Memory-Based On-Line Tuning of PID Controllers for Nonlinear Systems

Kenji Takao
Graduate School of Engineering
Hiroshima University
Hiroshima, Japan
takao17@hiroshima-u.ac.jp

Toru Yamamoto
Graduate School of Education
Hiroshima University
Hiroshima, Japan
yama@hiroshima-u.ac.jp

Takao Hinamoto
Graduate School of Engineering
Hiroshima University
Hiroshima, Japan
hinamoto@hiroshima-u.ac.jp

***Abstract*— Since most processes have nonlinearities, controller design schemes to deal with such systems are required. On the other hand, PID controllers have been widely used for process systems. Therefore, in this paper, a new design scheme of PID controllers based on a memory-based(MB) modeling is proposed for nonlinear systems. According to the MB modeling method, some local models are automatically generated based on input/output data pairs of the controlled object stored in the data-base. The proposed scheme generates PID parameters using stored input/output data in the data-base. This scheme can adjust the PID parameters in an on-line manner even if the system has nonlinear properties. Finally, the effectiveness of the newly proposed control scheme is numerically evaluated on a simulation example.**

I. INTRODUCTION

In recent years, many complicated control algorithms such as adaptive control theory or robust control theory have been proposed and implemented. However, in industrial processes, PID controllers[1], [2], [3] have been widely employed for about 80% or more of control loops. The reasons are summarized as follows. (1) the control structure is quit simple; (2) the physical meaning of control parameters is clear; and (3) the operators' know-how can be easily utilized in designing controllers. Therefore, it is still attractive to design PID controllers. However, since most process systems have nonlinearities, it is difficult to obtain good control performances for such systems simply using the fixed PID parameters. Therefore, PID parameters tuning methods using neural networks(NN)[4] and genetic algorithms(GA)[5] have been proposed until now. According to these methods, the learning cost is considerably large, and these PID parameters cannot be adequately adjusted due to the nonlinear properties. Therefore, it is quite difficult to obtain good control performances using these conventional schemes.

By the way, development of computers enables us to memorize, fast retrieve and read out a large number of data. By these advantages, the following method has been proposed: Whenever new data is obtained, the data is stored. Next, similar neighbors to the information requests, called '*queries*', are selected from the stored data. Furthermore, the local model is constructed using these neighbors. This memory-based(MB) modeling method, is called *Just-In-Time (JIT)* method[6], [7] , *Lazy Learning* method[8] or Model-on-Demand(MoD)[9], and these scheme have lots of attention in last decade.

In this paper, a design scheme of PID controllers based on the MB modeling method is discussed. A few PID controllers have been already proposed based on the JIT method[10] and the MoD method[11] which belong to the MB modeling methods. According to the former method, the JIT method is used as the purpose of supplementing the feedback controller with a PID structure. However, the tracking property is not guaranteed enough due to the nonlinearities in the case where reference signals are changed, because the controller does not includes any integral action in the whole control system. On the other hand, the latter method has a PID control structure. PID parameters are tuned by operators' skills, and they are stored in the data-base in advance. And also, a suitable set of PID parameters is generated using the stored data. However, the good control performance cannot be necessarily obtained in the case where nonlinearities are included in the controlled object and/or system parameters are changed, because PID parameters are not tuned in an on-line manner corresponding to characteristics of the controlled object.

Therefore, in this paper, a design scheme of PID controllers based on the MB modeling method is newly proposed. According to the proposed method, PID parameters which are obtained using the MB modeling method are adequately tuned in proportion to control errors, and modified PID parameters are stored in the data-base. Therefore, more suitable PID parameters corresponding to characteristics of the controlled object are newly stored. Moreover, an algorithm to avoid the excessive increase of the stored data, is further discussed. This algorithm yields the reduction of memories and computational costs. Finally, the effectiveness of the newly proposed control scheme is examined on a simulation example.

