8th International IFAC Symposium on
Dynamics and Control of Process Systems
Preprints Vol.2, June 6-8, 2007, Cancún, Mexico

DESIGN AND EXPERIMENTAL EVALUATION OF A DATA-DRIVEN PID CONTROLLER

Toru Yamamoto * **Kenji Takao** *

* *Hiroshima University, Higashi-Hiroshima, Hiroshima, 739-8524, Japan*

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 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 data-base. This scheme can adjust the PID parameters in an on-line 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 a pilot-scale temperature control system.

Keywords: PID control, data-base, nonlinear systems, process systems, self-adjust systems

1. 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. However, in process industries, PID controllers(Åström and Hägglund, 1988; Chien, *et al.*, 1972) have been widely employed for about 80 percent 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.

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 or the local controller is constructed using these neighbors. This data-driven(DD) technique, is called *Just-In-Time (JIT)* method(Stenman, *et al.*, 1996), *Lazy Learning* method(Zhang, *et al.*, 1997) or Model-on-Demand(MoD)(Stenman, 1990), and these schemes have lots of attention in last decade.

This paper 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 data-base. Therefore, more suitable PID parameters corresponding to characteristics of the controlled ob-

[1] This work was partially supported by the Grant-in-Aid for Scientific Research(B)(No.1736019900) in Japan Society for the Promotion of Science

ject 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 experimentally evaluated on a pilot-scale temperature control system.

2. DATA-DRIVEN PID CONTROLLER DESIGN

2.1 DD technique

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:

$$\begin{aligned}\phi(t) : \quad &= [y(t-1), \cdots, y(t-n_y), \\ &\qquad u(t-1), \cdots u(t-n_u)],\end{aligned} \tag{2}$$

where $u(t)$ denotes the control input. Also, n_y and n_u respectively denote the orders of the system output and the control input, respectively. According to the DD technique, 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 prediction of the system can be obtained using these neighbors.

2.2 Controller design

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

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

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

$$e(t) := r(t) - y(t). \tag{4}$$

$r(t)$ 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.(3) 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.(3) are fixed. Therefore, a new control scheme is proposed, which can adjust PID parameters in an on-line manner corresponding to properties of the system. Thus, instead of Eq.(3), the following PID control law with time-variant PID parameters is employed:

$$\begin{aligned}\Delta u(t) = K_I(t) e(t) - K_P(t) \Delta y(t) \\ - K_D(t) \Delta^2 y(t).\end{aligned} \tag{5}$$

Now, Eq.(5) can be rewritten as the following relations:

$$u(t) = g(\phi'(t)) \tag{6}$$

$$\begin{aligned}\phi'(t) := [&\mathbf{K}(t), r(t), y(t), y(t-1), \\ &y(t-2), u(t-1)]\end{aligned} \tag{7}$$

$$\mathbf{K}(t) := [K_P(t), K_I(t), K_D(t)], \tag{8}$$

where $g(\cdot)$ denotes a function. By substituting Eq.(6) and Eq.(7) into Eq.(1) and Eq.(2), the following equation can be derived:

$$y(t+1) = h(\tilde{\phi}(t)) \tag{9}$$

$$\begin{aligned}\tilde{\phi}(t) := [\ &y(t), \cdots, y(t-n_y+1), \mathbf{K}(t), r(t), \\ &u(t-1), \cdots, u(t-n_u+1)\ ],\end{aligned} \tag{10}$$

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:

$$\mathbf{K}(t) = F(\bar{\phi}(t)) \tag{11}$$

$$\begin{aligned}\bar{\phi}(t) := [\ &y(t+1), y(t), \cdots, y(t-n_y+1), \\ &r(t), u(t-1), \cdots, u(t-n_u+1)\ ],\end{aligned} \tag{12}$$

where $F(\cdot)$ denotes a function. Since the future output $y(t+1)$ included in Eq.(12) cannot be obtained at t, $y(t+1)$ is replaced by $r(t+1)$. Because the purpose of control to be considered in this paper is to realize $y(t+1) \rightarrow r(t+1)$. Therefore, $\bar{\phi}(t)$ included in Eq.(12) is newly rewritten as follows:

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

After the above preparation, a new PID control scheme is designed 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 is not saved at all. Therefore, PID parameters are firstly calculated using Chien, Hrones

