Design of a Data-Driven PID Controller

Toru Yamamoto, *Member, IEEE*, Kenji Takao, and Takaaki Yamada

***Abstract*—Since most processes have nonlinearities, controller design schemes to deal with such systems are required. On the other hand, proportional-integral-derivative (PID) controllers have been widely used for process systems. Therefore, in this paper, a new design scheme of PID controllers based on a data-driven (DD) technique is proposed for nonlinear systems. According to the DD technique, a suitable set of PID parameters is automatically generated based on input/output data pairs of the controlled object stored in the database. This scheme can adjust the PID parameters in an online manner even if the system has nonlinear properties and/or time-variant system parameters. Finally, the effectiveness of the newly proposed control scheme is evaluated on some simulation examples, and a pilot-scale temperature control system.**

***Index Terms*—Database, nonlinear control, proportional-integral-derivative (PID) control, process control, self-tuning control.**

I. Introduction

In recent years, many complicated control algorithms such as adaptive control theory and/or robust control theory have been proposed and implemented for real systems. Even though these complicated and subtle control algorithms exist, less sophisticated proportional-integral-derivative (PID) controllers continue to be widely employed in process industries. The reasons for the continued popularity of PID controllers are summarized as follows: 1) the control structure is quite simple; 2) the physical meaning of control parameters is clear; and 3) the operators' know-how can be easily utilized in designing controllers.

Given these reasons, it is still attractive to design PID controllers, but they do have their drawbacks. Since most process systems have nonlinearities, it is difficult to obtain good control performances for such systems simply using the fixed PID parameters. The adaptive or self-tuning PID control schemes [1], [2] have been frequently employed for systems with weak nonlinearity. However, since these methods have been mainly researched and discussed for use with linear systems, it is impossible to obtain suitable control results for systems with strong nonlinearity. Where there is strong nonlinearity, PID parameters tuning methods using neural networks (NN) [3] and genetic algorithms (GA) [4] have been proposed, but there are obstacles to these methods. According to these methods, the learning cost is considerably large, and these PID parameters cannot be adequately adjusted due to the nonlinear properties. Therefore, to the best of our knowledge, there are currently few PID parameter tuning schemes of practical use for nonlinear systems.

In contrast, the development of computers has been enormous in the same time period. These advances in computers enable them to store, quickly retrieve, and read out an ever-increasing amount of data. With these advantages in mind, the following method has been proposed based on the computer data storage system: whenever new data is obtained, the data is stored. Next, similar neighbors to the information requests, called "queries," are selected from the stored data. Additionally, the local model or the local controller is constructed using these neighbors. This is known as a data-driven (DD) technique. The DD techniques called the Just-in-Time (JIT) method [5], [6], the Lazy Learning method [7], [8], and the Model-on-Demand (MoD) method [9] have received lots of attention in the last decade.

A few similar PID controllers have been already proposed based on the JIT method [10] and the MoD method [11]. Generally, the JIT method is used to supplement the feedback controller with a PID structure, however it is problematic. The tracking property is not reliable enough due to the nonlinearities occurring in the case where reference signals are changed. In this method, the controller does not create 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 database in advance. Also, a suitable set of PID parameters is generated using the stored data. Still, good control over the performance cannot be necessarily obtained in the case where nonlinearities are included in the controlled object and/or system parameters are changed. This is due to the fact that PID parameters are not adjusted in an online manner corresponding to properties of the controlled object.

In an effort to overcome the shortcomings of the JIT and MoD methods, this paper proposes and describes a new design scheme of PID controllers, based on the DD technique. According to the proposed method, PID parameters obtained using the DD technique are adequately updated in proportion to control errors, and modified PID parameters are stored in the database. The main feature of the newly proposed scheme is that the PID parameters are updated corresponding to the control error in an online manner and they are stored in the database. Therefore, the learning effect is gradually accumulated in the database. In order to avoid the excessive accumulation of the stored data, an algorithm is put forward which reduces the need for memory storage and computational costs. Finally, the effectiveness of the newly proposed control scheme is examined on some simulation examples, and experimentally evaluated on a pilot-scale temperature control system.

Manuscript received April 19, 2006; revised February 04, 2007 and October 06, 2007. Manuscript received in final form January 31, 2008. First published September 26, 2008; current version published December 24, 2008. Recommended by Associate Editor S.-L. Jamsa-Jounela.

T. Yamamoto is with the Department of Technology and Information Education, Hiroshima University, Higashi-Hiroshima 739-8524, Japan (e-mail: yama@hiroshima-u.ac.jp).

K. Takao is with the Technical Headquarters, Mitsubishi Heavy Industries, Ltd., Mihara 729-0393, Japan.

T. Yamada is with the Industrial Automation Company, OMRON Corporation, Kusatsu 525-0035, Japan.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCST.2008.921808

II. DD PID Controller Design

A. DD Technique

In order to create the DD technique, several mathematical considerations must first be undertaken. The first consideration is the following discrete-time nonlinear system:

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

where $y(t)$ denotes the system output and $f(\cdot)$ denotes the nonlinear function. In this case, $\phi(t-1)$ is called the "*information vector*," which is defined by the following equation:

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

where $u(t)$ denotes the control input. Also, n_y and n_u denote the orders of the system output and the control input, respectively. Accordingly, (2) means that the controlled object is described using elements included in $\phi(t-1)$. In addition, it is assumed that, with regard to the controlled object, the sign of the partial derivative of $y(t)$ with respect to $u(t-1)$ is known. The sign of the partial derivative is also the sign of the system Jacobian, so the system Jacobian is a known variable.

According to the DD technique, the data is stored in the form of the information vector ϕ expressed in (2). Moreover, the query $\phi(t-1)$ is required in calculating the estimate of the output $y(t)$, that is, after some similar neighbors to the query are selected from the database, the prediction of the system can be obtained using these neighbors. The controller design is discussed next.

B. Controller Design

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

$$\Delta u(t) = \frac{k_c T_s}{T_I} e(t) - k_c \left(\Delta + \frac{T_D}{T_s} \Delta^2 \right) y(t) \tag{3}$$

$$= K_I e(t) - K_P \Delta y(t) - K_D \Delta^2 y(t) \tag{4}$$

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

$$e(t) := r(t) - y(t). \tag{5}$$

The reference signal is denoted by $r(t)$. 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 (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 $\Delta := 1 - z^{-1}$. However, it is quite difficult to obtain a good control performance due to nonlinearities if the PID parameters (K_P, K_I, and K_D) in (4) are fixed. Therefore, a new control scheme is proposed that can adjust PID parameters in an online manner corresponding to properties of the system. Thus, instead of (4), the following PID control law with time-variant PID parameters is employed:

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

Thus, with those additional pieces of information, (6) can be rewritten as

$$u(t) = g(\phi'(t)) \tag{7}$$

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

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

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

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

$$\tilde{\phi}(t) := [y(t), \dots, y(t-n_y+1), \mathbf{K}(t), r(t), u(t-1), \dots, u(t-n_u+1)] \tag{11}$$

where $n_y \geq 3$, $n_u \geq 2$, and $h(\cdot)$ denotes a nonlinear function. Therefore, the PID parameters $\mathbf{K}(t)$ are related to the information vector $\bar{\phi}(t)$, which is expressed as

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

$$\bar{\phi}(t) := [y(t+1), y(t), \dots, y(t-n_y+1), r(t), u(t-1), \dots, u(t-n_u+1)]. \tag{13}$$

Since the future output $y(t+1)$ included in (13) cannot be obtained at t, $y(t+1)$ is replaced by $r(t+1)$, because the purpose of the control considered in this paper is to realize $y(t+1) \to r(t+1)$. Therefore, $\bar{\phi}(t)$ included in (13) is newly rewritten as follows:

$$\bar{\phi}(t) := [r(t+1), r(t), y(t), \dots, y(t-n_y+1), u(t-1), \dots, u(t-n_u+1)]. \tag{14}$$

After making the above preparations, it is possible to design a new PID control scheme based on the DD technique. The procedure of the controller design is summarized as follows.