II. PID CONTROLLER DESIGN BASED ON MEMORY-BASED MODELING METHOD

A. MB modeling method

First, the following discrete-time nonlinear system is considered:

$$y(t) = f(\phi(t-1)), \tag{1}$$

where $y(t)$ denotes the system output and $f(\cdot)$ denotes the nonlinear function. Moreover, $\phi(t-1)$ is called '*information vector*', which is defied by the following equation:

$$\phi(t) := [\ y(t-1), \cdots, y(t-n_y), \\ u(t-1), \cdots u(t-n_u)\ ], \tag{2}$$

where $u(t)$ denotes the system input. Also, n_y and n_u respectively denote the orders of the system output and the system input, respectively. According to the MB modeling method, the data is stored in the form of the information vector ϕ expressed in Eq.(2). Moreover, $\phi(t)$ is required in calculating the estimate of the output $y(t+1)$ called '*query*'. That is, after some similar neighbors to the query are selected from the data-base, the predictive value of the system can be obtained using these neighbors.

B. Controller design based on MB modeling method

In this paper, the following control law with a PID structure is considered:

$$\begin{aligned}\Delta u(t) &= \frac{k_c T_s}{T_I} e(t) - k_c \left(\Delta + \frac{T_D}{T_s} \Delta^2 \right) y(t) \quad (3) \\ &= K_I e(t) - K_P \Delta y(t) - K_D \Delta^2 y(t), \quad (4)\end{aligned}$$

where $e(t)$ denotes the control error signal defined by

$$e(t) := r(t) - y(t). \quad (5)$$

$r(t)$ denotes the reference signal. Also, k_c, T_I and T_D respectively denote the proportional gain, the reset time and the derivative time, and T_s denotes the sampling interval. Here, K_P, K_I and K_D included in Eq.(4) are derived by the relations $K_P = k_c$, $K_I = k_c T_s / T_I$ and $K_D = k_c T_D / T_s$. Δ denotes the differencing operator defined by $\Delta := 1 - z^{-1}$. Here, it is quite difficult to obtain a good control performance due to nonlinearities, if PID parameters(K_P, K_I, K_D) in Eq.(4) are fixed. Therefore, a new control scheme is proposed, which can adjust PID parameters in an on-line manner corresponding to characteristics of the system. Thus, instead of Eq.(4), the following PID control law with variable PID parameters is employed:

$$\Delta u(t) = K_I(t) e(t) - K_P(t) \Delta y(t) - K_D(t) \Delta^2 y(t). \quad (6)$$

Now, Eq.(6) can be rewritten as the following relations:

$$u(t) = g(\phi'(t)) \quad (7)$$

$$\phi'(t) := [\mathbf{K}(t), r(t), y(t), y(t-1), y(t-2), u(t-1)] \quad (8)$$

$$\mathbf{K}(t) := [K_P(t), K_I(t), K_D(t)], \quad (9)$$

where $g(\cdot)$ denotes a linear function. By substituting Eq.(7) and Eq.(8) into Eq.(1) and Eq.(2), the following equation can be derived:

$$y(t+1) = h(\tilde{\phi}(t)) \quad (10)$$

$$\begin{aligned}\tilde{\phi}(t) := [\ & y(t), \cdots, y(t-n_y+1), \mathbf{K}(t), r(t), \\ & u(t-1), \cdots, u(t-n_u+1)\ ], \quad (11)\end{aligned}$$

where $n_y \geq 3$, $n_u \geq 2$, and $h(\cdot)$ denotes a nonlinear function. Therefore, $\mathbf{K}(t)$ is given by the following equations:

$$\mathbf{K}(t) = F(\bar{\phi}(t)) \quad (12)$$

$$\begin{aligned}\bar{\phi}(t) := [\ & y(t+1), y(t), \cdots, y(t-n_y+1), \\ & r(t), u(t-1), \cdots, u(t-n_u+1)\ ], \quad (13)\end{aligned}$$

where $F(\cdot)$ denotes a nonlinear function. Since the future output $y(t+1)$ included in Eq.(13) cannot be obtained at t, $y(t+1)$ is replaced by $r(t+1)$. Because the control system so that can realize $y(t+1) \to r(t+1)$, is designed in this paper. Therefore, $\bar{\phi}(t)$ included in Eq.(13) is newly rewritten as follows:

$$\begin{aligned}\bar{\phi}(t) := [\ & r(t+1), r(t), y(t), \cdots, y(t-n_y+1), \\ & u(t-1), \cdots, u(t-n_u+1)\ ]. \quad (14)\end{aligned}$$

After the above preparation, a new PID control scheme is designed based on the MB modeling method. The controller design algorithm is summarized as follows.