& Reswick(CHR) method[2] based on historical data of the controlled object in order to generate the initial data-base. Of course, PID parameters determined using operators' skill and/ or know-how can be also utilized as the initial data-base. That is, $\mathbf{\Phi}(j)$ indicated in the following equation is generated as the initial data-base:

$$\mathbf{\Phi}(j) := [\ \bar{\phi}(j)\ ,\ \mathbf{K}(j)\ ]\ ,\ j = 1, 2, \cdots, N(0) \quad (14)$$

where $\bar{\phi}(j)$ and $\mathbf{K}(j)$ are given by Eq.(13) and Eq.(8). Moreover, $N(0)$ denotes the number of information vectors stored in the initial data-base. If typical one set of fixed PID parameters is chosen, all PID parameters included in the initial information vectors may be 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 $\bar{\phi}(i)(i \neq t)$ are calculated using the following $\mathcal{L}_1$-norm with some weights:

$$d(\bar{\phi}(t), \bar{\phi}(j)) = \sum_{l=1}^{n_y+n_u+1} \left| \frac{\bar{\phi}_l(t) - \bar{\phi}_l(j)}{\max_m \bar{\phi}_l(m) - \min_m \bar{\phi}_l(m)} \right|, \quad (15)$$

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

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] 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)(Atkeson, *et al.*, 1997):

$$\mathbf{K}^{old}(t) = \sum_{i=1}^{k} w_i \mathbf{K}(i), \qquad \sum_{i=1}^{k} w_i = 1, \quad (16)$$

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

$$w_i = \sum_{l=1}^{n_u+n_y+1} \left(1 - \frac{[\bar{\phi}_l(t) - \bar{\phi}_l(i)]^2}{[\max_m \bar{\phi}_l(m) - \min_m \bar{\phi}_l(m)]^2} \right). \quad (17)$$

[STEP 4] PID parameters adjustment

In the case where information vectors corresponding to the current operating state of the controlled object is not effectively saved in the data-base, a suitable set of PID parameters cannot be computed. 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:

$$\mathbf{K}^{new}(t) = \mathbf{K}^{old}(t) - \eta \frac{\partial J(t+1)}{\partial \mathbf{K}(t)} \quad (18)$$

$$\eta := [\eta_P, \eta_I, \eta_D], \quad (19)$$

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

$$J(t+1) := \frac{1}{2}\varepsilon(t+1)^2 \quad (20)$$

$$\varepsilon(t) := y_r(t) - y(t). \quad (21)$$

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

$$y_r(t) = \frac{z^{-1}T(1)}{T(z^{-1})} r(t) \quad (22)$$

$$T(z^{-1}) := 1 + t_1 z^{-1} + t_2 z^{-2}. \quad (23)$$

Here, $T(z^{-1})$ is designed based on the reference(Takao, *et al.*, 2004). Moreover, each partial differential of Eq.(18) is developed as follows:

$$\left.\begin{aligned}
\frac{\partial J(t+1)}{\partial K_P(t)} &= \frac{\partial J(t+1)}{\partial \varepsilon(t+1)} \frac{\partial \varepsilon(t+1)}{\partial y(t+1)} \frac{\partial y(t+1)}{\partial u(t)} \frac{\partial u(t)}{\partial K_P(t)} \\
&= \varepsilon(t+1)(y(t) - y(t-1)) \frac{\partial y(t+1)}{\partial u(t)} \\
\frac{\partial J(t+1)}{\partial K_I(t)} &= \frac{\partial J(t+1)}{\partial \varepsilon(t+1)} \frac{\partial \varepsilon(t+1)}{\partial y(t+1)} \frac{\partial y(t+1)}{\partial u(t)} \frac{\partial u(t)}{\partial K_I(t)} \\
&= -\varepsilon(t+1) e(t) \frac{\partial y(t+1)}{\partial u(t)} \\
\frac{\partial J(t+1)}{\partial K_D(t)} &= \frac{\partial J}{\partial \varepsilon(t+1)} \frac{\partial \varepsilon(t+1)}{\partial y(t+1)} \frac{\partial y(t+1)}{\partial u(t)} \frac{\partial u(t)}{\partial K_D(t)} \\
&= \varepsilon(t+1)(y(t) - 2y(t-1) + y(t-2)) \frac{\partial y(t+1)}{\partial u(t)}.
\end{aligned}\right\} \quad (24)$$

