A DESIGN OF PID CONTROLLERS FUSED CMACS WITH NEURAL NETWORKS

Kenji Takao * **Ryota Kurozumi** **
Toru Yamamoto ** **Takao Hinamoto** *

* *Graduate School of Engineering, Hiroshima University, 1-4-1 Kagamiyama, Higashi-hiroshima, Hiroshima, 739-8527, Japan*
** *Graduate School of Education, Hiroshima University 1-1-1 Kagamiyama, Higashi-hiroshima, Hiroshima, 739-8524, Japan*

Abstract: Several neural-net based PID controllers have been proposed for nonlinear process systems. However, they have been not so widely used in process industries due to the considerably computational cost. This paper presents a new intelligent PID tuning scheme, whose PID tuner is constructed by the fusional structure of a cerebellar model articulation controller and a neural network. This PID tuner gives us the higher learning efficiency which has not been realized by the conventional neural-net based controllers, and it enables us to tune PID gains in an on-line manner. The behaviour of the proposed scheme is examined by a simulation example for a chemical reactor model.

Keywords: PID Control, CMAC, Neural Network, Nonlinear Control, Process Control

1. INTRODUCTION

Neural-net based controllers have been proposed for nonlinear systems. The reason why the neural network(NN)(K.S.Narendra and K.Parthasarathy, 1990; C.-Y.Seong and B.Widrow, 2001) is employed for such systems is that they have the capability of highly approximating nonlinear properties. The back propagation(BP) scheme is usually employed in updating weights included in neural networks. According to the neural-net based controllers using the BP scheme, however, the problem is pointed out which considerably computational time is required until the good control performance is obtained. Therefore, the neural-net based controllers are not necessarily suitable in the real-time control.

On the other hand, PID control(J.G.Zieglar and N.B.Nichols, 1942; K.L.Chien and J.B.Reswick, 1972; K.J. Åström, 1988) schemes still continue to be widely used for most industrial control systems, particularly in the chemical process industry. This is mainly because PID controllers have simple control structures, and are simple to maintain and tune. Therefore, it is still attractive to design discrete-time control systems with PID control structures. Especially, some neural-net based PID controllers(S.Omatu *et al.*, 1995) have been considered for nonlinear systems in the reason that most real process systems have nonlinear properties. As mentioned above, however, since these controllers require much computational cost to training the neural networks, they have been hardly utilized in real process systems.

By the way, a cerebellar model articulation controller(CMAC) has been proposed by Albus (J.S.Albus, 1975*b*; J.S.Albus, 1975*a*), which is a kind of artificial neural networks, and especially applied to some robot control systems and chemical process systems. The weights included in the CMAC are trained based on a learning scheme that has the common memory reference structure. Therefore, since the learning effect effectively spreads around the learning point, it has the feature that learning cost is drastically reduced in comparison with the conventional BP method. However, it has also the following problems.

(1) Since the conventional CMAC directly generates the control input, it is difficult to explicitly grasp the nonlinear properties of the controlled object. It is a very important factor to know characteristics of the controlled object in operating equipments in real process systems.

(2) Since input signals for the CMAC are quantized, it is difficult to obtain accurate control performance even if it has the rapidity in the initial stage of training the CMAC. This problem can be solved by increasing the number of weight tables included in the CMAC. However, since the capacity of the computer memories becomes considerably large, it is not useful in the practical systems.

The objective of this paper is to propose a new design method of neural-net based intelligent PID controllers with high learning efficiency. The proposed controller has a fusional structure of the CMAC and the NN. The new scheme to generate PID parameters using the CMAC, is firstly proposed. As mentioned above, although it is impossible to know the properties of the controlled object using the conventional CMAC, the operators to deal with real systems can roughly grasp them through PID gains. To the best of our knowledge, such a controller that generates PID gains using the CMAC, has been not reported until now. Next, the fusional structure is considered, which is constructed using the CMAC and the NN. This structure can complement each problem in the CMAC and the NN mutually. In other words, the CMAC is suitable to work in the initial stage of learning, because it demonstrates effect in the static mapping. On the other hand, the accurate learning performance can be finally obtained using the NN, although the NN requires much learning time. Therefore, the CMAC is firstly worked in tuning PID gains, and the PID tuner which has been roughly trained, is switched from the CMAC to the NN. Then, the NN works so that the control performance is further improved. This structure gives us the higher learning efficiency which has not been realized by the conventional neural-net based controllers.