[STEP 1] Generate initial data-base

The MB modeling method cannot work if the past data is not saved at all. Therefore, PID parameters are firstly calculated using Zieglar & Nichols method[2] or Chien, Hrones & Reswick(CHR) method[3] based on historical data of the controlled object in order to generate the initial data-base. That is, $\mathbf{\Phi}(j)$ indicated in the following equation is generated as the initial data-base:

$$\mathbf{\Phi}(j) := [\ \bar{\phi}(j)\ ,\ \mathbf{K}(j)\ ]\,, \quad j = 1, 2, \cdots, N(0) \quad (15)$$

where $\bar{\phi}(j)$ and $\mathbf{K}(j)$ are given by Eq.(14) and Eq.(9). Moreover, $N(0)$ denotes the number of information vectors stored in the initial data-base. Note that all PID parameters included in the initial information vectors are equal, that is, $\mathbf{K}(1) = \mathbf{K}(2) = \cdots = \mathbf{K}(N(0))$ in the initial stage.

[STEP 2] Calculate distance and select neighbors

Distances between the query $\bar{\phi}(t)$ and the information vectors $\bar{\phi}(i)(i \neq k)$ are calculated using the following $\mathcal{L}_1$-norm with some weights:

$$d(\bar{\phi}(t), \bar{\phi}(j)) = \sum_{l=1}^{n_y+n_u+1} \left| \frac{\bar{\phi}_l(t) - \bar{\phi}_l(j)}{\max\limits_m \bar{\phi}_l(m) - \min\limits_m \bar{\phi}_l(m)} \right|, \quad (16)$$

$$(\ j = 1, 2, \cdots, N(t)\ )$$

where $N(t)$ denotes the number of information vectors stored in the data-base when the query $\bar{\phi}(t)$ is given. Furthermore, $\bar{\phi}_l(j)$ denotes the l-th element of the j-th information vector. Similarly, $\bar{\phi}_l(t)$ denotes the l-th element of the query at t. Moreover, $\max \bar{\phi}_l(m)$ denotes the maximum element among the l-th element of all information vectors($\bar{\phi}(j)$, $j = 1, 2, \cdots, N(t)$) stored in the data-base. Similarly, $\min \bar{\phi}_l(m)$ denotes the minimum element. Here, k pieces with the smallest distances are chosen from all information vectors.

[STEP 3] Construct local model

Next, using k neighbors selected in STEP 2, the local model is constructed based on the following Linearly

Weighted Average(LWA)[12]:

$$\mathbf{K}^{old}(t) = \sum_{i=1}^{k} w_i \mathbf{K}(i), \qquad \sum_{i=1}^{k} w_i = 1, \tag{17}$$

where w_i denotes the weight corresponding to the i-th information vector $\bar{\phi}(i)$ in the selected neighbors, and is calculated by:

$$w_i = \sum_{l=1}^{n_u+n_y+1} \left(1 - \frac{[\bar{\phi}_l(t) - \bar{\phi}_l(i)]^2}{[\max_m \bar{\phi}_l(m) - \min_m \bar{\phi}_l(m)]^2} \right). \tag{18}$$

[STEP 4] Data adjustment

In the case where information corresponding to the current state of the controlled object is not effectively saved in the data-base, a suitable set of PID parameters cannot be effectively calculated. That is, it is necessary to adjust PID parameters so that the control error decreases. Therefore, PID parameters obtained in STEP 3 are updated corresponding to the control error, and these new PID parameters are stored in the data-base. The following steepest descent method is utilized in order to modify PID parameters:

$$\mathbf{K}^{new}(t) = \mathbf{K}^{old}(t) - \eta \frac{\partial J(t+1)}{\partial \mathbf{K}(t)} \tag{19}$$

