta\Psi = \Psi(k+1) - \Psi(k) < 0 \tag{37}$$

is held, the asymptotic stability of the parameter error $\varphi(k)$ can be guaranteed by the stipulations of the Lyapunov stability technique. If the neural network is trained until $\epsilon^2(k) \approx 0,$ the sufficient condition of the local asymptotic stability is to choose the learning rate η as

$$\frac{2}{\varrho\zeta\|Q(k)\|_\infty} > \eta > 0 \tag{38}$$

$$\zeta = \sup_{0 \le \omega \le \infty} |H_n(e^{-j\omega T})| \tag{39}$$

$$\|Q(k)\|_\infty = \sup_{0 \le k \le k_L} \overline{\sigma}\{\boldsymbol{Q}(k)\} \tag{40}$$

where k_L is the learning time, T is the sampling period, $\overline{\sigma}\{Q(k)\}$ (hereafter, abbreviated as $\overline{\sigma}(k)$) is the maximum singular value of the matrix $Q(k) \in \Re^{N \times N}$ given by

$$\begin{aligned}\boldsymbol{Q}(k) = &\boldsymbol{U}_{IN}(k)H_n(z^{-1})[\boldsymbol{U}_{IN}^T(k)\boldsymbol{W}^T(k)\Omega_1(k)\boldsymbol{W}(k)\\ &+ \boldsymbol{V}^T(k)\Omega_2(k)\boldsymbol{V}(k)\boldsymbol{U}_{IN}^T(k)]\end{aligned} \tag{41}$$

(see Appendix). The diagonal elements $\omega_{1ii}(k), \omega_{2ii}(k)$ of the diagonal matrices $\Omega_1(k), \Omega_2(k)$ are given as

$$\omega_{1ii}(k) = \frac{\sigma'(\kappa)\sigma(s_i)}{s_i} \qquad (\omega_{1ii}(k) = 0, \text{ if } s_i = 0) \tag{42}$$

$$\omega_{2ii}(k) = \sigma'(\kappa)\sigma'(s_i) \tag{43}$$

where $\sigma'(\cdot)$ is the derivative of $\sigma(\cdot)$. It can be easily seen that when the small positive learning rate is chosen, the condition (38) can be generally satisfied.

In this subsection, only the local asymptotic stability for the plant with linear uncertainty was discussed. If the uncertainty included in the plant is not limited to a linear one, the stability analysis might be very difficult to be done. Therefore, we will examine the stability of the proposed method by using computer simulations and experiments in Sections III and IV. Future research should be directed to the stability analysis of the proposed system including nonlinear uncertainties.

III. Computer Simulation

To illustrate the effectiveness of the proposed method, we use the simulated plants with linear and nonlinear uncertainties. The simulation results under the proposed method and the conventional feedback control method are compared.

The nominal model used in the computer simulation is

$$H_n(z^{-1}) = \frac{1}{1 + 4z^{-1} + 2.4z^{-2} + 0.448z^{-3} + 0.0256z^{-4}} \tag{44}$$

and the following feedback controller $G_n(z^{-1})$ is designed by using the pole-zero cancellation method [31]

$$G_n(z^{-1}) = \frac{1.889 + 7.131z^{-1} + 2.878z^{-2}}{z^{-1}}. \tag{45}$$

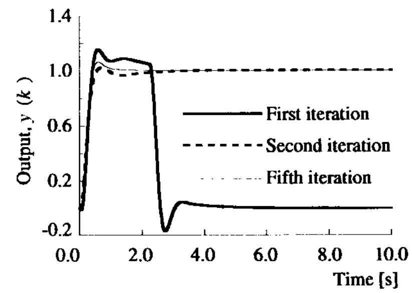


Fig. 5. Responses of the plant $H^1(z^{-1})$ by using the proposed method.

For the reference signal $r(k)$ of a unit step function and a rectangular function, the responses of the nominal model $H_n(z^{-1})$ of (44) using the control $G_n(z^{-1})$ of (45) are respectively shown in Fig. 7 and Fig. 9 as the desired response (DRE).

In the proposed method, $\gamma = 1$ in the sigmoid function (27) is used, and the weight's initial value of the neural network is chosen as the uniform random number in $[-2.0, +2.0]$. The learning rate is $\eta = 0.05$ and the sampling time is 10 ms. Also, because the order of the uncertainty $\Delta_H(z^{-1})$ is unknown, it is set as the maximum order, that is $h = n(=p), l = m(=q)$. This results $N = 5$ and $M = 5$ in Fig. 3.

The computer simulations performed in this section are divided into two parts: linear uncertainties and combined linear and nonlinear uncertainties.

A. Linear Uncertainties

The plant model

$$H^1(z^{-1}) = \frac{1.1}{1.2 + 1.1z^{-1} + 5.6z^{-2} + 0.48z^{-3} + 0.05z^{-4}} \tag{46}$$

was used with the reference signal of the unit step function. The simulation result under the proposed method is shown in Fig. 5. The response of the simulated plant is converging on the desired response in accordance with the learning of the neural network.

Fig. 6 shows the time history of the maximum singular value $\overline{\sigma}(k)$ in (40) during the first iteration of the neural network learning. It should be noted that other singular values of $Q(k)$ are always nonnegative during the simulation. Since the matrix $Q(k)$ includes the control input $U_{IN}(k), \overline{\sigma}(k)$ archives the largest value $\overline{\sigma}(k) = 19.92$ at the beginning of the control time where the control input for the step-like reference signal is rapidly changed. As the result, $\eta = 0.05$ satisfies the sufficient condition (38) and guarantees the local asymptotic stability of the proposed system. It can be seen that, if the learning rate η is chosen as a smaller positive number, the sufficient condition (38) can be generally satisfied for the local asymptotic stability. However, in order to make the learning more quickly, η should be large. The upper bound of the learning rate η in (38) is affected by randomly chosen initial weights and the control input, so that unfortunately, the stability cannot be checked beforehand. Future research should be directed to develop an adaptive regulation of η in order to speed up the learning.