This paper is organized as follows. The intelligent PID controller with the fusional structure of the CMAC and the NN, is proposed. Next, the CMAC-based PID tuner firstly is explained, followed by the NN-based PID tuner. Furthermore, a criterion in order to switch from the CMAC to the NN is considered. Finally, the effectiveness of the proposed scheme is numerically evaluated on a simulation example based on a chemical reactor model.

2. PID CONTROLLER DESIGN

2.1 Outline of controller

The velocity-type PID control law which has variable gains, is described as

$$\Delta u(t) = K_I(t)e(t) - K_P(t)\Delta y(t) - K_D(t)\Delta^2 y(t) \quad (1)$$

$$\mathbf{K}(t) = [K_P(t), K_I(t), K_D(t)], \quad (2)$$

where $u(t)$ and $y(t)$ denote the control input and the corresponding system output, respectively. And also, $e(t)$ denotes the control error, which is defined as follows:

$$e(t) := r(t) - y(t), \quad (3)$$

where $r(t)$ denotes the reference signal, and $K_P(t)$, $K_I(t)$ and $K_D(t)$ are respectively the proportional gain, the integral gain and the derivative gain. Furthermore, Δ denotes the differencing operator defined as $\Delta := 1 - z^{-1}$.

The control performance strongly depends on PID gains in (1). Especially, PID gains must be carefully determined for nonlinear systems. Several neural-net based PID controllers have been proposed for such systems, in which PID gains adequately change corresponding to the operating points. However, it is pointed out that these schemes require much computational time until the good control performance is obtained.

On the other hand, the CMAC which is one of artificial neural networks, can roughly approximate the nonlinear components with quite small learning time, although the CMAC does not give an accurate control performance such as the NN.

Therefore, this paper newly presents an intelligent PID tuner which has the fusional structure of the CMAC and the NN as shown in Fig.1. Here, the **S.W** in this figure denotes the switching from the CMAC to the NN, and two **S.W**s work together. That is, the CMAC firstly works in the initial stage in order to roughly approximate the nonlinear components, and the PID tuner is switched from the CMAC to the NN in order to further improve the control performance. This structure

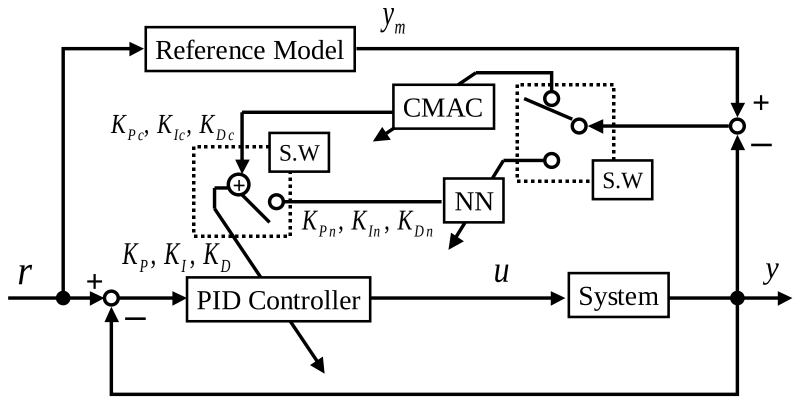

Fig. 1. Block diagram of the proposed controller.

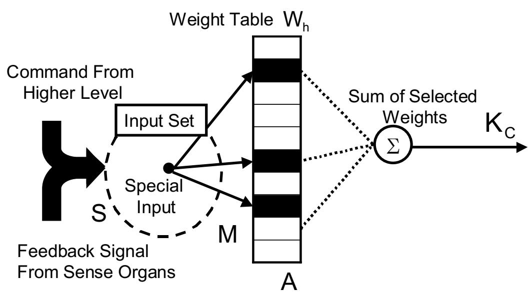

Fig. 2. Conceptual diagram of a CMAC.

gives us the high learning efficiency, because a good control performance can be obtained with quite small learning time. The detailed tuning scheme is discussed below.