$$\eta := [\eta_P, \eta_I, \eta_D], \tag{20}$$

where η denotes the learning rate, and the following $J(t+1)$ denotes the error criterion:

$$J(t+1) := \frac{1}{2}\varepsilon(t+1)^2 \tag{21}$$

$$\varepsilon(t) := y_r(t) - y(t). \tag{22}$$

$y_r(t)$ denotes the output of the reference model which is given by:

$$y_r(t) = \frac{z^{-1}T(1)}{T(z^{-1})} r(t) \tag{23}$$

$$T(z^{-1}) := 1 + t_1 z^{-1} + t_2 z^{-2}. \tag{24}$$

Here, $T(z^{-1})$ is designed based on the reference literature[13]. Moreover, each partial differential of Eq.(19) is developed as follows:

$$\left.\begin{aligned}
\frac{\partial J(t+1)}{\partial K_P(t)} &= \frac{\partial J(t+1)}{\partial \varepsilon(t+1)} \frac{\partial \varepsilon(t+1)}{\partial y(t+1)} \frac{\partial y(t+1)}{\partial u(t)} \frac{\partial u(t)}{\partial K_P(t)} \\
&= \varepsilon(t+1)(y(t) - y(t-1)) \frac{\partial y(t+1)}{\partial u(t)} \\
\frac{\partial J(t+1)}{\partial K_I(t)} &= \frac{\partial J(t+1)}{\partial \varepsilon(t+1)} \frac{\partial \varepsilon(t+1)}{\partial y(t+1)} \frac{\partial y(t+1)}{\partial u(t)} \frac{\partial u(t)}{\partial K_I(t)} \\
&= -\varepsilon(t+1)e(t) \frac{\partial y(t+1)}{\partial u(t)} \\
\frac{\partial J(t+1)}{\partial K_D(t)} &= \frac{\partial J}{\partial \varepsilon(t+1)} \frac{\partial \varepsilon(t+1)}{\partial y(t+1)} \frac{\partial y(t+1)}{\partial u(t)} \frac{\partial u(t)}{\partial K_D(t)} \\
&= \varepsilon(t+1)(y(t) - 2y(t-1) + y(t-2)) \frac{\partial y(t+1)}{\partial u(t)}.
\end{aligned}\right\} \tag{25}$$

Note that *a priori* information with respect to the system Jacobian $\partial y(t+1)/\partial u(t)$ is required in order to calculate Eq.(25). Here, using the relation $x = |x|\mathrm{sign}(x)$, the system Jacobian can be obtained by the following equation:

$$\frac{\partial y(t+1)}{\partial u(t)} = \left| \frac{\partial y(t+1)}{\partial u(t)} \right| \mathrm{sign}\left(\frac{\partial y(t+1)}{\partial u(t)} \right), \tag{26}$$

where $\mathrm{sign}(x) = 1(x > 0), -1(x < 0)$. Now, if the sign of the system Jacobian is known in advance, by including $|\partial y(t+1)/\partial u(t)|$ in η, the usage of the system Jacobian can make easy[14]. Therefore, it is assumed that the sign of the system Jacobian is known in this paper.

[STEP 5] Remove redundant data

In implementing to real systems, the newly proposed scheme has a constraint that the calculation from STEP 2 to STEP 4 must be finished within the sampling time. Here, storing the redundant data in the data-base needs excessive computational time. Therefore, an algorithm to avoid the excessive increase of the stored data, is further discussed. The procedure is carried out in the following two steps.

First, the information vectors $\mathbf{\Phi}(\bar{i})$ which satisfy the following first condition, are extracted from the data-base:

[First condition]

$$d(\bar{\phi}(t), \bar{\phi}(i)) \leq \alpha_1, \quad i = 1, 2, \cdots, N(t) - k \tag{27}$$

where $\mathbf{\Phi}(\bar{i})$ is defined by

$$\mathbf{\Phi}(\bar{i}) := \left[\bar{\phi}(\bar{i}) \ \mathbf{K}(\bar{i}) \right]. \quad \bar{i} = 1, 2, \cdots \tag{28}$$