On the other hand, Fig. 7 shows the desired response as the solid line, the response of the feedback control method by using $G_n(z^{-1}) = 1$ as the dotted line (abbreviated as FBC) and the response during the fifth learning iteration using the proposed method as the dashed line, respectively, where the proposed method is denoted as neuro-based adaptive control (NBAC). We can see that the response of the proposed method can almost achieve the desired response by only fifth learning iteration.

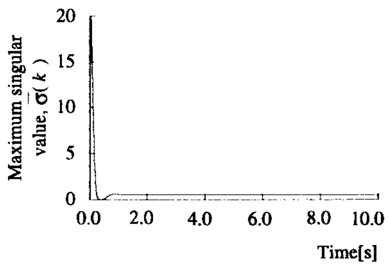


Fig. 6. Time history of the maximum singular value of $Q(k)$.

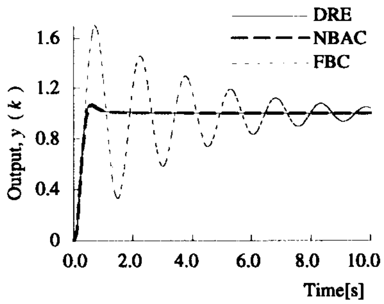


Fig. 7. Comparison of control results for the plant $H^1(z^{-1})$.

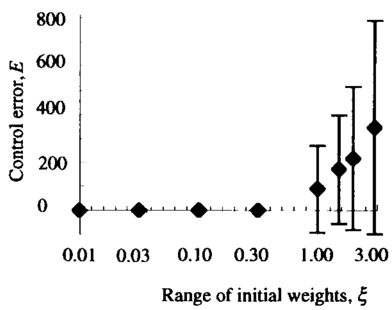


Fig. 8. Change of the control performance with the initial values of the neural networks weights.

The control performance using the neural network is closely related with the initial values of the network weights. When the neural network is not sufficiently trained, the local asymptotic stability may not be always guaranteed as shown in Section II-D. Therefore, we examine the relationship between the control performance and the initial values of the weights in the proposed method. In the simulation experiments, ten different sets of the initial values were chosen using the uniform random numbers within the range $[-\xi,\ \xi]$. The mean-square-error between the desired response and the plant output for the unit step reference signal are shown in Fig. 8 with the range ξ. As the range of the random numbers becomes large, the mean-square-error is increasing and the standard deviation is spreading. However, if smaller initial values are used, the mean-square-error converges almost zero.

Next, let us consider another plant model described as

$$H^2(z^{-1}) = \frac{1.45 + 0.25z^{-1} + 0.05z^{-2}}{0.75 + 3.2z^{-1} + 1.92z^{-2} + 0.35z^{-3} + 0.025z^{-4}}. \tag{47}$$

Fig. 9 shows the simulation results using the rectangular function as the reference signal. It should be noted that $H^2(z^{-1})$ includes not only the parameter uncertainties but also increase of the polynomial order of the denominator comparing to the nominal model $H_n(z^{-1})$. Thus, the response of the conventional feedback control (FBC) is oscillating, while the response of the proposed method almost agrees with the desired response after the fifth learning iteration.

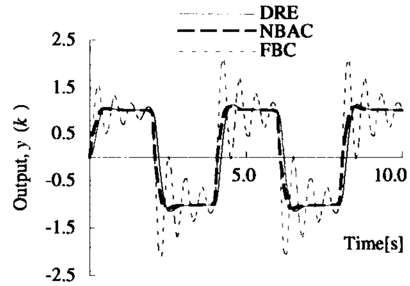


Fig. 9. Simulation results for the plant $H^2(z^{-1})$.

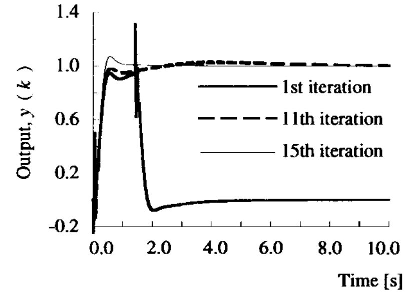


Fig. 10. Simulation results for the plant with linear and nonlinear uncertainties.

B. Linear and Nonlinear Uncertainties

Consider the following plant model with linear and nonlinear uncertainties:

$$y(t) = H^1(z^{-1})u(k) + \{1 - \exp[\pi u(k)]\} \tag{48}$$

where $\{1-\exp[\pi u(t)]\}$ is the nonlinear uncertainty. The parameter π represents the index of nonlinear extent and is set as $\pi = 0.01$ in this simulation. It should be noted that $H^1(z^{-1})$ includes the parameter perturbation shown in (46).

The proposed control method was applied to the plant model (48) with the reference signal of the unit step function and the feedback controller as (45). Fig. 10 shows the simulation results. The response during the 15 learning iterations almost agrees with the desired response shown in Fig. 10. Although the stability of the proposed system is proved to the plant with only linear uncertainty in Section II-D, it may be also effective to the class of plants with nonlinear uncertainty.