[STEP 1] Generate Initial Data-Base: The DD technique cannot work if the historical data are not saved at all, as there will be no information from which to work. It is necessary, then, to first create a database based on historical data of the controlled object. Therefore, PID parameters are initially calculated using either the Ziegler and Nichols method [12] or the Chien, Hrones, and Reswick(CHR) method [13]. Of course, PID parameters determined through the use of the operators' skill and/or know-how can be also utilized as the initial database, that is, $\mathbf{\Phi}(j)$ indicated in the following equation is generated as the initial database:

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

where $\bar{\phi}(j)$ and $\mathbf{K}(j)$ are given by (14) and (9). Moreover, $N(0)$ denotes the number of information vectors stored in the initial database. If one set of fixed PID parameters is chosen as being typical, then all PID parameters included in the initial information vectors may be equal. Expressed numerically, that is, $\mathbf{K}(1) = \mathbf{K}(2) = \cdots = \mathbf{K}(N(0))$ in the initial stage.

[STEP 2] Calculate Distance and Select Neighbors: It is necessary though, to determine the distances between the query

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

$$d(\bar{\phi}(t), \bar{\phi}(j)) = \sum_{l=1}^{n_y+n_u+1} \left| \frac{\bar{\phi}_l(t) - \bar{\phi}_l(j)}{\max_m \bar{\phi}_l(m) - \min_m \bar{\phi}_l(m)} \right|, \quad j = 1, 2, \ldots, N(t) \tag{16}$$

where $N(t)$ denotes the number of information vectors stored in the database when the query $\bar{\phi}(t)$ is given. To further explain the variables: $\bar{\phi}_l(j)$ denotes the lth element of the jth information vector, and, similarly, $\bar{\phi}_l(t)$ denotes the lth element of the query at t. Among the lth element of all information vectors ($\bar{\phi}(j), j = 1, 2, \ldots, N(t)$) stored in the database, the maximum element is denoted by $\max \bar{\phi}_l(m)$. Similarly, $\min \bar{\phi}_l(m)$ denotes the minimum element. Here, k-pieces with the smallest distances between them are chosen from all information vectors.

[STEP 3] Compute PID Parameters: Next, using k-neighbors selected in [STEP 2], the suitable set of PID parameters is computed around the query using the following linearly weighted average (LWA) [14]:

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

where w_i denotes the weight corresponding to the ith information vector $\bar{\phi}(i)$ in the selected neighbors. This is calculated by

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

Using the PID parameters computed in (17), the control input is generated, and the output $y(t+1)$ is measured.

[STEP 4] PID Parameters Adjustment: In the case where the suitable control performance cannot be obtained using the PID parameters computed in [STEP 3], these control parameters have to be updated and stored in the database, that is, it is necessary to adjust the PID parameters so that the control error is decreased. The following steepest descent method is utilized in order to modify the PID parameters:

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

$$\boldsymbol{\eta} := \text{diag}\{\eta_P, \eta_I, \eta_D\} \tag{20}$$

where $\boldsymbol{\eta}$ denotes the learning rate, and the following $J(t+1)$ denotes the error criterion:

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

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

The output of the reference model is denoted by $y_r(t)$, 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 following two features:

1) rise-time;
2) damping property.

Thus, let $T(z^{-1})$ be the denominator of the discrete-time version of the following desirable second-order continuous-time transfer function $G(s)$:

$$G(s) = \frac{1}{1 + \sigma s + \mu(\sigma s)^2}. \tag{25}$$

In other words, $T(z^{-1})$ is defined as follows:

$$T(z^{-1}) = 1 + t_1 z^{-1} + t_2 z^{-2} \tag{26}$$

$$t_1 = -2\exp\left(-\frac{\rho}{2\mu}\right) \cos\left(\frac{\sqrt{(}2\mu - 1)}{2\mu}\rho\right) \tag{27}$$

$$t_2 = \exp\left(-\frac{\rho}{\mu}\right) \tag{28}$$

$$\rho := \frac{T_s}{\sigma} \tag{29}$$

$$\mu := 0.25(1-\delta) + 0.51\delta \tag{30}$$

where σ denotes the rise-time. The damping coefficient is denoted by μ. The damping coefficient itself is adjusted by changing δ. Where $\delta = 0$ and $\delta = 1$, the step response of $G(s)$ shows the Binominal model response and the Butterworth model response, respectively.

Moreover, each partial differential of (19) is developed as follows:

$$\left.\begin{aligned}
\frac{\partial J(t+1)}{\partial K_P(t)} &= \frac{\partial J(t+1)}{\partial \varepsilon(t+1)} \frac{\partial \varepsilon(t+1)}{\partial y(t+1)} \frac{\partial y(t+1)}{\partial u(t)} \frac{\partial u(t)}{\partial K_P(t)} \\
&= \varepsilon(t+1)(y(t) - y(t-1)) \frac{\partial y(t+1)}{\partial u(t)} \\
\frac{\partial J(t+1)}{\partial K_I(t)} &= \frac{\partial J(t+1)}{\partial \varepsilon(t+1)} \frac{\partial \varepsilon(t+1)}{\partial y(t+1)} \frac{\partial y(t+1)}{\partial u(t)} \frac{\partial u(t)}{\partial K_I(t)} \\
&= -\varepsilon(t+1)e(t) \frac{\partial y(t+1)}{\partial u(t)} \\
\frac{\partial J(t+1)}{\partial K_D(t)} &= \frac{\partial J}{\partial \varepsilon(t+1)} \frac{\partial \varepsilon(t+1)}{\partial y(t+1)} \frac{\partial y(t+1)}{\partial u(t)} \frac{\partial u(t)}{\partial K_D(t)} \\
&= \varepsilon(t+1)(y(t) - 2y(t-1) + y(t-2)) \frac{\partial y(t+1)}{\partial u(t)}.
\end{aligned}\right\} \tag{31}$$

Note that *a priori* information with respect to the system Jacobian $\partial y(t+1)/\partial u(t)$ is required in order to calculate (31). Here, using the relation $x = |x|\text{sign}(x)$, the system Jacobian can be obtained by the following equation:

$$\frac{\partial y(t+1)}{\partial u(t)} = \left|\frac{\partial y(t+1)}{\partial u(t)}\right| \text{sign}\left(\frac{\partial y(t+1)}{\partial u(t)}\right) \tag{32}$$

where $\text{sign}(x) = 1(x > 0), -1(x < 0)$. From the assumption mentioned above, the sign of the system Jacobian is obtainable. And by including $|\partial y(t+1)/\partial u(t)|$ in η, the usage of the system Jacobian can be made easy [3]. Therefore, it is assumed that the sign of the system Jacobian is known in this paper.