Moreover, the information vectors $\mathbf{\Phi}(\hat{i})$ which satisfy the following second condition, are further chosen from the extracted $\mathbf{\Phi}(\bar{i})$:

[Second condition]

$$\sum_{l=1}^{3} \left\{ \frac{\mathbf{K}_l(\bar{i}) - \mathbf{K}_l^{new}(t)}{\mathbf{K}_l^{new}(t)} \right\}^2 \leq \alpha_2, \tag{29}$$

where $\mathbf{\Phi}(\hat{i})$ is defined by

$$\mathbf{\Phi}(\hat{i}) := \left[\bar{\phi}(\hat{i}), \ \mathbf{K}(\hat{i}) \right]. \quad \hat{i} = 1, 2, \cdots \tag{30}$$

If there exist plural $\mathbf{\Phi}(\hat{i})$, the information vector with the smallest value in the second condition among all $\mathbf{\Phi}(\hat{i})$, is only removed. By the above procedure, the redundant data can be removed from the data-base.

Here, a block diagram summarized mentioned above algorithms are shown in Fig.1.

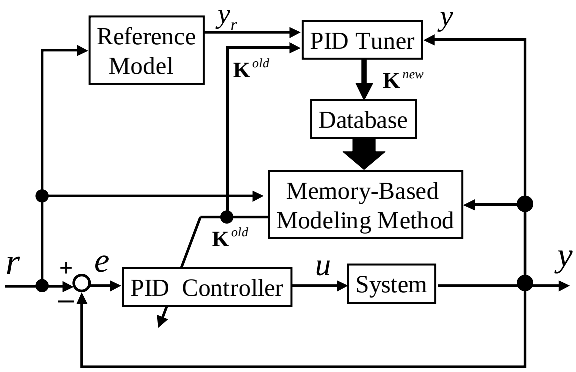

Fig. 1. Block diagram of the proposed system.

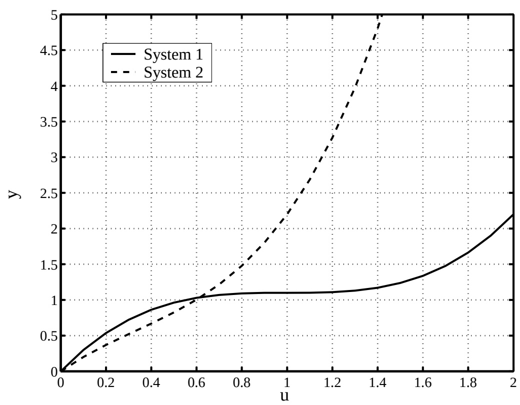

Fig. 2. Static properties of System 1 and System 2.

III. Simulation Example

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 Hammerstein model[15] is discussed:

[System 1]

$$\left.\begin{array}{l} y(t) = 0.6y(t-1) - 0.1y(t-2) \\ \qquad +1.2x(t-1) - 0.1x(t-2) + \xi(t) \\ x(t) = 1.5u(t) - 1.5u^2(t) + 0.5u^3(t) \end{array}\right\} \qquad (31)$$

[System 2]

$$\left.\begin{array}{l} y(t) = 0.6y(t-1) - 0.1y(t-2) \\ \qquad +1.2x(t-1) - 0.1x(t-2) + \xi(t) \\ x(t) = 1.0u(t) - 1.0u^2(t) + 1.0u^3(t) \end{array}\right\} \qquad (32)$$

where $\xi(t)$ denotes the white Gaussian noise with zero mean and variance 0.01^2. Static properties of System 1 and System 2 are shown in Fig.2. ¿From Fig.2, it is clear that gains of System 2 are larger than ones of System 1 at $y \geq 1.0$.

Here, the reference signal $r(t)$ is given by:

$$r(t) = \begin{cases} 0.5(0 \leq t < 50) \\ 1.0(50 \leq t < 100) \\ 2.0(100 \leq t < 150) \\ 1.5(150 \leq t \leq 200). \end{cases} \qquad (33)$$

The information vector $\bar{\phi}$ is defined as follows:

$$\begin{aligned} \bar{\phi}(t) := [\ & r(t+1), r(t), \\ & y(t), y(t-1), y(t-2), u(t-1)\ ]. \end{aligned} \qquad (34)$$

The desired characteristic polynomial $T(z^{-1})$ included in the reference model was designed as follows:

$$T(z^{-1}) = 1 - 0.271z^{-1} + 0.0183z^{-2}, \qquad (35)$$

where $T(z^{-1})$ was designed based on the reference literature[13]. Furthermore, the user-specified parameters included in the proposed method are determined as shown in Table I.