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.(24). Here, using the relation $x = |x|\text{sign}(x)$, the system Jacobian can be obtained by the following equation:

$$\frac{\partial y(t+1)}{\partial u(t)} = \left| \frac{\partial y(t+1)}{\partial u(t)} \right| \text{sign}\left(\frac{\partial y(t+1)}{\partial u(t)} \right), \quad (25)$$

where $\text{sign}(x) = 1 (x > 0), -1 (x < 0)$. 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. 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 between STEP 2 and 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 vector $\mathbf{\Phi}(\bar{i})$ which satisfies the following first condition, is extracted from the data-base:

(i)First condition

$$d(\bar{\phi}(t), \bar{\phi}(i)) \leq \alpha_1, \quad i = 1, 2, \cdots, N(t) - k \quad (26)$$

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

$$\mathbf{\Phi}(\bar{i}) := \left[\bar{\phi}(\bar{i})\ \mathbf{K}(\bar{i})\right]. \quad \bar{i} = 1, 2, \cdots \quad (27)$$

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

(ii)Second condition

$$\sum_{l=1}^{3} \left\{ \frac{\mathbf{K}_l(\bar{i}) - \mathbf{K}_l^{new}(t)}{\mathbf{K}_l^{new}(t)} \right\}^2 \leq \alpha_2, \quad (28)$$

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

$$\mathbf{\Phi}(\hat{i}) := \left[\bar{\phi}(\hat{i}),\ \mathbf{K}(\hat{i})\right]. \quad \hat{i} = 1, 2, \cdots \quad (29)$$

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

Here, the above algorithm can be visually illustrated by the block diagram as shown in Fig.1.

3. EXPERIMENTAL RESULTS

The proposed method is experimentally evaluated on a pilot-scale temperature control system. The photograph of this equipment is shown in Fig.2. Furthermore, the schematic figure of this system is illustrated in Fig.3.

Two heaters are equipped on the steel plate, and these heaters work synchronously corresponding to the input signal from the computer. one

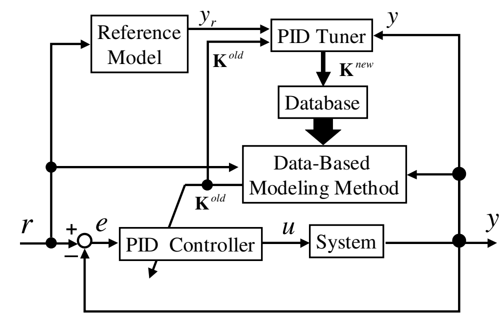

Fig. 1. Block diagram of the proposed data-driven PID control system.

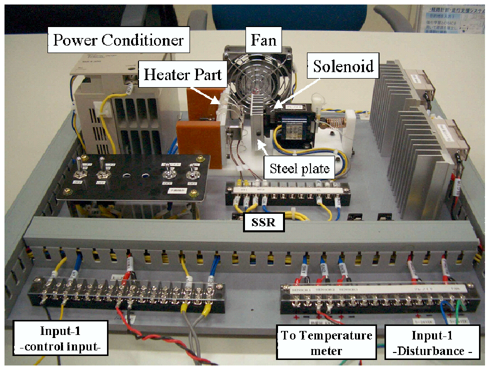

Fig. 2. Photograph of the experimental temperature control system.

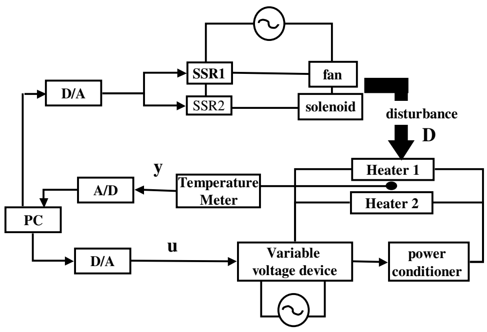

Fig. 3. Schematic figure of the experimental temperature control system.