[STEP 5] Remove Redundant Data: In implementing the proposed DD technique in real systems, the newly proposed scheme has a constraint: the calculations between [STEP 2] and [STEP 4] must be finished within the sampling time. Here, storing the redundant data in the database results in the computer wasting time that could be used on the process. Therefore, an algorithm

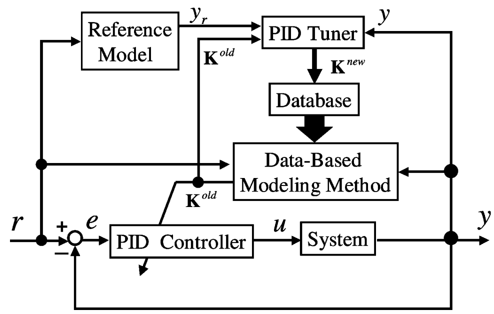


Fig. 1. Block diagram of the proposed data-driven PID control system.

to avoid excessive stored data is put forward. The procedure is carried out in the following two steps.

[**First condition**] For information vectors in which k-neighbors are excepted beforehand, the information vectors which satisfy the following condition are extracted:

$$d(\bar{\phi}(t), \bar{\phi}(i)) \leq \alpha_1, \quad i = 1, 2, \ldots, N(t) - k. \tag{33}$$

Note that the distance is computed using only input/output data $\bar{\phi}$ included in the information vector. Meeting the first condition means that only the information vectors with short distance to the query $\bar{\phi}(t)$, are extracted from the database.

[**Second condition**] For information vectors extracted in the first condition, the information vectors which satisfy the following condition are extracted:

$$\sum_{l=1}^{3} \left\{ \frac{\mathbf{K}_l(i) - \mathbf{K}_l^{new}(t)}{\mathbf{K}_l^{new}(t)} \right\}^2 \leq \alpha_2 \tag{34}$$

where $\mathbf{K}_1$, $\mathbf{K}_2$, and $\mathbf{K}_3$ mean K_P, K_I, and K_D, respectively. Using the above procedure, it is possible to extract and delete from the database redundant data. This is achieved by extracting and deleting from the database information vectors with high similarity in the relationship between the newly generated PID gains and PID gains included in the extracted information vectors. If plural information vectors exist which satisfy the second condition, then only the information vector with the smallest value in the second condition is removed.

Note that the query and the corresponding updated PID parameters are always stored in the database. In practice, α_1 and α_2 should be set between 0.1 and 1.0, but some trial and error may be necessary.

The above algorithm is illustrated by the block diagram in Fig. 1.

III. Simulation Examples

In order to evaluate the effectiveness of the newly proposed scheme, some simulation examples for nonlinear systems are considered. These include a Hammerstein model and a system with a hysteresis. They are discussed below.

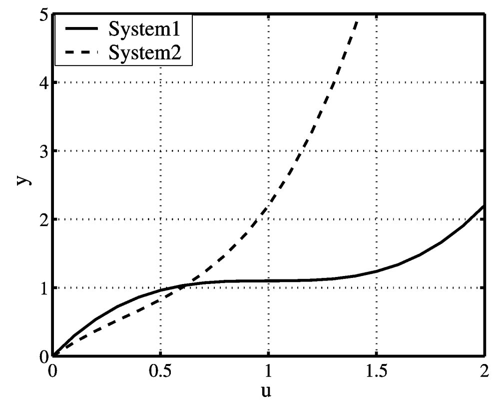


Fig. 2. Static properties of System 1 and System 2.

A. *Ex.1 Hammerstein Model*

To begin, the following Hammerstein models are discussed (see [15] for more information).

[**System 1**]

$$\left.\begin{aligned} y(t) &= 0.6y(t-1) - 0.1y(t-2) \\ &\quad +1.2x(t-1) - 0.1x(t-2) + \xi(t) \\ x(t) &= 1.5u(t) - 1.5u^2(t) + 0.5u^3(t) \end{aligned}\right\}. \tag{35}$$

[**System 2**]

$$\left.\begin{aligned} y(t) &= 0.6y(t-1) - 0.1y(t-2) \\ &\quad +1.2x(t-1) - 0.1x(t-2) + \xi(t) \\ x(t) &= 1.0u(t) - 1.0u^2(t) + 1.0u^3(t) \end{aligned}\right\} \tag{36}$$

where $\xi(t)$ denotes the white Gaussian noise with zero mean and a variance of 0.01^2. The static properties of the System 1 and System 2 are shown in Fig. 2. On examining Fig. 2, it is apparent that the gains of System 2 are larger than those of System 1 at $y \geq 1.0$.

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

$$r(t) = \begin{cases} 0.5(0 \leq t < 50) \\ 1.0(50 \leq t < 100) \\ 2.0(100 \leq t < 150) \\ 1.5(150 \leq t \leq 200). \end{cases} \tag{37}$$

The information vector $\bar{\phi}$ is given by

$$\bar{\phi}(t) := [r(t+1), r(t), y(t), y(t-1), y(t-2), u(t-1)]. \tag{38}$$

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

$$T(z^{-1}) = 1 - 0.271z^{-1} + 0.0183z^{-2} \tag{39}$$

where $T(z^{-1})$ was designed by setting σ as 1.0 and δ as 0.0. Here, T_s is equal to 1.0. Table I shows the user-specified parameters as determined in the proposed method.

For the purpose of comparing the proposed method with conventional schemes, the fixed PID control scheme widely used

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	$N(0) = 6$

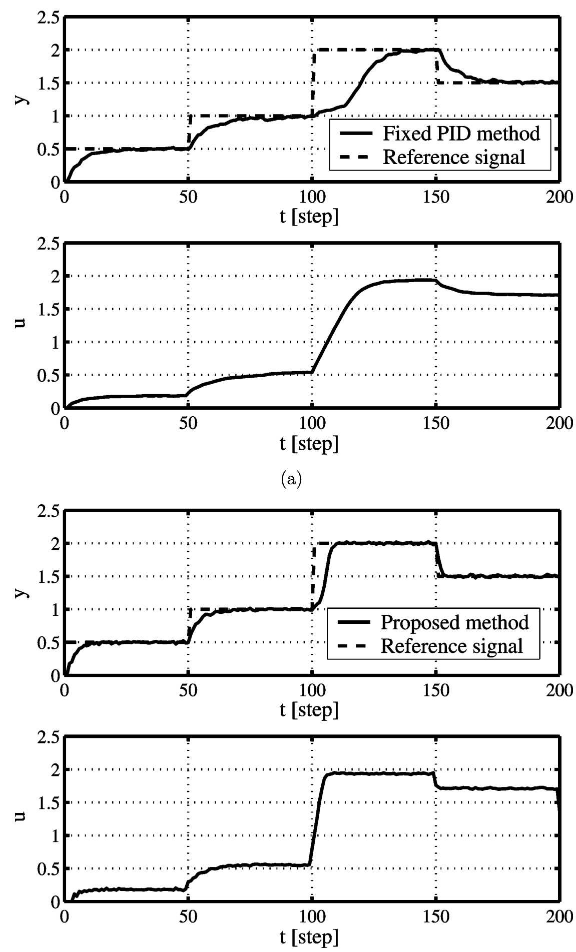


Fig. 3. Control results using (a) the fixed PID control and (b) the proposed method for System 1.

in industrial processes is first employed. These PID parameters, tuned by the CHR method [13], are as follows:

$$K_P = 0.486 \qquad K_I = 0.227 \qquad K_D = 0.122. \qquad (40)$$

In addition, a certain type of neural-net-based PID controller [3], called the NN-PID controller, was applied. This particular NN was used because it complements the fixed PID controller.