TABLE I

User-specified parameters included in the proposed method (Hammerstein model).

Orders of the information vector	$n_y = 3$ $n_u = 2$
Number of neighbors	$k = 6$
Learning rates	$\eta_P = 0.8$ $\eta_I = 0.8$ $\eta_D = 0.2$
Coefficients to inhibit the data	$\alpha_1 = 0.5$ $\alpha_2 = 0.1$
Initial number of data	$N(0) = 6$

For the purpose of comparison, the fixed PID control scheme which has widely used in industrial processes was first employed, whose PID parameters were tuned by CHR method[3]. Then, PID parameters were calculated as

$$K_P = 0.486, \quad K_I = 0.227, \quad K_D = 0.122. \qquad (36)$$

Moreover, the PID controller using the NN, called NN-PID controller, was applied for the purpose of the comparison, where the NN was utilized in order to supplement the fixed PID controller.

The control results for System 1 are summarized in Fig.3, where the solid line and dashed line denote the control results of the proposed method and the fixed PID controller, respectively. Furthermore, trajectories of PID parameters using the proposed method are shown in Fig.4. From Fig.3, owing to nonlinearities of the controlled object, the control result by the fixed PID controller is not good. On the other hand, from Fig.3 and Fig.4, the good control result can be obtained using the proposed method, because PID parameters are adequately adjusted. Moreover, the number of data stored in the database was 49. Using the algorithm to remove needless data, the number of data stored in the data-base can be effectively

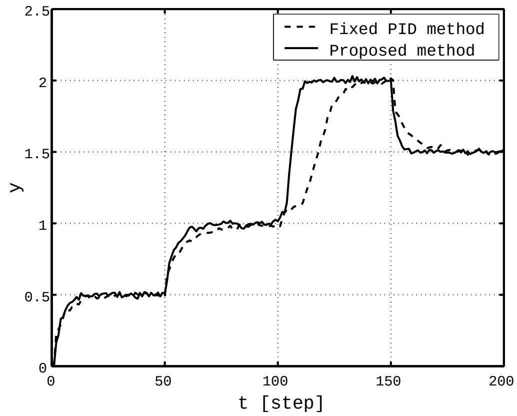

Fig. 3. Control results using the proposed method(solid line) and the fixed PID control(broken line) for System1.

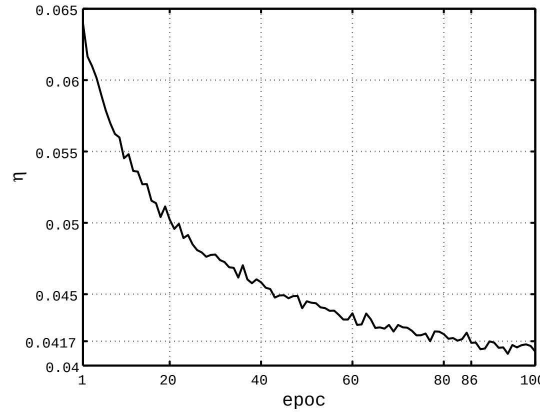

Fig. 5. Error behaviors using the controller fused the fixed PID with the NN-PID for Hammerstein model.

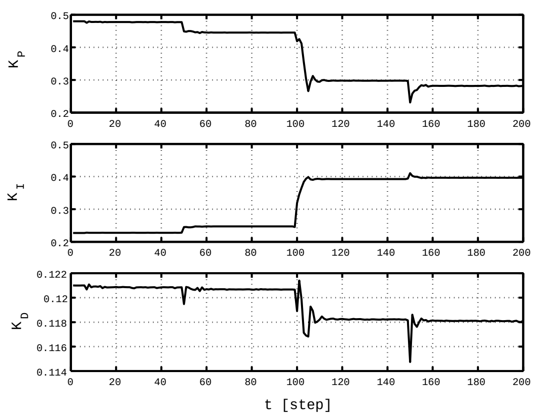

Fig. 4. Trajectories of PID parameters corresponding to Fig.3.