IV. Torque Control of a Flexible Beam

In this section, we apply the proposed method to a real control problem that is torque control of a flexible beam as shown in Fig. 11. While the flexible beam contacts with a fixed object, we would like to control the joint torque of the flexible beam in accordance with a reference signal. The contact point between the flexible beam and the fixed object can be detected by active motion of the joint [32]–[34]. Therefore, if the joint torque is controlled, the force applied to the fixed object can be also controlled. However, the dynamic characteristics of the flexible beam under consideration nonlinearly depend on the material and shape of the flexible beam, the external contact force, the contact friction and so on. Moreover, the rotational

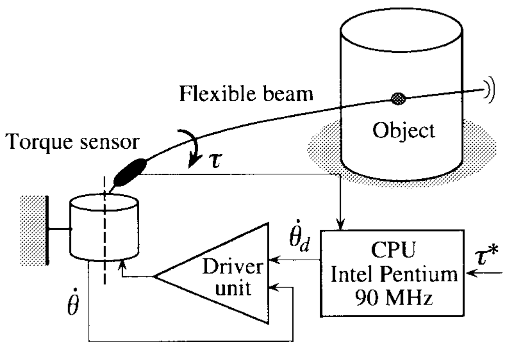


Fig. 11. Flexible beam in contact with an object.

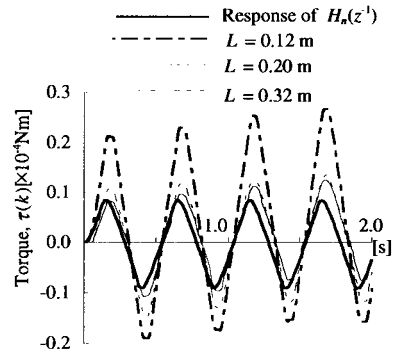


Fig. 12. Measured and predicted torque of the flexible beam.

stiffness of the beam is largely changed depending on the position of the contact point. When the distance from the joint to the contact point is small, the rotational stiffness is increased. When the contact point goes away from the joint, the joint becomes less stiffer. Thus, it is very difficult to obtain the exact dynamic model of the flexible beam beforehand, and a precise torque control of the flexible beam cannot be achieved by a conventional adaptive control technique.

Fukuda *et al.* [35] presented a method of adaptive force control of a rigid manipulator taking the object's characteristics into consideration, and also proposed a feedback decoupling control for suppressing vibrations in position control of a flexible robot arm [36]. Then, Tokita and Fukuda [37] presented an adaptive force control of a robotic manipulator using neural networks. Also, Takahashi [22] used an adaptive neural identifier and the direct neural controller [38] for controlling a flexible arm. The neural identifier can identify the parameter of the arm and the neural controller can work on the basis of the identified parameters. Moreover, using a linearized model of the flexible arm, robust control of the flexible arm has been actively studied by using the model matching control and H_∞ control methods [39], [40] in recent years.

In this section, the proposed method using one neural network is applied to the torque control of the flexible beam, and the control performance and identification ability of the proposed method are shown with the comparison of the experimental results using the model reference adaptive control and other neural adaptive control methods.

A. Experimental Device and Formulation

An experimental device for the torque control of the flexible beam is shown in Fig. 11 [34]. The beam is steel, 0.32 m in length and 0.8 mm in diameter. The torque sensor is made of a semiconductor gauge glued on an aluminum sheet. When the beam contacts with a fixed object, the torque $\tau(k)$ at the joint of the beam can be measured by the torque sensor. The actuator is velocity-controlled with the desired angular velocity $\dot{\theta}_d(k)$ of the joint being assigned by the computer. It should be noted that the driving torque of the actuator can not be controlled directly.

For this experimental device, first, let us consider a nonlinear plant described by

$$\tau(k) = H_n(z^{-1})u(k) + f(u(k)) \tag{49}$$

where $u(k)$ and $\tau(k)$ are the input to the actuator and the joint torque of the flexible beam, respectively; $f(u(k))$ represents the nonlinear uncertainty and unknown parameter's perturbation of this experimental device; and $H_n(z^{-1})$ represents the linear nominal model that is estimated from measured data by using a conventional identification technique. For the nonlinear function $f(u(k))$, it is assumed that its linear approximation is given by

$$f(u(k)) \approx H_n(z^{-1})\Delta_H(z^{-1})u(k) \tag{50}$$

where $\Delta_H(z^{-1})$ is the uncertainty. Then, (49) becomes the same form as the plant (1).

Next, the nominal model used in the proposed method is identified. The desired angular velocity $\dot{\theta}_d(k)$ is considered as the input to the flexible beam, so that the transfer function from $\dot{\theta}_d(k)$ to the torque $\tau(k)$ at the joint can be approximately described by

$$H_n(s) = \frac{K_s K_b}{s(t_s s + 1)} \tag{51}$$

where K_s is the gain, t_s is the time constant in the velocity-controlled system, and K_b is the elastic constant of the beam. The discrete form of (51) is given as

$$H_n(z^{-1}) = \frac{b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_1 z^{-2}}. \tag{52}$$

In order to identify the parameters of (52), the contact point $L = 0.20$ m was chosen and the beam was fixed to the environment. The rectangular input signal with its amplitude of 2.0×10^{-4} rad/s and a period of 0.5 s was used as $\dot{\theta}_d$ after passing through the first-order low-pass filter with a cutoff frequency 5 Hz. The joint torque was measured with the sampling frequency 100 Hz. The identified values of the model parameters were $\hat{a}_1 = -1.202\,96, \hat{a}_2 = 0.201\,21, \hat{b}_1 = 0.030\,63, \hat{b}_2 = 0.073\,41$ by using the least-squares method. The response of the nominal model with the identified parameters is shown in Fig. 12 as the thick line. From Fig. 12 we can see that the error between the output of $H_n(z^{-1})$ and the measured torque of the flexible beam with $L = 0.2$ m is increasing with time.

Using the same experimental device, the fixed position L of the beam was changed. The measured results are also shown in Fig. 12, where the alternate long and short dashed line represents the result with $L = 0.12$ m and the dashed line represents the result with $L = 0.32$ m. When the contact position L is varied, the joint torque becomes significantly different from the output of the nominal model with $L = 0.20$ m. In the next subsection, the proposed method using the identified parameters for $L = 0.20$ m is applied to the torque control of the flexible beam with different contact positions L.