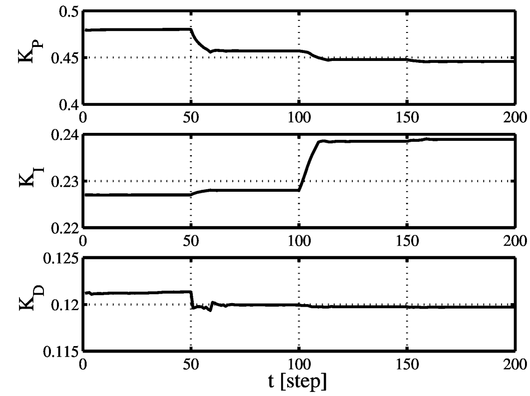


Fig. 4. Trajectories of PID parameters corresponding to Fig. 3.

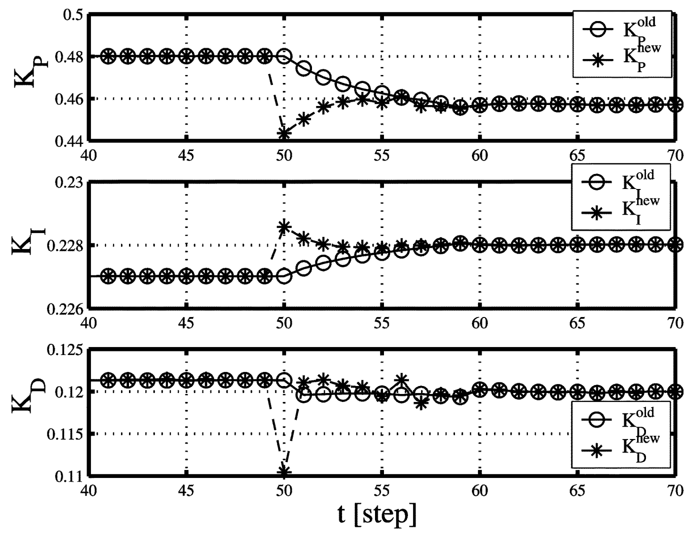


Fig. 5. Behavior of PID parameters adjustment around $t = 100$ [step].

The control results for System 1 [see (35)] are shown in Fig. 3 using the fixed PID controller and the proposed control scheme. Trajectories of the PID parameters in the proposed scheme are summarized in Fig. 4.

From Fig. 3, it is clear that the control results of the fixed PID controller are not good. This is owing to nonlinearities of the controlled object. The nonlinearities had the most influence in the third step where the control responses become slow. On the other hand, looking at Figs. 3 and 4, it is apparent that good control results can be obtained using the proposed method because PID parameters are adequately adjusted.

The adjustment of PID parameters merits further discussion. Fig. 5 shows the behavior of the updating scheme set out in [STEP 4], where the reference signal is changed from 0.5 thru 1.0 around 50 [step]. According to Fig. 5, it is clear that a new set of PID parameters is gradually built up in the database, and PID parameters are suitably adjusted according to the change of the reference signal or the operating point.

Moreover, $*$ in Fig. 6 shows the point in which a new information vector is stored in the database, that is, Fig. 6 shows that the process of removing the redundant data in [STEP 5] works adequately, and the new data (i.e., the new information vector) are stored instead of the redundant data. Fig. 6 also illustrates

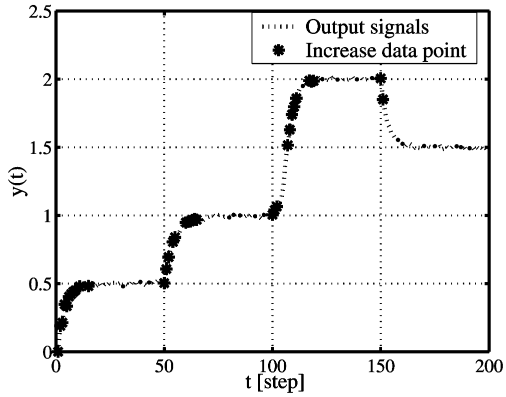


Fig. 6. Replacement behavior of the redundant data by the new data, where * shows the point in which a new information vector is stored in the database.

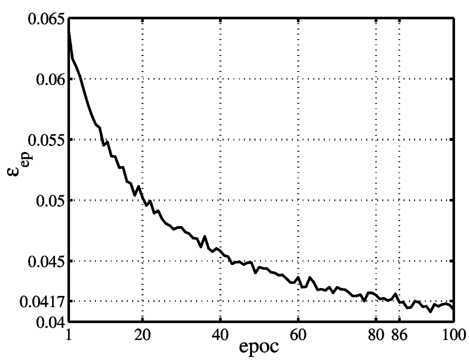


Fig. 7. Error behaviors using the NN-PID controller for Hammerstein model.

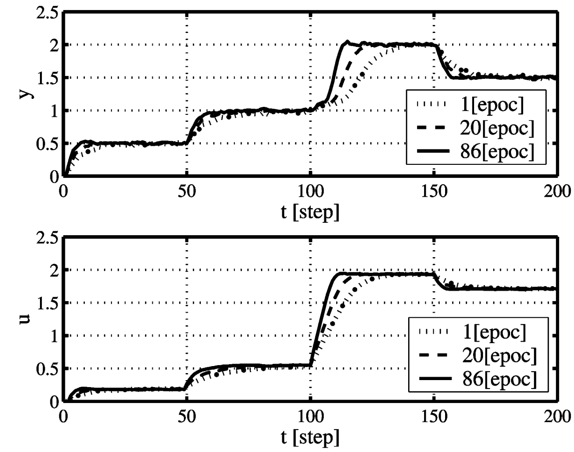


Fig. 8. Control result using the NN-PID controller for Hammerstein model.

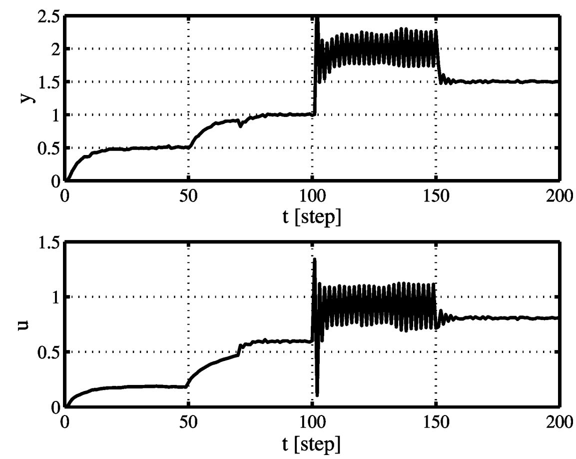


Fig. 9. Control result using the fixed PID controller in the case where the system is changed from System1 to System2.

that new data are stored only in the transient state where the reference signals are changed. Note that there is very little replacement of the redundant data by the new data when the system is in a steady state (i.e., where the reference signal is constant). As a result of using the algorithm in [STEP 5] to remove needless data, the number of data items stored in the database was quite low: 38, reduced from an original count of 206. In addition, using the proposed method, the error $\varepsilon_{\rm ep}$ given by the following criterion is 0.0417:

$$\varepsilon_{\rm ep}(epoc) := \frac{1}{N}\sum_{t=1}^{N}\left\{\frac{\varepsilon(t)}{r(t)}\right\}^2 \tag{41}$$

where N denotes the number of steps per 1 [epoc], and N is set as 200 in this case. Furthermore, because PID parameters can be adjusted in an online manner by the proposed method, the number of iterations was set as 1.