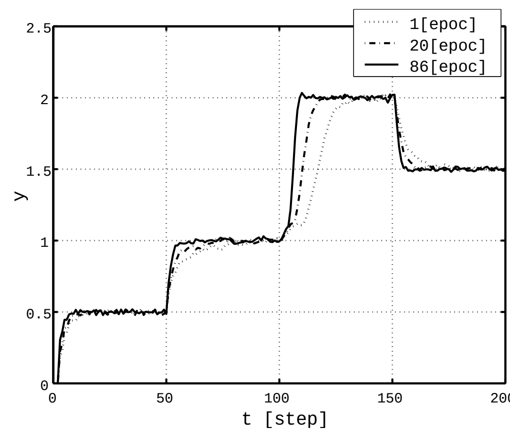

Fig. 6. Control result using the controller fused the fixed PID with the NN-PID for Hammerstein model.

reduced from 206 to 49. In addition, the error ϵ given by the following equation was 0.0417 using the proposed method:

$$\epsilon(epoc) := \frac{1}{N}\sum_{t=1}^{N}\left\{\frac{\varepsilon(t)}{r(t)}\right\}^2, \qquad (37)$$

where N denotes the number of steps per 1[epoc]. Furthermore, the number of iteration was set as 1, because PID parameters can be adjusted in an on-line manner by the proposed method. Moreover, the NN-PID controller was applied to System 1. Error behaviors of ϵ expressed in Eq.(37) are shown in Fig.5, and control results are shown in Fig.6. ¿From Fig.5, the necessary number for learning iterations was 86[epoc] until the control result using the NN-PID controller could be obtained the same control performances as the proposed method, that is, until $\epsilon \leq 0.0417$ was satisfied. Therefore, the effectiveness of the proposed method is also verified in comparison with the NN-PID controller for nonlinear systems.

Next, the case where the system has time-variant parameters is considered. That is, the system changes from Eq.(31) to Eq.(32) at $t = 70$. First, the control result with the fixed PID controller, is shown in Fig.7, where PID parameters are set as the same parameters as shown in Eq.(36). Since the gain of the controlled object becomes high gain around $r(t) = 2.0$, the fixed PID controller does not work well. On the other hand, the proposed control scheme was employed in this case. The control result and trajectories of PID parameters are shown in Fig.8 and Fig.9. ¿From these figures, a good control performance can be also obtained because PID parameters are adequately adjusted using the proposed method. The usefulness for the nonlinear system with time-variant parameters is suggested in this example.

IV. Conclusions

In this paper, a new design scheme of PID controllers using the MB modeling method has been proposed. Many PID controller design schemes using NNs and GAs have been proposed for nonlinear systems up to now. In employing these scheme for real systems, however, it is a serious problem that the learning cost becomes considerably large.

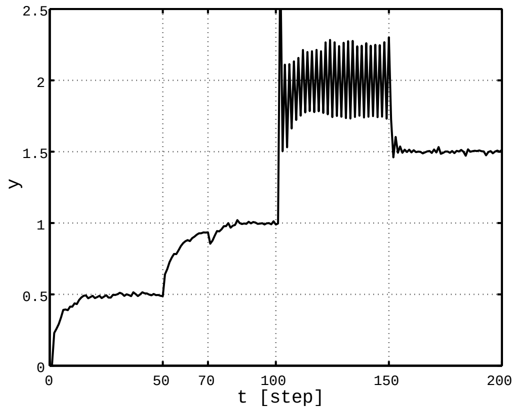

Fig. 7. Control result using the fixed PID controller in the case where the system is changed from System1 to System2.