2.2 PID parameter tuning by CMAC

The CMAC is one of artificial neural networks, and is a controller for the body motion system. The conceptual figure of the CMAC is shown in Fig.2. The CMAC is defined as the following mapping relation:

$$S \to M \to A \to \mathbf{K_c}, \tag{4}$$

where S, M, A and $\mathbf{K_c}$ denote the input vectors, the intermediate variables to code S, the association cell vectors with M and the output from the CMAC, respectively. The output from the CMAC, $\mathbf{K_c}$ consists of PID gains defined by

$$\mathbf{K_c}(t) := [K_{Pc}(t), K_{Ic}(t), K_{Dc}(t)]. \tag{5}$$

These mapping can be explained in detail as follows. Firstly, the CMAC composes the label space M for the input vector S ($S \to M$). Next, the weights are selected from the weight table for the label group M ($M \to A$). Finally, the output $\mathbf{K_c}$ is obtained by summing the selected weights($\mathbf{W_h}(t)$) ($A \to \mathbf{K_c}$). Thus, $\mathbf{K_c}(t)$ is generated by the following equation:

$$\mathbf{K_c}(t) = \sum_{h=1}^{k} \mathbf{W_h}(t), \tag{6}$$

where k denotes the number of weights which are extracted from the weight table.

In this paper, the inputs for the CMAC consist of the reference signal $r(t)$, $e(t)$ and $\Delta e(t)$. Here, $\Delta e(t)$ is defined as

$$e(t) := r(t) - y(t). \tag{7}$$

Furthermore, the only CMAC is worked until the error criterion is satisfied. Moreover, the learning rule to update the weights in the CMAC is as follows:

$$\mathbf{W_h}^{new}(t) = \mathbf{W_h}^{old}(t) - g(t)\frac{1}{k}\frac{\partial J(t+1)}{\partial \mathbf{K_c}(t)} \tag{8}$$

$$J(t+1) = \frac{1}{2}\varepsilon^2(t+1) \tag{9}$$

$$\varepsilon(t) := \frac{1}{2}\{y_m(t) - y(t)\}^2, \tag{10}$$

where $g(t)$ is given by

$$g(t) = \frac{1}{\beta_3 + \beta_1 \cdot \exp(-\beta_2|\varepsilon(t)|)}. \tag{11}$$

Here, β_1, β_2 and β_3 are the user-specified parameters. Moreover, each partial differentiation of Eq.(8) is developed as follows:

$$\left.\begin{aligned}
\frac{\partial J(t+1)}{\partial K_{Pc}(t)} &= \frac{\partial J}{\partial \varepsilon}\frac{\partial \varepsilon}{\partial y}\frac{\partial y}{\partial u}\frac{\partial u}{\partial K_{Pc}} \\
&= -\varepsilon(t+1)\Theta_1(t)\frac{\partial y(t+1)}{\partial u(t)} \\
\frac{\partial J(t+1)}{\partial K_{Ic}(t)} &= \frac{\partial J}{\partial \varepsilon}\frac{\partial \varepsilon}{\partial y}\frac{\partial y}{\partial u}\frac{\partial u}{\partial K_{Ic}} \\
&= -\varepsilon(t+1)\Theta_2(t)\frac{\partial y(t+1)}{\partial u(t)} \\
\frac{\partial J(t+1)}{\partial K_{Dc}(t)} &= \frac{\partial J}{\partial \varepsilon}\frac{\partial \varepsilon}{\partial y}\frac{\partial y}{\partial u}\frac{\partial u}{\partial K_{Dc}} \\
&= -\varepsilon(t+1)\Theta_3(t)\frac{\partial y(t+1)}{\partial u(t)},
\end{aligned}\right\} \tag{12}$$

where

$$\Theta_k(t) = \begin{cases} -\Delta y(t) & : k = 1 \\ e(t) & : k = 2 \\ -\Delta^2 y(t) & : k = 3. \end{cases} \tag{13}$$