Moreover, the NN-PID controller [3] was applied to this system [see (35)], where the NN was utilized in order to supplement the fixed PID controller. The trajectory of the learning error $\varepsilon_{\rm ep}$ [expressed in (41)] is shown in Fig. 7, and control results are shown in Fig. 8. We see from Fig. 7 that the number of learning iterations remains 86 [epoc] until the control results using the NN-PID controller can produce the same control performances as the proposed method. In other words, until $\epsilon_{\rm ep} \leq 0.0417$ is satisfied. Since this condition has been satisfied, we can go ahead and examine the effectiveness of the proposed method nonlinear systems.

The next step is to consider the case where the system changes from (35) to (36) at $t = 70$. First, the control results using the fixed PID controller are shown in Fig. 9, where the PID parameters used are the same as those shown in (40). Although the gain of the controlled object becomes high after $t = 70$, the PID parameters are not changed. Therefore, the control performance becomes oscillatory after $t = 100$. On the other hand, the proposed control scheme was also employed in this case. The control results and trajectories of PID parameters are shown in Figs. 10 and 11, respectively. Figs. 10 and 11 demonstrate that, even if system parameters are changed, a fairly good control performance can be obtained because PID parameters are adjusted adequately using the proposed method. This result shows the adaptability of the proposed method remarkably well in the Hammerstein model. Next, the proposed method was applied to another type of nonlinear system: a system with a hysteresis.

B. *Ex.2 System With a Hysteresis*

Next, the following model with another nonlinear type of system was tested. In this system, like that set out by Narendra and Parthasarathy [16], the following equation applies:

$$y(t+1) = \frac{y(t)y(t-1)[y(t)+2.5]}{1+y^2(t)+y^2(t-1)} + u(t) + \xi(t) \tag{42}$$

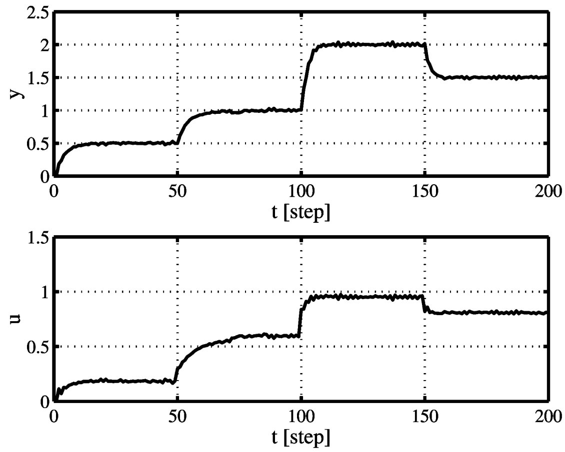


Fig. 10. Control result using the proposed method in the case where the system parameters are changed.

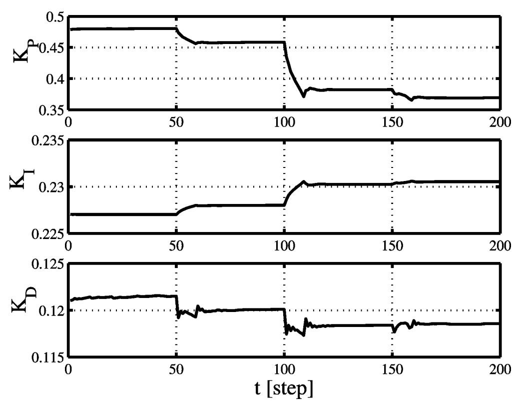


Fig. 11. Trajectories of PID parameters corresponding to Fig. 10.

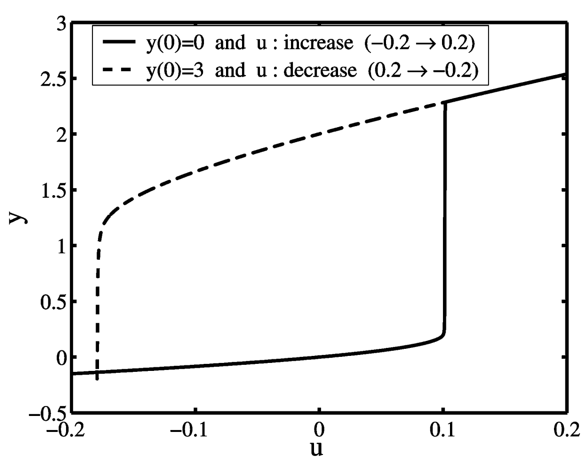


Fig. 12. Static property of the system (42).

where $\xi(t)$ denotes the white Gaussian noise with zero mean and a variance of 0.01^2. The static property of this model is shown in Fig. 12 and discussed below. From Fig. 12, it is clear that this model has strong nonlinearities. In particular, there is a kind of hysteresis occurring between $y = 0$ and $y = 2.0$. Here, the reference signal $r(t)$ was given by

$$r(t) = \begin{cases} 1.5(0 \leq t < 100) \\ 0.8(100 \leq t < 200) \\ 2.5(200 \leq t < 300) \\ -1.0(300 \leq t \leq 400). \end{cases} \tag{43}$$

TABLE II
USER-SPECIFIED PARAMETERS INCLUDED IN THE PROPOSED METHOD

Orders of the information vector	$n_y = 3$ $n_u = 2$
Number of neighbors	$k = 6$
Learning rates	$\eta_P = 0.6$ $\eta_I = 0.03$ $\eta_D = 0.01$
Coefficients to inhibit the data	$\alpha_1 = 0.1$ $\alpha_2 = 0.3$
Initial number of data	$N(0) = 6$

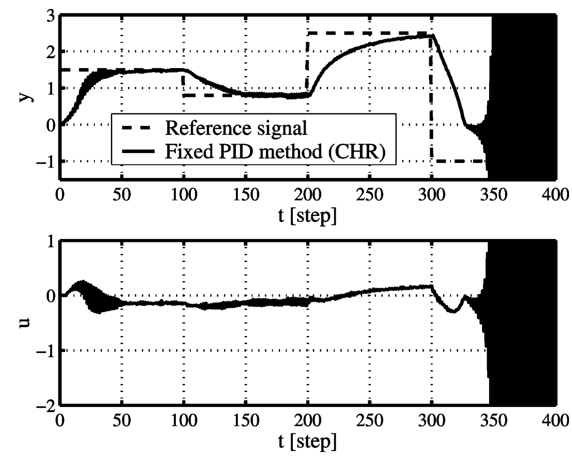


Fig. 13. Control result using the fixed PID controller for Ex.2.

Equation (38) was utilized as the information vector $\bar{\phi}(t)$, and the desired characteristic polynomial $T(z^{-1})$ included in the reference model was designed as

$$T(z^{-1}) = 1 - 0.271z^{-1} + 0.0183z^{-2} \tag{44}$$

where σ and δ are set as 1.0 and 0.0, respectively, and T_s equals 1.0. In addition, the user-specified parameters included in the proposed method were determined and are summarized in Table II.