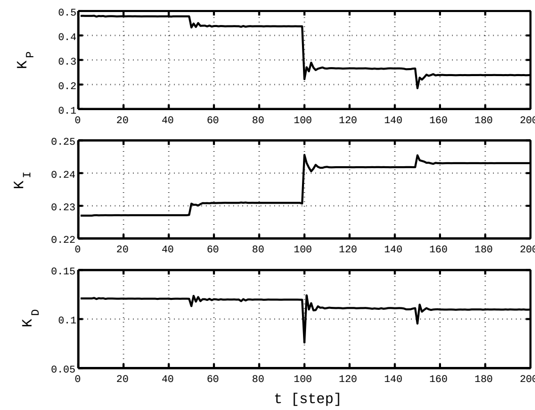

Fig. 9. Trajectories of PID parameters corresponding to Fig.8.

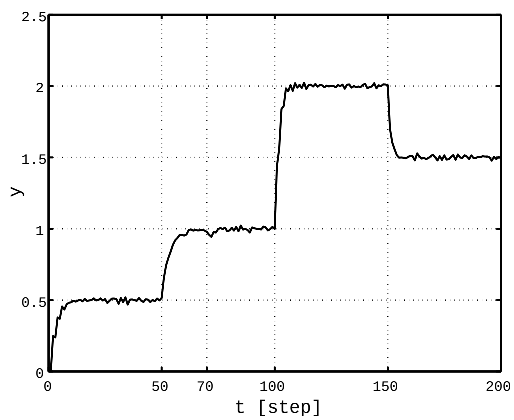

Fig. 8. Control result using the proposed method in the case where the system parameters are changed.

On the other hand, according to the proposed method, such computational burdens can be effectively reduced using the algorithm for removing the redundant data. In addition, the effectiveness of the proposed method have been verified by a numerical simulation example.

The application of the newly proposed scheme for real systems and the extension to multivariable cases are currently under consideration.

V. REFERENCES

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

[2] J.G.Zieglar and N.B.Nichols: Optimum settings for automatic controllers, Trans. ASME, Vol.64, No.8, 759/768 (1942)

[3] K.L.Chien, J.A.Hrones and J.B.Reswick: On the Automatic Control of Generalized Passive Systems, Trans. ASME, Vol.74, 175/185 (1972)

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

[5] B.Porter and A.H.Jones: Genetic tuning of digital PID controllers, Electronics Letter, Vol.28, 843/844 (1992)

[6] A.Stenman, F.Gustafsson and L.Ljung: Just in time models for dynamical systems, 35th IEEE Conference on Decision and Control, 1115/1120 (1996)

[7] Q.Zheng and H.Kimura: A New Just-In-Time Modeling Method and Its Applications to Rolling Set-up Modeling (in Japanese) ; Trans. on SICE, vol.37, No.7, 640/646 (2001)

[8] G.Bontempi, M.Birattari and H.Bersini: Lazy learning for local modeling and control design, International Journal of Control, Vo.72, No.7-8, 643/658 (1999)

[9] A.Stenman: Model on demand: Algorithms, analysis and applications, PhD thesis, Department of Electrical Engineering Linköping University (1990)

[10] Q.Zheng and Hidenori Kimura: Just-in-time PID Control, The 44th Japan Joint Automatic Control Conference, Tokyo, 336/339 (2001)

[11] J.Ohta and S.Yamamoto: Auto-Tuning of PID Controllers Via Model-on-Demand; Proceedings of the 47th Annual Conference of the Institute of Systems, Control and Information Engineers (ISCI), No.1045, 105/106, Kyoto (2003)

[12] C.G.Atkeson, A.W.Moore and S.Schaal: Locally weighted learning for control, Artificial Intelligence Review, Vol.11, 75/114 (1997)

[13] T.Yamamoto and S.L.Shah: Design and Experimental Evaluation of a Multivariable Self-Tuning PID Controller; Proc. of IEEE Conference on Control Applications, Trieste, 1230/1234 (1998)

[14] S.Omatu and T.Yamamoto: Self-Turning Control (in Japanese); SICE Learned Book, (1996)

[15] L.Zi-Qiang: On identification of the controlled plants described by the Hammerstein system, IEEE Trans. Automat. Contr., Vol.AC-39, 569/573 (1994)