B. Control Performance

In the neural network used in the experiment, the initial value of the weight was set as an uniform random number in $[-1.0 \times 10^{-3}, 1.0 \times 10^{-3}]$. The learning rate was $\eta = 0.05$ and the parameter γ of the sigmoid function was $\gamma = 1$. In order to cover the maximum order ($p = 2, q = 2$) of the uncertainties $\Delta_H(z^{-1})$, the neural network consisted of five units in the input layer, ten units in the hidden layer, and one unit in the output layer. Also, the reference signal $r(k)$ was

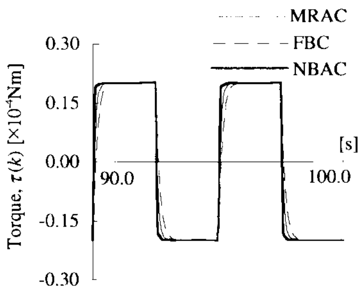
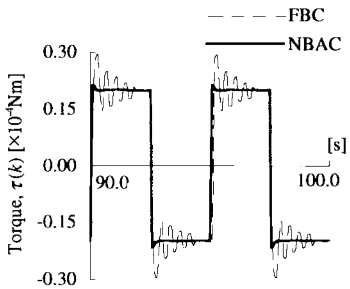
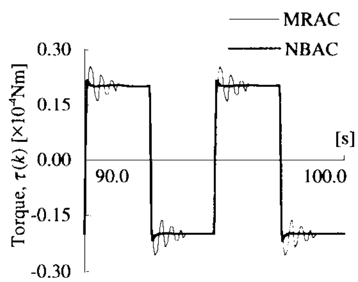


Fig. 13. Experimental results of the torque control of the flexible beam with $L = 0.20$ m.



(a)



(b)

Fig. 14. Experimental results of the torque control of the flexible beam with $L = 0.12$ m. (a) Proposed method and the feedback control using $G_n(z^{-1}) = 1$ and (b) proposed method and the model reference adaptive control.

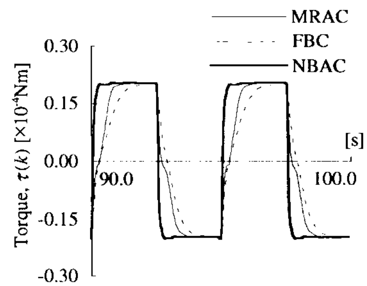


Fig. 15. Experimental results of the torque control of the flexible beam with $L = 0.32$ m.

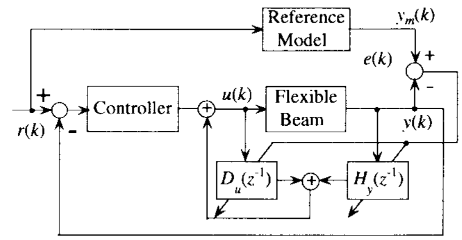


Fig. 16. MRAC for the torque control of the flexible beam.

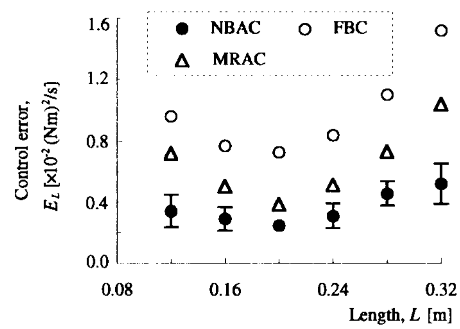


Fig. 17. Change of the control performance with the length of the beam.

of a rectangular form with its amplitude of 2.0×10^{-4} Nm and a period of 5 s. The feedback controller $G_n(z^{-1})$ was $G_n(z^{-1}) = 1$ and the control duration was 100 s. The proposed method was applied to 6 different contact points that were $L = 0.12$, 0.16, 0.20, 0.24, 0.28, 0.32 m.

Figs. 13–15 show the experimental results corresponding to the case of $L = 0.20$, 0.12, 0.32 m, respectively. In all cases, the dashed lines represent the results corresponding to the use of the feedback controller (FBC) $G_n(z^{-1})$, the thick lines the results by the proposed method (NBAC), the thin lines the results obtained with the model reference adaptive control (MRAC) of Fig. 16. In the MRAC, as the reference model and the controller, the nominal model $H_n(z^{-1})$ of (52) with the identified parameters ($L = 0.2$ m) and $G_n(z^{-1}) = 1$ were used, respectively, and the polynomials $D_u(z^{-1})$ and $H_y(z^{-1})$ of the compensators as shown in Fig. 16 were set as $D_u(z^{-1}) = 1 + d_{u1}z^{-1}$ and $H_y(z^{-1}) = h_0 + h_1 z^{-1}$, where d_{u1}, h_0 and h_1 are the updating coefficients at one sampling time [26], [41].

In Fig. 13, due to the fact that the same value of $L = 0.20$ m was used for identifying the nominal model, the experimental results obtained under three control methods were not obviously different.

However, when L is varied as shown in Figs. 14 and 15, the feedback control using $G_n(z^{-1})$ results significant overshoot or undershoot. The model reference adaptive control works well for linear parameter perturbation, so that it improves the control performance slightly. On the other hand, it can be seen from the experimental results of Figs. 14 and 15, the proposed method always produces stable responses. It should be noted that the identified parameters of the nominal model for $L = 0.20$ m were used for all cases.