However, the system Jacobian $\partial y(t+1)/\partial u(t)$ is required in order to calculate Eq.(12). Here, if this relation $x = |x|\text{sign}(x)$ is used, the system Jacobian is given as follows:

$$\frac{\partial y(t+1)}{\partial u(t)} = \left|\frac{\partial y(t+1)}{\partial u(t)}\right| \text{sign}\left(\frac{\partial y(t+1)}{\partial u(t)}\right), \quad (14)$$

where $\text{sign}(x) = 1(x > 0), -1(x < 0)$. Now, if $|\partial y(t+1)/\partial u(t)|$ can be included in $g(t)$, it is enough to know only the sign of the system gradient in advance(S.Omatu and T.Yamamoto, 1996). Therefore, in this paper, it is assumed that this sign of the system gradient is known.

$y_m(t)$ denotes the output of the reference model, which is designed as follows:

$$y_m(t) = \frac{z^{-1}T(1)}{T(z^{-1})} r(t), \quad (15)$$

where $T(z^{-1})$ is the desired polynomial and is defined by

$$T(z^{-1}) := 1 + t_1 z^{-1} + t_2 z^{-2}. \quad (16)$$

Furthermore, $T(1)$ denotes the static gain of $T(z^{-1})$, and $T(z^{-1})$ is designed based on a literature (T.Yamamoto and S.L.Shah, 1998).

Next, the error criterion to switch from the CMAC to the NN is determined as follows:

$$\eta(epoc) := \frac{1}{N} \sum_{t=1}^{N} \left\{ \frac{\varepsilon(t)}{r(t)} \right\}^2 \quad (17)$$

$$\Delta\eta(epoc) \le \bar{\eta}, \quad (18)$$

where N and $\eta(epoc)$ denote the number of steps per 1[epoc] and the scaling error, respectively. Here, since the design of $\bar{\eta}$ influences the control performance greatly, $\bar{\eta}$ is set as 0.01 as a measure in this paper, which is determined by examining several cases.

2.3 PID parameter tuning by NN

After the PID tuner is switched from the CMAC to the NN, PID gains are tuned using the NN whose detailed tuning rule is discussed below. The multilayered NN shown in Fig.3 is used in this paper. From Fig.3, the input layer for the NN is given as

$$\mathbf{I}(t) = [y(t-1), \cdots, y(t-n), u(t-1), \cdots, u(t-m), u_c(t-1), r(t), I_o], \quad (19)$$

where I_o denotes the threshold of the input layer, and $u_c(t)$ is given by

$$\Delta u_c(t) = K_{Ic} e(t) - K_{Pc} \Delta y(t) - K_{Dc} \Delta^2 y(t). \quad (20)$$

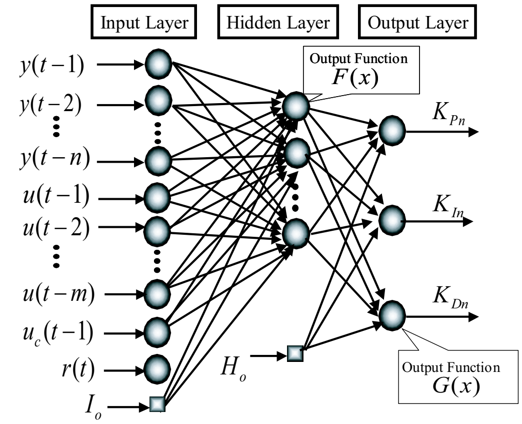

Fig. 3. Structure of the NN.