For the purpose of comparison, the fixed PID control scheme was first employed, wherein the PID parameters were tuned by the CHR method [13]. Then, PID parameters were calculated as

$$K_P = 0.654 \qquad K_I = 0.028 \qquad K_D = 0.327. \tag{45}$$

The control results are shown in Fig. 13. As seen in Fig. 13, due to nonlinearities of the controlled object, the control results using the fixed PID controller include excessive oscillation around $y = 0.0$. The consequence is that the system falls into an unstable state.

Next, the self-tuning PID control scheme was employed, and the control results are shown in Fig. 14. The self-tuning PID control scheme examined in this paper has been previously considered in [2]. For details, see Yamamoto and Shah [2]. Thus, the system parameters were recursively estimated and the PID

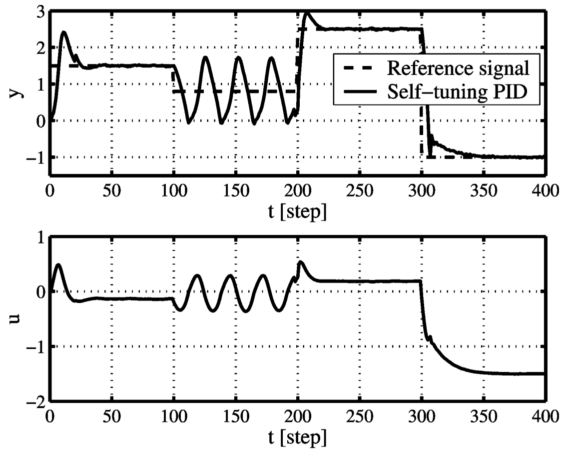


Fig. 14. Control result using the self-tuning PID controller for Ex.2.

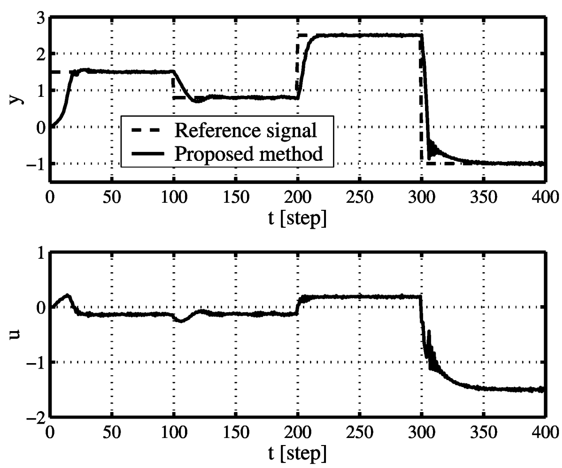


Fig. 15. Control result using the proposed method for Ex.2.

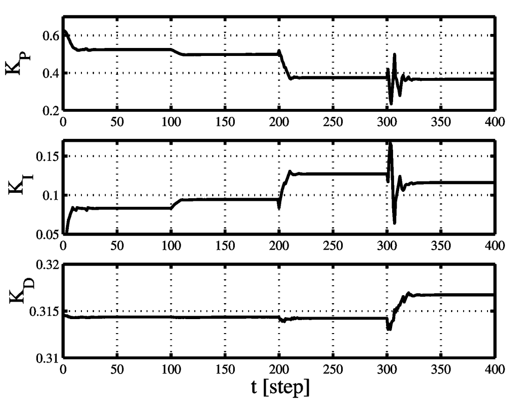


Fig. 16. Trajectories of PID parameters corresponding to Fig. 15.

parameters were computed using the estimates based on the relationship between the generalized minimum variance control and PID control. Due to the strong nonlinearity, the control performance was inferior in the second reference signal, that is, $r(t) = 0.8$.

Finally, the control results using the proposed scheme are shown in Fig. 15, and then trajectories of PID parameters are shown in Fig. 16. Figs. 15 and 16 show that the domination of the proposed method is quite clear. The PID adjustment in an online manner works exceptionally well and removes the excessive oscillation.

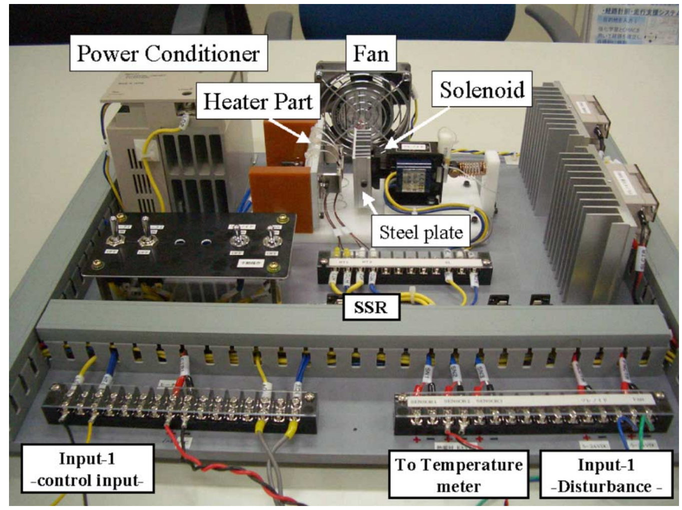


Fig. 17. Photograph of the experimental temperature control system.

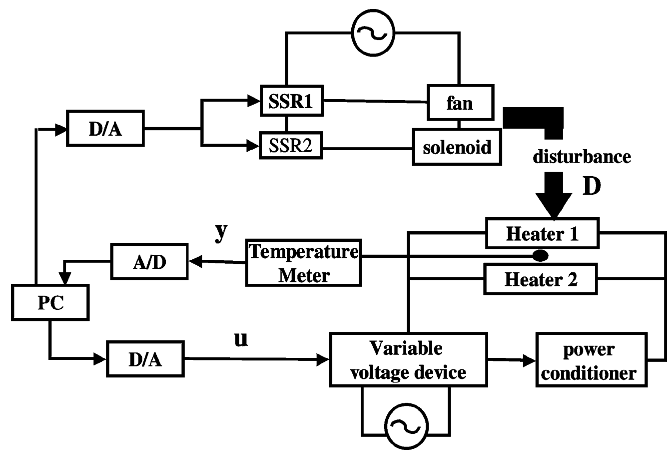


Fig. 18. Schematic figure of the experimental temperature control system.

IV. Experimental Results

The proposed DD method was experimentally evaluated on a pilot-scale temperature control system. Fig. 17 shows a photograph of this equipment, and the corresponding schematic figure of this system is illustrated in Fig. 18.

Two heaters were secured on a steel plate. These heaters worked synchronously, corresponding to the input signal from the computer. One thermo-couple was also prepared on the steel plate, and the measured temperature of the steel plate was sent to the computer as the system output signal. The control objective was to regulate the temperature of the steel to the desired reference signal by manipulating the power of the heater. The static properties of the temperature control system are shown in Fig. 19. Especially, a system gain of around 100° in the temperature is about a third of the temperature around 60°. The system gain changes drastically during the control.

The fixed PID controller and the proposed method were employed for this system. The reference signal was changed alternately from 60° in the temperature and 100°. The fixed PID parameters were computed using the CHR method [13] as follows:

$$K_P = 1.20 \qquad K_I = 0.0120 \qquad K_D = 0.60. \tag{46}$$

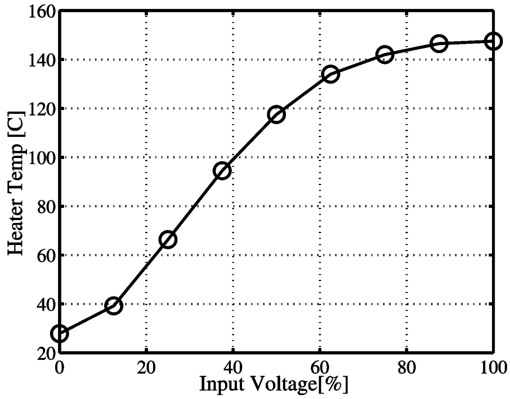


Fig. 19. Static properties of the temperature control system.