Fig. 17 shows the mean-square-error

$$E_L = \frac{1}{10} \sum_{k=9000}^{10000} e^2(k) \tag{53}$$

between 90 and 100 s. In the figure, the white circle, the triangle and the black circle represent the results by the feedback control using $G_n(z^{-1})$, the model reference adaptive control and the proposed method, respectively. The errors corresponding to the proposed method are shown with their mean values and standard deviations for 10 different initial values of the weights. From the results shown in Fig. 17, we can see that even if there is a large error between the nominal model and the real dynamics of the flexible beam, the stable response is always obtained using the proposed method.

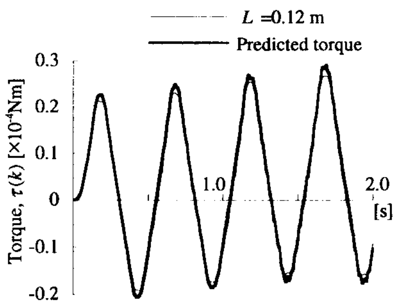


Fig. 18. Predicted torque by the proposed method.

C. Identification Ability

The another feature of the proposed method is that it can construct the identification model for the controlled plant (see $\widehat{y}(k)$ shown in Fig. 2). Let us investigate the identification ability of the proposed method for the contact point $L = 0.12$ m after learning of 100 s which is corresponding to Fig. 15.

Fig. 18 shows the identification model's output $\hat{y}(k)$ for the case of $L = 0.12$ m with the same input signal as shown in Fig. 12. It should be noted that the identified parameters for $L = 0.20$ m were used as the nominal model $H_n(z^{-1})$. We can see that the adaptive control and identification for the controlled plant can be achieved using only one neural network simultaneously.

D. Comparison

In this subsection, we make a comparison of the proposed method with other neural adaptive control methods, which are the self-tuning neural adaptive control (STNC) [14], [42] the feedforward neural adaptive control (FNAC) [13], [16] and the parallel neural adaptive control (PNAC) [11]. Fig. 19 shows three block diagrams of STNC, FNAC, and PNAC, respectively. It should be noted that the direct neural adaptive control did not result any stable learning in our experiments.

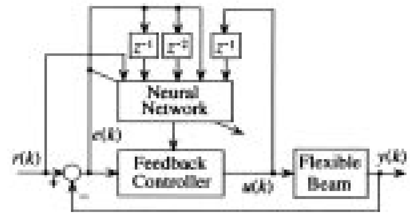
In the experiments, experimental conditions were the same as the ones described in the Section IV-B except for the control duration 60 s. The mean-square-error E_n during the one period of the reference signal, that is,

$$E_n = \frac{1}{5} \sum_{k=1}^{500} e^2[500(n-1)+k] \qquad (n = 1, 2, \cdots, 12) \tag{54}$$

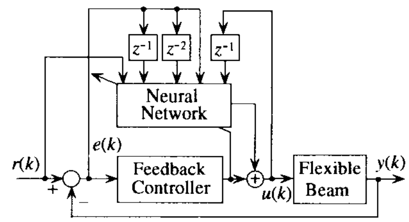
is computed for each control method. The sampling frequency was 100 Hz.

Fig. 20 shows the comparison of the learning history. From Fig. 20 the learning speed of the proposed method is faster than those of other control methods. STNC, PNAC, and FNAC need to learn the inverse model, while the proposed method requires to learn only the forward model of the uncertainty included in the controlled plant. Thus, the learning load of the proposed method is much less than the ones of other control methods.

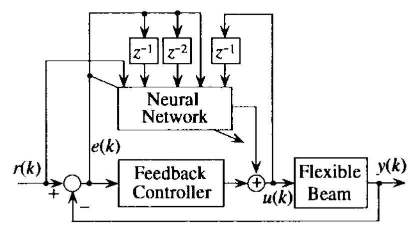
Next, the effect of the choice of the weight's initial value of the neural network used in adaptive control was examined. In the early stages of training, the neural network's output works as an undesirable disturbance to the controlled plant. If the weight's initial value that produces a large output of the neural network is set, it may happen that the control system becomes unstable and the learning of the neural network begins to diverge. Thus, the control performance of the neural adaptive control methods are examined for the range of the weight's initial value.



(a)



(b)



(c)

Fig. 19. Block diagrams of the adaptive control systems using neural networks. (a) Self-tuning neural adaptive control. (b) Feedforward neural adaptive control. (c) Parallel neural adaptive control.

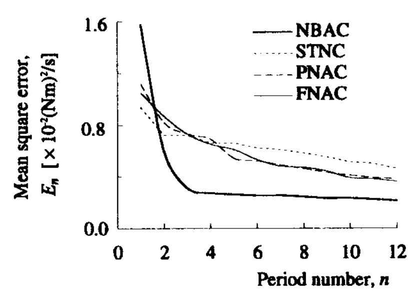


Fig. 20. Comparison of the learning history.

The range of the uniform random number $[-\xi, \xi]$ for the weight's initial value was changed and the mean-square-error $E = 1/2(E_{11} + E_{12})$ between 50 s and 60 s [see (54)] was computed as shown in Fig. 21. For larger values of the range of the initial weight ($\xi > 0.01$ for STNC, $\xi > 0.03$ for FNAC and $\xi > 0.1$ for PNAC), the control system became unstable and the neural network learning could not be converged. On the other hand, the proposed method always achieved stable learning of the neural network and adaptive control for the range $\xi \leq 3.0$.