Furthermore, output functions of the hidden layer and the output layer are respectively defined as the following sigmoidal functions:

$$F(x) = 2\left\{\frac{1}{1+\exp(-ax)} - \frac{1}{2}\right\} \quad (21)$$

$$G(x) = c\left\{\frac{1}{1+\exp(-bx)} - \frac{1}{2}\right\}. \quad (22)$$

The output of the NN, $\mathbf{K_n} := [K_{Pn}, K_{In}, K_{Dn}]$, is calculated using the following equations:

$$T_j(t) = \sum_{i=0}^{p} V_{ij}(t) I_i(t) \quad (23)$$

$$H_j(t) = 2\left\{\frac{1}{1+\exp(-aT_j(t))} - \frac{1}{2}\right\} \quad (24)$$

$$S_k(t) = \sum_{j=0}^{q} W_{jk}(t) H_j(t) \quad (25)$$

$$O_k(t) = c\left\{\frac{1}{1+\exp(-bS_k(t))} - \frac{1}{2}\right\} \quad (26)$$

$$\mathbf{K_n}(t) = [O_1(t), O_2(t), O_3(t)], \quad (27)$$

where a, b and c denote the user-specified parameters, and p and q denote the number of neurons included in the input layer and the hidden layer, respectively. Furthermore, V_{ij} and W_{jk} respectively denote the weights of the input layer and the hidden layer. And H_o is the threshold of the hidden layer. Here, the PID gains $\mathbf{K}$ can be obtained by summing $\mathbf{K_c}$ and $\mathbf{K_n}$.

$$\mathbf{K}(t) = \mathbf{K_c}(t) + \mathbf{K_n}(t). \quad (28)$$

Next, the weights of the NN (V_{ij}, W_{jk}) are updated based on the BP method as follows:

$$W_{jk}(t+1) = W_{jk}(t) - \alpha_1 \frac{\partial J(t+1)}{\partial W_{jk}(t)}$$
$$= W_{jk}(t) + \alpha_1 \Theta_k(t) \varepsilon(t) \left(\frac{c}{2} - O_k(t)\right)\left(\frac{c}{2} + O_k(t)\right)$$
$$\cdot \frac{b}{c} H_j(t) \frac{\partial y(t+1)}{\partial u_n(t)} \quad (29)$$

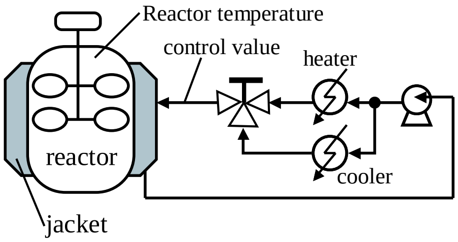

Fig. 4. The schematic figure of the polystyrene reactor model.

$$V_{ij}(t+1) = V_{ij}(t) - \alpha_2 \frac{\partial J(t)}{\partial V_{ij}(t)}$$
$$= V_{ij}(t) + \alpha_2 \Theta_k(t)\varepsilon(t)\frac{b}{c}\left(\frac{c}{2} - O_k(t)\right)\left(\frac{c}{2} + O_k(t)\right)$$
$$\cdot\frac{a}{2}(1-H_j(t))(1+H_j(t))I_i(t)W_{jk}(t)\frac{\partial y(t+1)}{\partial u_n(t)}, \quad (30)$$

where $\alpha_1(>0)$ and $\alpha_2(>0)$ are the learning coefficients. Furthermore, if the relation $\eta(\text{epoc}) \leq \eta_0$ is satisfied, the NN learning is stopped.

3. SIMULATION EXAMPLES

In order to evaluate the effectiveness of the newly proposed scheme, a simulation example for a nonlinear system is considered. As the nonlinear system, the following polystyrene reactor model is discussed. The schematic figure of the system is shown in Fig.4. The relation between the jacket temperature $u(t)$ and the reaction temperature $y(t)$ is described as follows(E.Nakanisi and Y.Hanakuma, 1992):

$$y(t) = 0.804y(t-1) + 5.739 \times 10^{15} \cdot \exp\{-E_a/R(y(t-1)+273)\} + 0.148u(t-1) + \xi(t), \quad (31)$$

where $E_a = 240$, $R = 0.01986$, and $\xi(t)$ denotes the white Gaussian noise with zero mean and variance 0.1^2. Nonlinear properties of this system become strong in the case of $y(t) \geq 75$. The reference signal $r(t)$ is given as follows:

$$r(t) = \begin{cases} 60(0 \leq t < 100) \\ 70(100 \leq t < 200) \\ 80(200 \leq t < 300) \\ 75(300 \leq t \leq 400). \end{cases} \quad (32)$$

Furthermore, the desired polynomial $T(z^{-1})$ was designed as follows:

$$T(z^{-1}) = 1 - 1.558z^{-1} + 0.449z^{-2}. \quad (33)$$

And, the user-specified parameters included in the CMAC and the NN were determined as shown in Table 1.