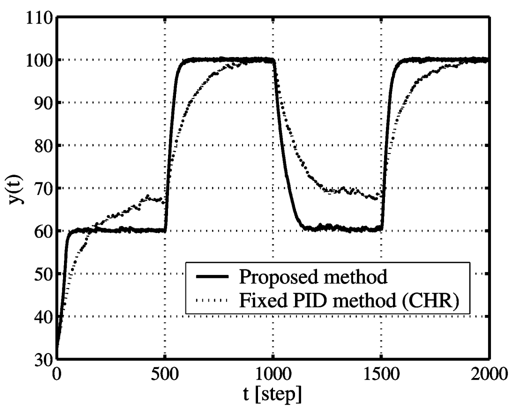


Fig. 20. Control result using the fixed PID controller(dotted line) and the proposed controller(solid line).

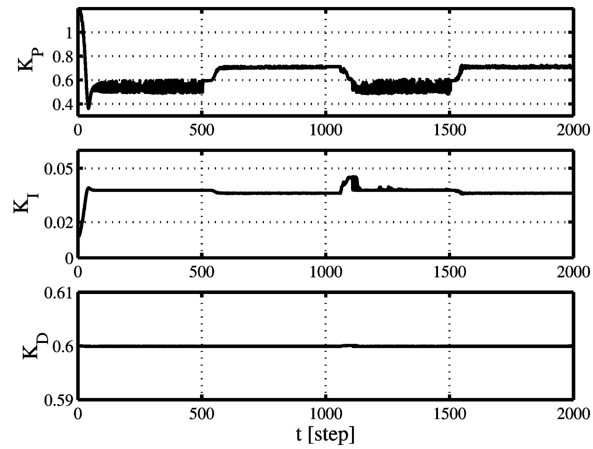


Fig. 21. Trajectories of PID parameters corresponding to Fig. 20.

TABLE III
USER-SPECIFIED PARAMETERS INCLUDED IN THE PROPOSED METHOD (TEMPERATURE CONTROLLER)

Sampling time	$T_s = 1$
Orders of the information vector	$n_y = 3$
	$n_u = 2$
Number of neighbors	$k = 10$
Learning rates	$\eta_P = K_P/100$
	$\eta_I = K_I/100$
	$\eta_D = K_D/10^4$
Coefficients to inhibit the data	$\alpha_1 = 0.5$
	$\alpha_2 = 0.1$
Initial number of data	$N(0) = 10$

Fig. 20 shows the control results. The dotted line and the solid line show the results using the fixed PID and the proposed method, respectively. The above PID parameters were determined using the historical data gathered when the system output was around 80° in temperature. Because the PID parameters were not suitably calculated, the control results are not so good. In particular, due to the small integral gain, the tracking property to the reference signal is considerably inferior.

On the other hand, according to the proposed method, PID parameters were appropriately adjusted in an online manner according to the reference signal as shown in Fig. 21. Here, the user-specified parameters included in the proposed method were determined as shown in Table III, and PID parameters in the initial database were set the same as in (46).

Next, the robustness and the adaptability to cyclic disturbances was investigated. Key to this investigation is the solenoid coil (see Fig. 17 or Fig. 18 which is part of the apparatus in this experiment). Because this solenoid works only periodically, it creates a cyclic disturbance in the following way. The solenoid is attached to the second steel plate which comes into contact with the main steel plate. It returns to its original position when the temperature is kept constant. This results in the main steel plate acting as the periodic disturbance. In other words, when the solenoid and the steel plate come in contact, the temperature of the main steel plate falls. The temperature rises if it returns.

This cycle of temperature rising and dropping was created as an imitation of a common industrial process wherein the object is first put on the main steel board, and the object is processed while keeping the temperature constant. It moves to the next process when the processing ends, and the next new object is put on the main steel board. This procedure is repeated many times. As a result, if the temperature recovers quickly, a lot of objects can be processed. Therefore, the tracking properties of the PID controller are strongly demanded in industries.

For the purpose of comparison, the fixed PID controller and the proposed method were employed for the case where the periodic disturbance is introduced. The control results are summarized in Fig. 22: the dotted line shows the result given by the fixed PID controller and the solid line shows the proposed method. Moreover, Fig. 23 shows the trajectories of PID parameters created by the proposed method. Here, the fixed PID parameters were computed using the CHR method [13] as follows:

$$K_P = 0.12 \qquad K_I = 0.0024 \qquad K_D = 0.06. \tag{47}$$

Also, the user-specified parameters included in the proposed method utilized were the same as those given in Table III. It is clear that the tracking property to the reference signal is gradually improved according to the proposed method. The behavior is shown more clearly in Figs. 24 and 25, where these figures

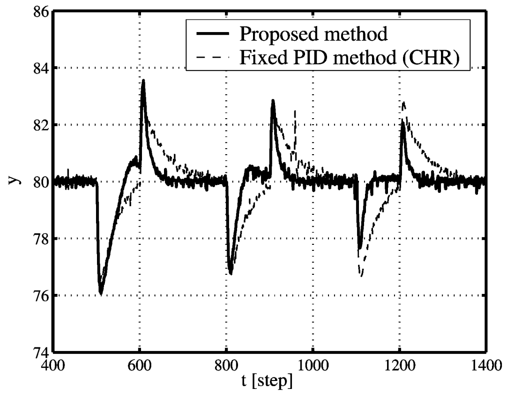


Fig. 22. Control result using the fixed PID controller(dotted line) and the proposed control method(solid line) in the case where the disturbance is periodically added.

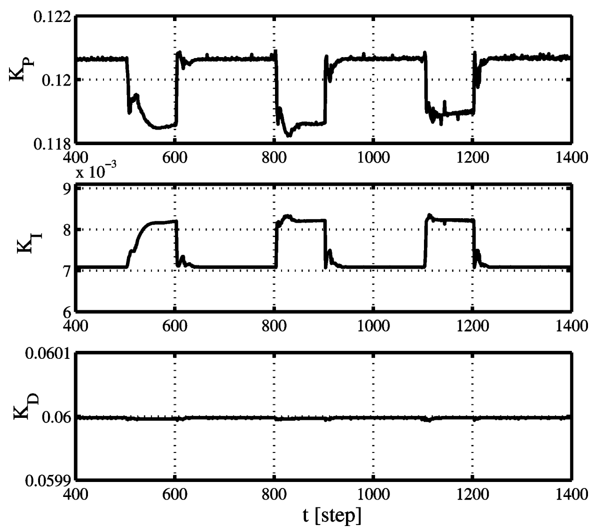


Fig. 23. Trajectories of PID parameters corresponding to Fig. 22.