V. Conclusion

In this paper, the new neural adaptive control method that can regulate the control input and identify the controlled plant with linear and nonlinear uncertainties by using only one neural network has been proposed. The working principle of the proposed method was explained and the sufficient condition of the local asymptotic stability near the optimal weight's set was derived. Computer simulations were

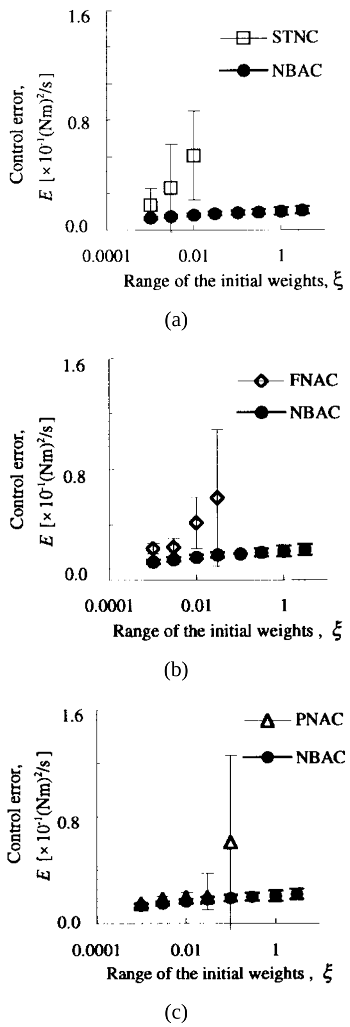


Fig. 21. Change of the control performance with the initial weights of the neural network. (a) NBAC and STNC, (b) NBAC and FNAC, and (c) NBAC and PNAC.

performed to the plant with linear and nonlinear uncertainties, so that the effectiveness and asymptotic stability of the proposed method were clearly confirmed. Also the proposed method was applied to the torque control of a flexible beam in contact with the external environment. Even though the dynamic characteristics of the flexible beam were largely varied, the precise control was realized in the experiment. The comparison of the experimental results under the proposed method and other neural adaptive control methods was done in order to show the distinctive feature of the proposed method.

In order to improve the performance of the controlled plant using the neural network, this paper have concentrated on the control structure of the controlled plant. The other way to improve the control performance is to revise the neural network model itself. In the future, we plan to improve the learning speed of the neural network and extend the proposed method to a general nonlinear plant and a multivariable system.

APPENDIX

Near the optimal set of the weights, the updating weight rules (29), (30) with the sigmoid function can be approximated as follows:

$$\boldsymbol{V}(k+1) \approx \boldsymbol{V}(k) - \eta\epsilon(k)\boldsymbol{H}_n(z^{-1})\boldsymbol{\Omega}_1(k)\boldsymbol{W}(k)\boldsymbol{U}_{IN}(k) \tag{55}$$

$$\boldsymbol{W}(k+1) \approx \boldsymbol{W}(k) - \eta\epsilon(k)\boldsymbol{H}_n(z^{-1})\boldsymbol{\Omega}_2(k)\boldsymbol{V}(k)\boldsymbol{U}_{IN}^T(k). \tag{56}$$

The diagonal elements of the matrices $\boldsymbol{\Omega}_1(k)$ and $\boldsymbol{\Omega}_2(k)$ are $|\omega_{1ii}(k)| \le M_1, |\omega_{2ii}(k)| \le M_2$ where M_1, M_2 are constants, since the sigmoid function is of the tanh function. Therefore, $\boldsymbol{\Omega}_1(k), \Omega_2(k)$ are bounded matrices.

By (55) and (56), assuming the identified error to be sufficiently small, we get

$$\begin{aligned}\boldsymbol{V}^T&(k+1)W(k+1)\\ &= [\boldsymbol{V}^T(k) - \eta\epsilon(k)\boldsymbol{H}_n(z^{-1})\boldsymbol{U}_{IN}^T(k)\boldsymbol{W}^T(k)\boldsymbol{\Omega}_1(k)]\\ &\quad\cdot[\boldsymbol{W}(k) - \eta\epsilon(k)\boldsymbol{H}_n(z^{-1})\boldsymbol{\Omega}_2(k)\boldsymbol{V}(k)\boldsymbol{U}_{IN}^T(k)]\\ &\approx \boldsymbol{V}^T(k)\boldsymbol{W}(k) - \eta\epsilon(k)\boldsymbol{H}_n(z^{-1})\\ &\quad\cdot[\boldsymbol{U}_{IN}^T(k)\boldsymbol{W}^T(k)\boldsymbol{\Omega}_1(k)\boldsymbol{W}(k)\\ &\quad+\boldsymbol{V}^T(k)\boldsymbol{\Omega}_2(k)\boldsymbol{V}(k)\boldsymbol{U}_{IN}^T(k)].\end{aligned} \tag{57}$$

Substituting (34) into the parameter error (57) yields

$$\begin{aligned}\boldsymbol{V}^T&(k+1)W(k+1)\\ &\approx \boldsymbol{V}^T(k)W(k) - \eta\boldsymbol{H}_n(z^{-1})\varphi^T(k)\boldsymbol{U}_{IN}(k)\boldsymbol{H}_n(z^{-1})\\ &\quad\cdot[\boldsymbol{U}_{IN}^T(k)\boldsymbol{W}^T(k)\boldsymbol{\Omega}_1(k)\boldsymbol{W}(k)\\ &\quad+\boldsymbol{V}^T(k)\boldsymbol{\Omega}_2(k)\boldsymbol{V}(k)\boldsymbol{U}_{IN}^T(k)]\\ &= \boldsymbol{V}^T(k)\boldsymbol{W}(k) - \eta H_n(z^{-1})\varphi^T(k)\boldsymbol{Q}(k).\end{aligned} \tag{58}$$