First, for the purpose of comparison with the conventional schemes, the fixed PID control scheme

Table 1. User-specified parameters on the CMAC and the NN

(a) **CMAC**

Width of quantization	10
Number of weight tables !!	10
User-specified parameters !! included in $g(t)$!!	β_1=100 β_2=0.01 β_3=10^5

(b) **NN**

Number of units in input layer	$p = 9$
Number of units in hidden layer	$q = 20$
User-specified parameters !! included in sigmoidal functions	$a = 1$ $b = 1$ $c = 2$
Learning rate (hidden layer)	α_1=0.0001
Learning rate (output layer)	α_2=0.0001
Threshold in input layer	I_o=1
Threshold in hidden layer	H_o=1
Permissible error to finish training	η_0=1.0×10^{-4}

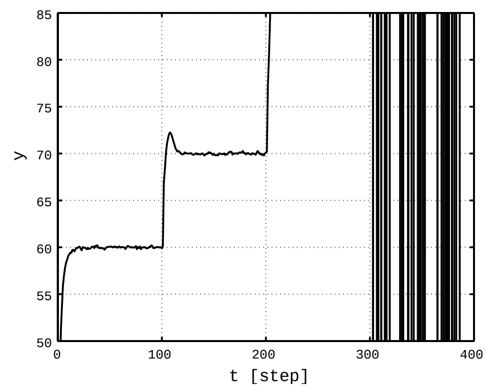

Fig. 5. Control result using the fixed PID controller.

widely used in industrial processes is employed, whose PID gains are tuned by using Chien, Hrones & Reswick (CHR) method (K.L.Chien and J.B.Reswick, 1972). Then, PID gains are as follows:

$$K_P = 6.5422, \quad K_I = 1.5522, \quad K_D = 1.8856. \quad (34)$$

The control result is shown in Fig.5. From Fig.5, owing to the nonlinearities of the controlled object, the control result by the fixed PID controller is oscillatory after 300[step].

Next, the control result using the proposed control scheme is shown in Fig.6, and then trajectories of PID gains are shown in Fig.7. From Fig.6 and Fig.7, the good control result can be obtained using the proposed method, because PID gains are adjusted adequately. Furthermore, error behaviors corresponding to the proposed method, the NN-PID scheme and the CMAC-PID scheme are shown in Fig.8. From Fig.8, it is clear that the learning speed using the NN-PID scheme is slow, and learning speed using the CMAC-PID scheme is very fast in the initial stage, but the learning speed is very slow subsequently. Here, the number of the first learning by the CMAC was 18[epoc] and the number of the additional learning by the

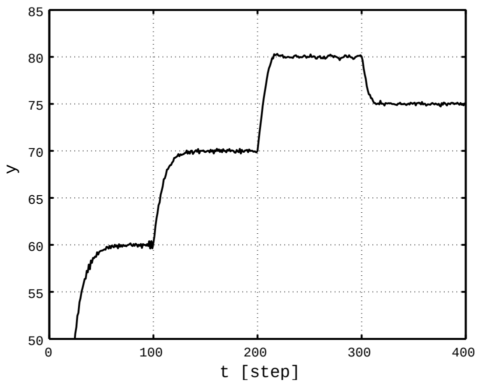

Fig. 6. Control result using the newly proposed control scheme.