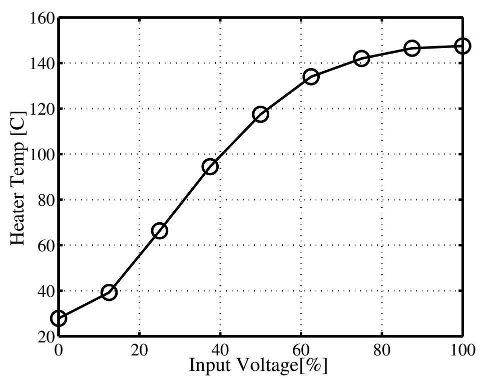

Fig. 4. Static properties of the temperature control system.

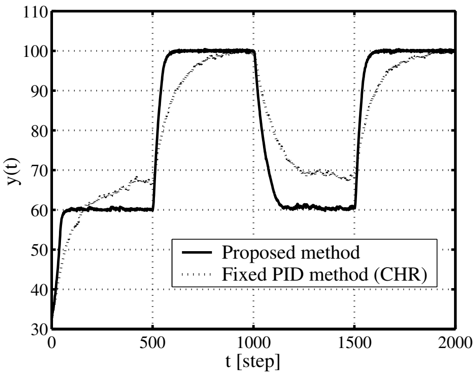

Fig. 5. Control result using the fixed PID controller(dotted line) and the proposed controller(solid line).

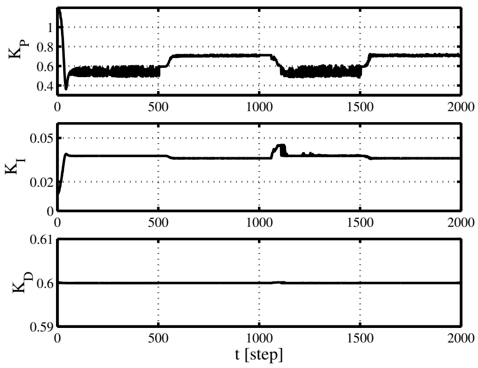

Fig. 6. Trajectories of PID parameters corresponding to Fig.5.

thermo-couple is also prepared on the steel plate, and the temperature of the steel plate measured is sent to the computer as the system output signal. The control objective is to regulate the temperature of the steel to the desired reference signal by manipulating the power of the heater. The static property is shown in Fig.4. This system has a kind of nonlinear properties. Especially, the system gain around 100 degrees in the temperature is about a third of the temperature around degrees 60 degrees. Therefore, 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 degrees in the temperature and 100 degrees. The fixed PID parameters were computed using CHR method(Chien, *et al.*, 1972) as follows.

$$K_P = 1.20, \quad K_I = 0.0120, \quad K_D = 0.60. \quad (30)$$

Fig.5 shows the control results. The dotted line and the solid line shows respectively the results using the fixed PID and the proposed method. The above PID parameters were determined using the historical data around 80 degrees in the temperature. Because PID parameters are not suitably calculated, the control result is not so good. Especially, 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 on-line manner according to the reference signal as shown in Fig.6. Here, the user-specified parameters included in the proposed method were determined as shown in Table 1, and PID parameters in the initial database were set as the same as Eq.(30).

Table 1. 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	$N(0) = 10$

Next, the robustness and the adaptability for cyclic disturbances are investigated. The solenoid in Fig.2 or Fig.3 works periodically. The solenoid equipped with the another steel plate comes in contact with the main steel plate and returns when the temperature is kept constant. This operation gives the main steel plate the influence as the periodic disturbance. In other words, if it comes in contact, the temperature of the main steel plate falls down, and the temperature rises up if it returns. This is an imitation of one operation in the industrial process. That is, the object is firstly 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. At this time, if the temperature recovers early, a lot of objects can be processed. Therefore, the tracking property is 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 put. The control results are summarized in Fig.7, where the dotted line and the solid line show the result by the fixed PID controller and the proposed method, respectively. Moreover, Fig.8 shows the trajectories of PID parameters by the proposed method. Here, the fixed PID parameters were computed using CHR method(Chien, *et al.*, 1972) as follows.

$$K_P = 0.12, \quad K_I = 0.0024, \quad K_D = 0.06. \quad (31)$$

And also, the user-specified parameters included in the proposed method were utilized as the same

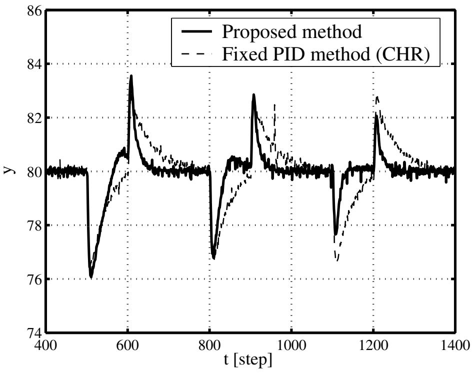

Fig. 7. Control result using the fixed PID controller(dotted line) and the proposed control method(solid line) in the case where the disturbance is periodically added .