Then substituting (58) into (35) yields

$$\begin{aligned}\varphi^T(k+1) &= \varrho\boldsymbol{V}^T(k+1)\boldsymbol{W}(k+1) - \boldsymbol{\theta}^T\\ &= \varphi^T(k)[\boldsymbol{\Gamma} - \eta\varrho\boldsymbol{H}_n(z^{-1})Q(k)]\end{aligned} \tag{59}$$

where $\boldsymbol{\Gamma} \in \Re^{N\times N}$ is the unit matrix. So, the difference $\Delta\Psi$ can be written as

$$\begin{aligned}\Delta\Psi &= \Psi(k+1) - \Psi(k)\\ &= \varphi^T(k)[\boldsymbol{\Gamma} - \eta\varrho\boldsymbol{H}_n(z^{-1})Q(k)]\\ &\quad\cdot[\boldsymbol{\Gamma} - \eta\varrho\boldsymbol{H}_n(z^{-1})\boldsymbol{Q}^T(k)]\varphi(k) - \varphi^T(k)\varphi(k)\\ &= -\eta\varrho\boldsymbol{H}_n(z^{-1})\varphi^T(k)[\boldsymbol{Q}(k) + \boldsymbol{Q}^T(k)\\ &\quad - \eta\varrho\boldsymbol{H}_n(z^{-1})\boldsymbol{Q}(k)\boldsymbol{Q}^T(k)]\varphi(k).\end{aligned} \tag{60}$$

When the learning rate η satisfies to the following expression

$$\boldsymbol{Q}(k) + \boldsymbol{Q}^T(k) - \eta\varrho\boldsymbol{H}_n(z^{-1})\boldsymbol{Q}(k)\boldsymbol{Q}^T(k) > 0 \tag{61}$$

the condition of $\Delta\Psi < 0$ can be guaranteed.

Next, we derive the condition that the learning rate η satisfies (61). Using the matrix norm, (61) becomes

$$\|\boldsymbol{Q}(k) + \boldsymbol{Q}^T(k)\|_\infty > \eta\varrho\|\boldsymbol{H}_n(z^{-1})Q(k)Q^T(k)\|_\infty. \tag{62}$$

Defining $\boldsymbol{Q}(k)$ to be a positive semi-definite matrix yields [43]

$$\begin{aligned}\|\boldsymbol{Q}(k) + Q^T(k)\|_\infty &= \|\boldsymbol{Q}(k)\|_\infty + \|\boldsymbol{Q}^T(k)\|_\infty\\ &= 2\|\boldsymbol{Q}(k)\|_\infty.\end{aligned} \tag{63}$$

Finally, (62) can be represented as

$$2\|\boldsymbol{Q}(k)\|_\infty > \eta\varrho\zeta\{\|\boldsymbol{Q}(k)\|_\infty\}^2 \tag{64}$$

that is

$$\frac{2}{\varrho\zeta\|\boldsymbol{Q}(k)\|_\infty} > \eta > 0 \tag{65}$$

where

$$\zeta = \sup_{0\le\omega\le\infty} |\boldsymbol{H}_n(e^{-j\omega T})|. \tag{66}$$

REFERENCES

[1] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Trans. Neural Networks*, vol. 1, no. 2, pp. 4–27, 1990.

[2] F.-C. Chen and H. K. Khalil, "Adaptive control of nonlinear systems using neural networks," *Int. J. Contr.*, vol. 55, no. 6, pp. 1299–1317, 1992.

[3] C.-C. Liu and F.-C. Chen, "Adaptive control of nonlinear continuous-time systems using neural networks–general relative degree and mimo cases," *Int. J. Contr.*, vol. 58, no. 2, pp. 317–335, 1993.

[4] F.-C. Chen and C.-C. Liu, "Adaptively controlling nonlinear continuous-time systems using multilayer neural networks," *IEEE Trans. Automat. Contr.*, vol. 39, no. 6, pp. 1306–1310, 1994.

[5] F.-C. Chen and H. K. Khalil, "Adaptive control of a class of nonlinear discrete-time systems using neural networks," *IEEE Trans. Automat. Contr.*, vol. 40, no. 5, pp. 791–801, 1995.

[6] F. L. Lewis, K. Liu, and A. Yesildirek, "Neural net robot controller with guaranteed tracking performance," *IEEE Trans. Neural Networks*, vol. 6, no. 3, pp. 703–715, 1995.

[7] S. Jagannathan and F. L. Lewis, "Multilayer discrete-time neural-net controller with guaranteed performance," *IEEE Trans. Neural Networks*, vol. 7, no. 1, pp. 107–130, 1996.

[8] F. L. Lewis, A. Yesildirek, and K. Liu, "Multilayer neural-net robot controller with guaranteed tracking performance," *IEEE Trans. Neural Networks*, vol. 7, no. 2, pp. 388–399, 1996.

[9] T. Yabuta and T. Yamada, "Neural network controller characteristics with regard to adaptive control," *IEEE Trans. Syst., Man, Cybern.*, vol. 22, no. 1, pp. 170–176, 1992.

[10] T. Yamada and T. Yabuta, "Some remarks on characteristics of direct neuro-controller with regard to adaptive control," *Trans. Soc. Instrum. Contr. Eng.*, vol. 27, no. 4, pp. 784–791, 1991.

[11] L. G. Kraft, III, and D. P. Campagna, "A summary comparison of cmac neural network and traditional adaptive control systems," in *Neural Networks for Control*, W. T. Miller, R. S. Sutton, and P. J. Werbos, Eds. Cambridge, MA: MIT Press, 1990.

[12] N. Sadegh, "A perceptron network for functional identification and control of nonlinear systems," *IEEE Trans. Neural Networks*, vol. 4, no. 6, pp. 982–988, 1993.

[13] R. Carelli, E. F. Camacho, and D. Patino, "A neural network based feedforward adaptive controller for robots," *IEEE Trans. Syst., Man, Cybern.*, vol. 25, pp. 1281–1288, May 1995.

[14] S. Akhyar and S. Omatsu, "Self-tuning pid control by neural networks," in *Proc. Int. Joint Conf. Neural Networks*, 1993, pp. 2749–2752.