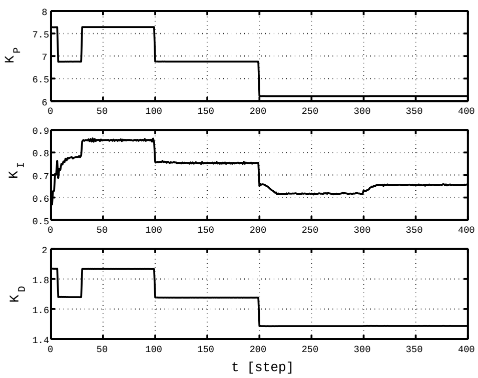

Fig. 7. Trajectories of PID gains corresponding to Fig.6.

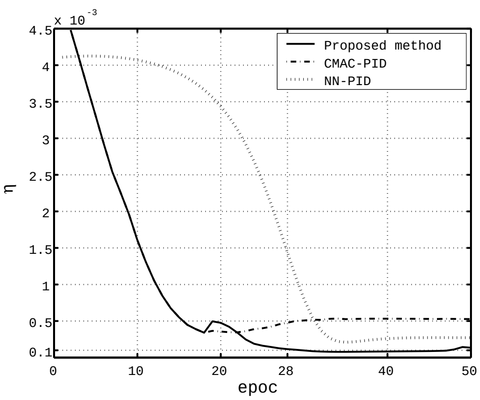

Fig. 8. Error behaviors where solid line, dashed and dotted line and dotted line are respectively corresponding to the proposed scheme, CMAC-PID scheme and NN-PID scheme.

NN was 10[epoc], that is, the total number of the learning was 28[epoc] till this relation $\eta \leq \eta_0$ was satisfied. On the other hand, control results could not be obtained so that the criterion $\eta \leq \eta_0$ was satisfied in the case where the only CMAC or the only NN was employed.

From these results, it is clear that the newly proposed scheme works well. The adaptability for non-trained reference signals has been examined. The control results are omitted due to the page limit.

4. CONCLUSIONS

In this paper, a new intelligent PID tuner has been proposed, which has a fusional structure of the CMAC and the NN. According to the newly proposed scheme, the roles in learning the PID tuner are divided into the CMAC and the NN. That is, the CMAC firstly work in the initial stage in order to roughly approximate the nonlinear components, and then the NN works to further improve the control performance. This structure enables us to drastically reduce the computational burden. The effectiveness of the newly proposed scheme has been verified on a simulation example of the chemical reactor model. PID gains have changed adequately corresponding to the operating points (reference signals), and suitable control results have been obtained. The application study of the proposed scheme is our future work.

REFERENCES

C.-Y.Seong and B.Widrow (2001). Neural dynamic optimization for control systems. *IEEE Trans. on Systems, Man, and Cybernetics* **31**, 482–513.

E.Nakanisi and Y.Hanakuma (1992). *The foundation and the practice of a process control(in Japanese)*. Asakura. Tokyo.

J.G.Zieglar and N.B.Nichols (1942). Optimum settings for automatic controllers. *Trans. ASME* **64**, 759–768.

J.S.Albus (1975*a*). Data storage in the control cerebellar model articulation control (cmac). *Trans. on ASME, J. of Dynamic Systems, Measurement, and Control* **9**, 228–233.

J.S.Albus (1975*b*). A new approach to manipulator control cerebellar model articulation control (cmac). *Trans. on ASME, J. of Dynamic Systems, Measurement, and Control* **9**, 220–227.

K.J. Åström, T. Hägglund (1988). *Automatic Tuning of PID Controllers*. Instrument Society of America. America.

K.L.Chien, J.A.Hrones and J.B.Reswick (1972). On the automatic control of generalized passive systems. *Trans. ASME* **74**, 175–185.

K.S.Narendra and K.Parthasarathy (1990). Identification and control of dynamical systems using neural networks. *IEEE Trans. on Neural Networks* **1**, 1–27.

S.Omatu and T.Yamamoto (1996). *Self-Turning Control*. SICE Learned Book. Japan.

S.Omatu, K.Marzuki and Y.Rubiyah (1995). *Neuro-Control and Its Applications*. Springer-Verlag. London.

T.Yamamoto and S.L.Shah (1998). Design and experimental evaluation of a multivariable self-tuning pid controller. *Proc. of IEEE Conference on Control Application* pp. 1230–1234.