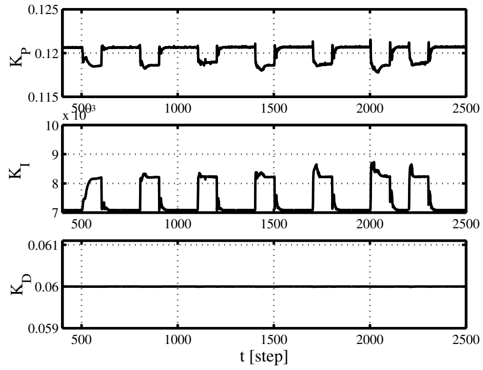

Fig. 8. Trajectories of PID parameters corresponding to Fig.7.

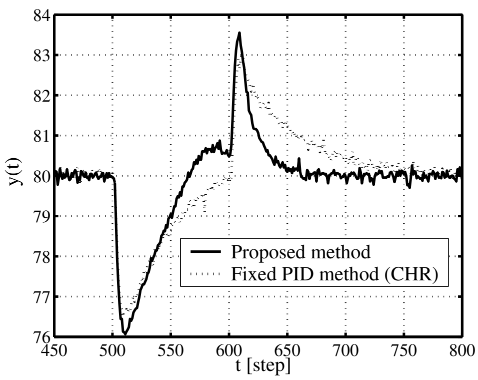

Fig. 9. Enlarged figure around t=500 in Fig.7.

as Table 1. 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 Fig.9 and Fig.10, where these figures show the control result around $t = 500$ and $t = 1100$, respectively. For the disturbance, another set of PID parameters is immediately extracted from the data-base corresponding to the current state It is clear that the proposed method effectively uses the knowledge that accumulated in the past. This is one of typical feature of the proposed method.

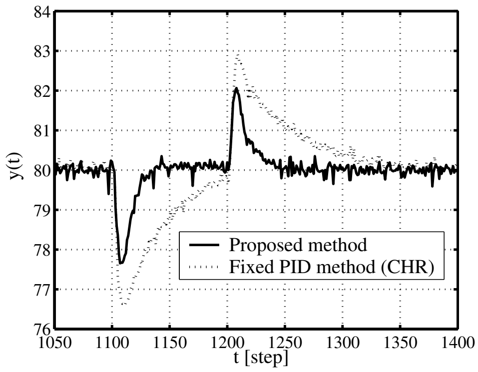

Fig. 10. Enlarged figure around t=1100 in Fig.7.

4. CONCLUSIONS

In this paper, a new design scheme of PID controllers using the DB 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. 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 experimentally verified on the pilot-scale temperature control system.

REFERENCES

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

Atkeson, C.G., A.W.Moore and S.Schaal (1997). Locally Weighted Learning for Control, Artificial Intelligence Review, **11**, pp.75-114.

Chien, K.L., J.A.Hrones and J.B.Reswick (1972). On the Automatic Control of Generalized Passive Systems, Trans. ASME, **74**, pp.175-185.

Stenman, A (1990). *Model on Demand: Algorithms, Analysis and Applications*, PhD thesis, Department of Electrical Engineering Linköping University.

Stenman, A., F.Gustafsson and L.Ljung (1996). Just in Time Models for Dynamical Systems, Proc. of 35th IEEE Conference on Decision and Control, pp.1115-1120.

Takao, K., T.Yamamoto and T.Hinamoto (2004). A Design of Model Driven Cascade PID Controllers Using a Neural Network, IEICE Trans. on Fundamentals., **E87-A**, pp.2322-2330.

Zhang, J., Y.Yim and J.Yang (1997). Intelligent Selection of Instances for Prediction Functions in Lazy Learning Algorithms, Artificial Intelligence Review, **11**, pp.175-191.