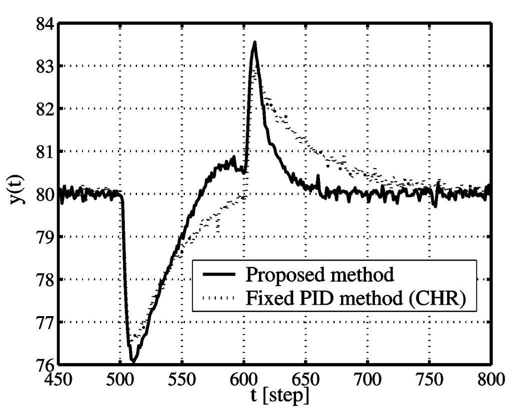


Fig. 24. Enlarged figure around $t = 500$ in Fig. 22.

show the control results around $t = 500$ and $t = 1100$, respectively. For the disturbance, another set of PID parameters is immediately extracted from the database corresponding to the current state. It is clear that the proposed method effectively uses the previously accumulated knowledge. This is one of the typical features of the proposed method.

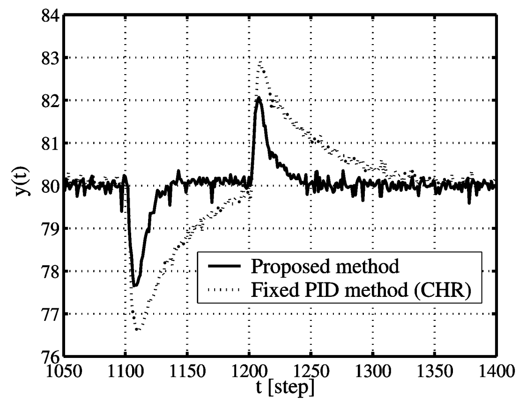


Fig. 25. Enlarged figure around $t = 1100$ in Fig. 22.

V. Conclusion

In this paper, a new design scheme of PID controllers using the DB modeling method has been proposed. To date, many PID controller design schemes using NNs and GAs have been proposed for nonlinear systems. In employing these schemes for real systems, however, the considerably large learning cost becomes a serious problem. This problem can be avoided by using the proposed method because such computational burdens can be effectively reduced by using the algorithm for removing the redundant data. In addition, the effectiveness of the proposed method has been verified by some numerical simulation examples. Given the success of this newly proposed DD technique for PID controllers on the temperature control system, there is good reason to continue to explore its success on other systems. Applications of the newly proposed scheme for existing systems and the extension to multivariable cases are currently under consideration.

References

[1] K. J. Åström and T. Hägglund, *Automatic Tuning of PID Controllers*. Research Triangle Park, NC: Instrum. Soc. of America, 1988.

[2] T. Yamamoto and S. L. Shah, "Design and experimental evaluation of a multivariable self-tuning PID controller," *IEE Proc. Control Theory Appl.*, vol. 151, no. 5, pp. 645/652–645/652, 2004.

[3] S. Omatu, K. Marzuki, and Y. Rubiyah, *Neuro-Control and Its Applications*. London, U.K.: Springer-Verlag, 1995.

[4] B. Porter and A. H. Jones, "Genetic tuning of digital PID controllers," *Electron. Lett.*, vol. 28, pp. 843/844–843/844, 1992.

[5] A. Stenman, F. Gustafsson, and L. Ljung, "Just in time models for dynamical systems," in *Proc. 35th IEEE Conf. Decision Control*, 1996, pp. 1115/1120–1115/1120.

[6] Q. Zheng and H. Kimura, "A new just-in-time modeling method and its applications to rolling set-up modeling," (in Japanese) *Trans. SICE*, vol. 37, no. 7, pp. 640/646–640/646, 2001.

[7] J. Zhang, Y. Yim, and J. Yang, "Intelligent selection of instances for prediction functions in lazy learning algorithms," *Artif. Intell. Rev.*, vol. 11, pp. 175/191–175/191, 1997.

[8] G. Bontempi, M. Birattari, and H. Bersini, "Lazy learning for local modeling and control design," *Int. J. Control*, vol. 72, no. 7-8, pp. 643/658–643/658, 1999.

[9] A. Stenman, "Model on demand: Algorithms, analysis and applications," Ph.D. dissertation, Dept. Elect. Eng., Linköping University, Linköping, Sweden, 1990.

[10] Q. Zheng and H. Kimura, "Just-in-time PID control," in *Proc. 44th Japan Joint Automatic Control Conf.*, Tokyo, Japan, 2001, pp. 336–339.

[11] J. Ohta and S. Yamamoto, "Detabase-driven tuning of PID controllers," (in Japanese) *Trans. SICE*, vol. 40, no. 6, pp. 664–669, 2004.

[12] J. G. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," *Trans. ASME*, vol. 64, no. 8, pp. 759/768–759/768, 1942.
[13] K. L. Chien, J. A. Hrones, and J. B. Reswick, "On the automatic control of generalized passive systems," *Trans. ASME*, vol. 74, pp. 175/185–175/185, 1972.
[14] C. G. Atkeson, A. W. Moore, and S. Schaal, "Locally weighted learning for control," *Artif. Intell. Rev.*, vol. 11, pp. 75/114–75/114, 1997.
[15] L. Zi-Qiang, "On identification of the controlled plants described by the hammerstein system," *IEEE Trans. Automat. Contr.*, vol. AC-39, no. 2, pp. 569–573, Apr. 1994.
[16] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Trans. Neural Netw.*, vol. 1, no. 1, pp. 4/27–4/27, Feb. 1990.

Toru Yamamoto (M'92) received the B.Eng. and M.Eng. degrees from the University of Tokushima, Tokushima, Japan, in 1984 and 1987, respectively, and the D.Eng. degree from Osaka University, Osaka, Japan, in 1994.

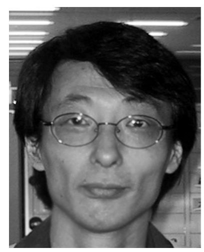
He is currently a Professor with the Department of Technology and Information Education, Hiroshima University, Higashi-Hiroshima, Japan. He was a Visiting Researcher with the Department of Mathematical Engineering and Information Physics, University of Tokyo, Tokyo, Japan, in 1991, and, during the autumn of 1996, he was a Visiting Professor with the Department of Chemical and Materials Engineering, University of Alberta, Edmonton, AB, Canada. In addition, he was an Overseas Research Fellow of the Japan Society for Promotion of Science with the University of Alberta for six months in 2006. His current research interests are in the area of self-tuning control, intelligent control, and process control.

Dr. Yamamoto won the Best Poster Paper Award at the 3rd Asian Control Conference in 2000 and was the recipient of the Prize of Progress from the Institute of Electrical Engineers in Japan and the Technology Award from the Society of Instrumentation and Control Engineers in Japan in 2003.

Kenji Takao received the M.Eng. and D.Eng degrees from Hiroshima University, Higashi-Hiroshima, Japan, in 2004 and 2007, respectively.

He is currently a Research Engineer with the Technical Headquarters, Mitsubishi Heavy Industries, Ltd., Mihara, Japan. He was a Research Fellow of the Japan Society for the Promotion of Science for two years starting in April 2005. His current research interests are in the area of PID control, intelligent control, process control, and data-based modeling/control.

Dr. Takao is a member of the Society of Instrumentation and Control Engineers in Japan and the Institute of Systems, Control and Information Engineers in Japan.



Takaaki Yamada received the B.Eng. degree from Osaka University, Osaka, Japan, in 1985.

He joined in the Industrial Automation Company, OMRON Corporation, Kusatsu, Japan, in 1985. His work is mainly in the field of development of temperature control.

Mr. Yamada is a member of the Society of Instrumentation and Control Engineers in Japan (SICE) and the Information Processing Society of Japan. He received a license of instrument and control engineers from SICE in 2005.