[15] M. Khalid, S.Omatu, and R. Yusof, "Temperature regulation with neural networks and alternative control schemes," *IEEE Trans. Neural Networks*, vol. 6, no. 3, pp. 572–582, 1995.

[16] M. Kawato, Y. Uno, M. Isobe, and R. Suzuki, "A hierarchical model for voluntary movement and its application to robotics," *IEEE Contr. Syst. Mag.*, vol. 8, no. 2, pp. 8–16, 1988.

[17] Y. Iiguni, H. Sakai, and H. Tokumaru, "A nonlinear regulator design in the presence of system uncertainties using multilayered neural network," *IEEE Trans. Neural Networks*, vol. 2, no. 4, pp. 410–417, 1991.

[18] C. Ku and K. Y. Lee, "Diagonal recurrent neural networks for dynamic systems control," *IEEE Trans. Neural Networks*, vol. 6, no. 1, pp. 144–156, 1995.

[19] M. M. Polycarpou and A. J. Helmicki, "Automated fault detection and accommodation: A learning systems approach," *IEEE Trans. Syst., Man, Cybern.*, vol. 25, pp. 1447–1458, Feb. 1995.

[20] G. A. Rovithakis and M. A. Christodoulou, "Direct adaptive regulation of unknown nonlinear dynamical systems via dynamic neural networks," *IEEE Trans. Syst., Man, Cybern.*, vol. 25, pp. 1578–1594, Dec. 1995.

[21] A. U. Levin and K. S. Narendra, "Control of nonlinear dynamical systems using neural networks–part ii: Observability, identification, and control," *IEEE Trans. Neural Networks*, vol. 7, no. 1, pp. 30–42, 1996.

[22] K. Takahashi, "Neural-network-based learning control applied to a single-link flexible arm," in *Proc. 2nd Int. Conf. Motion Vibration Control*, 1994, pp. 811–816.

[23] G. A. Rovithakis and M. A. Christodoulou, "Adaptive control of unknown plants using dynamical neural networks," *IEEE Trans. Syst., Man, Cybern.*, vol. 24, pp. 400–412, Mar. 1994.

[24] M. A. Kaashoek, J. H. van Schuppen, and A. C. M. Ran, *Robust Control of Linear Systems and Nonlinear Control*. Boston, MA: Birkhäuser, 1990.

[25] B. A. Francis, *A Course in H_∞ Control Theory*. New York: Springer-Verlag, 1987.

[26] K. J. Åströ and B. Wittenmark, *Adaptive Control*. Norwell, MA: Addison-Wesley, 1989.

[27] T. Kailath, *Linear Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1980.

[28] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by error propagation," in *Parallel Distributed Processing*, D. E. Rumelhart, J. L. McClelland, and PDP Research Group, Eds. Cambridge, MA: MIT Press, 1986, vol. 1, pp. 318–362.

[29] G. Cybenko, "Approximation by superposition of a sigmoidel function," in *Artificial Neural Networks: Concepts and Theory*, P. Mehra and B. W. Wah, Eds. Los Alamitos, CA: IEEE Comput. Soc. Press, 1992, pp. 488–499.

[30] K. Funahashi, "On the approximate realization of continuous mappings by neural networks," *Neural Networks*, vol. 2, pp. 183–192, 1989.

[31] R. N. Clark, *Control System Dynamic*. Cambridge, U.K.: Cambridge Univ. Press, 1996.

[32] M. Kaneko, "Active antenna," in *Proc. IEEE Int. Conf. Robotics Automation*, 1994, pp. 2665–2671.

[33] M. Kaneko, N. Kanayama, and T. Tsuji, "3-D active antenna for contact sensing," in *Proc. IEEE Int. Conf. Robotics Automation*, 1995, vol. 1, pp. 1113–1119.

[34] N. Ueno, M. Kaneko, and M. Svinin, "Theoretical and experimental investigation on dynamic active antenna," in *Proc. IEEE Int. Conf. Robotics Automation*, vol. 3, pp. 3557–3563, 1996.

[35] T. Fukuda, N. Kitamura, and K. Tanie, "Adaptive force control in robot manipulation with consideration of characteristics of objects," *Trans. Jpn. Soc. Mech. Eng.*, vol. 53, no. 487, pp. 726–730, 1987 (in Japanese).

[36] T. Fukuda and A. Arakawa, "Control of flexible robotic arms," *Trans. Jpn. Soc. Mech. Eng.*, vol. 53, no. 488, pp. 954–961, 1987 (in Japanese).

[37] M. Tokita and T. Fukuda, "Force control of robotic manipulator using neural network," *J. Robot. Soc. Jpn.*, vol. 14, no. 1, pp. 75–82, 1996 (in Japanese).

[38] T. Yamada and T. Yabuta, "Nonlinear neural network controller for dynamic systems," in *Proc. 16th Annu. Conf. IEEE Industrial Electronics Soc.*, 1990, pp. 1244–1249.

[39] E. Rijanto, A. Moran, T. Kurihara, and M. Hayase, "Robust tracking control of flexible arm using inverse dynamics method," in *Proc. 4th Int. Workshop Advanced Motion Control*, 1996, pp. 669–674.

[40] J. Lu, M. Shafiq, and T. Yahagi, "Vibration control of flexible robotic arms using robust model matching control," in *Proc. 4th Int. Workshop Advanced Motion Control*, 1996, pp. 663–668.

[41] K. S. Narendra and A. M. Annaswamy, *Stable Adaptive Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[42] W. T. Miller, R. S. Sutton, and P. J. Werbos, *Neural Networks for Control*. Cambridge, MA: MIT Press, 1990.

[43] A. Weinmann, *Uncertain Models and Robust Control*. New York: Springer-Verlag, 1991.