DESIGN OF A MEMORY-BASED PREFILTER SUPPLEMENTING A ROBUST PID CONTROL SYSTEM

Kenji Takao, Toru Yamamoto, and Takao Hinamoto

ABSTRACT

For systems with uncertainties, lots of PID parameter tuning methods have been proposed from the view point of the robust stability theory. However, the control performance becomes conservative using robust PID controllers. In this paper, a new two-degree-of-freedom (2DOF) controller, which can improve the tracking properties, is proposed for nonlinear systems. According to the proposed method, the prefilter is designed as the PD compensator whose control parameters are tuned by the idea of a memory-based modeling (MBM) method. Since the MBM method is a type of local modeling methods for nonlinear systems, PD parameters can be tuned adequately in an online manner corresponding to nonlinear properties. Finally, the effectiveness of the newly proposed control scheme is numerically evaluated on a simulation example.

Key Words: Two-degree-of-freedom control, PID control, robust control, nonlinear systems, process control, memory-based modeling.

I. INTRODUCTION

PID controllers [1] consist of proportional (P), integral (I) and derivative (D) control actions. In the time domain, P, I, and D actions correspond to the present, past, and future control errors, respectively. On the other hand, in the frequency domain, these actions roughly play the role of gain, phase-lag and phase-lead compensations, respectively. Since the control structure is quite simple and the physical meaning of control parameters is clear, PID controllers have been widely employed for about 80% or more of the control loops in industrial processes. Therefore, many PID parameter tuning methods have been previously proposed [2].

On the other hand, the generalized predictive controller (GPC) [3] is widely employed for chemical processes as one of model predictive controllers. Although the GPC is a control technique based on a multi-step prediction and is effective for systems with large time-delays, it is necessary to mount an extra control device for implementing the GPC.

Therefore, a tuning scheme of PID parameters has already been proposed based on the relation of the GPC [4]. According to the PID parameter tuning method based on the GPC (GPC-PID method), a PID controller can be designed whose performance is approximately equivalent to the GPC without any extra economic burden. Moreover, most process systems include system uncertainties, such as modeling errors and/or system perturbations. In order to solve these problems, a robust GPC-PID controller [5] has also been proposed. The PID controller is designed by adequately choosing a user-specified parameter included in the GPC design from the viewpoint of robust stability. However, the control performance may become quite conservative if the systems uncertainties are large. Therefore, in order to improve the set-point tracking properties,
two-degree-of-freedom (2DOF) control schemes have been proposed. However, most 2DOF controllers have been designed for linear systems. If the system has nonlinear properties, the control result may become worse.

Development of computers enables us to memorize, fast retrieve and read out a large amount of data. Utilizing these advantages, the following method has been proposed: whenever new data is obtained, the data is stored. Next, neighbors similar to the information requests, called “queries”, are selected from the stored data. Furthermore, the local model is constructed using these neighbors. This memory-based modeling (MBM) method, is called the Just-In-Time (JIT) method [6, 7], Lazy Learning [8, 9] method or Model-on-Demand (MoD) [10, 11], and these schemes have had lots of attention in last decade.

In this paper, authors have proposed the PID parameters tuning method based on the MBM method [12, 13]. According to the method, although the good control performances can be obtained, the robust stability for system uncertainties cannot be satisfied completely.

A new 2DOF controller is proposed for nonlinear systems. The robust GPC-PID controller is firstly designed so that uncertainties which are estimated beforehand are covered. Next, the prefilter is designed as the PD compensator, and PD parameters are tuned by the idea of MBM methods which can deal with nonlinear systems. Therefore, PD parameters can be adequately tuned corresponding to nonlinear properties of the controlled object. That is, the set-point tracking properties can be improved and the nonlinear properties can be compensated.

Finally, the effectiveness of the newly proposed scheme is numerically evaluated on a simulation example.

II. ROBUST PID CONTROLLER DESIGN

2.1 System description

In this paper, the following discrete-time nonlinear system is considered:

\[ y(t) = f(\phi(t)) \]  
\[ \phi(t) := [y(t - 1), \ldots, y(t - n_y), u(t - d - 1), \ldots, u(t - d - n_u)] \]  

where \( y(t) \) and \( u(t) \) denote the system output and the control input, \( n_y \) and \( n_u \) denote the orders of the system output and the control input, \( f(\cdot) \) denotes the nonlinear function, \( d \) denotes the time-delay, and \( \phi(t) \) is the historical data which consists of the system outputs and control inputs.

First, the system (1) is linearized by the following linear models at some equilibrium points.

\[ A_i(z^{-1})y(t) = z^{-(d+1)}B_i(z^{-1})u(t) \]  
\[ A_i(z^{-1}) = 1 + a_{i1}z^{-1} \]  
\[ B_i(z^{-1}) = b_{i0} + b_{i1}z^{-1} + \cdots + b_{im}z^{-m} \]  

where \( i \) denotes the reference number of the selected equilibrium points, and \( z^{-1} \) denotes the backward shift operator means \( z^{-1}y(t) = y(t - 1) \). Here, the least squares method is applied to estimate system parameters.

Moreover, some discrete-time linear models are transformed into the following continuous-time models [14]:

\[ G_i(s) = \frac{K_i}{1 + \tau_i s} e^{-L_i s}, \]  
\[ \tau_i = - \frac{T_s}{\log(-a_{i1})} \]  
\[ K_i = \sum_{j=1}^{m} b_{ij} \]  
\[ L_i = \left( \sum_{j=1}^{m} j \cdot b_{ij} + d \right) T_s. \]

From (6), a nominal model \( G_n(s) \) and a maximum perturbed model \( G_m(s) \) are designed using gain \( K_i \), time-constant \( \tau_i \) and time-delay \( L_i \) as follows:

\[ G_n(s) = \frac{K_n}{1 + T_h s} e^{-L_n s}, \]  
\[ G_m(s) = \frac{K_{max}}{1 + T_{min} s} e^{-L_{max} s} \]  
\[ K_{min} \leq K_i \leq K_{max}, \quad \tau_{min} \leq \tau_i \leq \tau_{max}, \]  
\[ L_{min} \leq L_i \leq L_{max} \]  
\[ K_n = \frac{K_{max} + K_{min}}{2}, \quad \tau_n = \frac{\tau_{max} + \tau_{min}}{2}, \]  
\[ L_n = \frac{L_{max} + L_{min}}{2}. \]

The maximum multiplicative uncertainty \( h_m(s) \) is computed by \( G_n(s) \) and \( G_m(s) \).

\[ h_m(s) = \frac{G_m(s) - G_n(s)}{G_n(s)}. \]
2.2 GPC-PID controller design

The PID tuning algorithm (GPC-PID) is derived based on the relationship between the PID control and the GPC [4, 5]. First, the continuous-time model given by (6) is approximated by the first-order system with the time-delay,

$$\hat{G}(s) = \frac{K}{1 + \tau_e s} \approx \frac{K}{1 + \tau_e + L s}. \quad (14)$$

Moreover, the discrete-time model corresponding to (14) is given as

$$\hat{A}(z^{-1})y(t) = z^{-1}\hat{B}(z^{-1})u(t) \quad (15)$$

where

$$\hat{A}(z^{-1}) = 1 + \hat{a}_1 z^{-1} + \hat{a}_2 z^{-2} \quad (16)$$

$$\hat{B}(z^{-1}) = \hat{b}_0 + \hat{b}_1 z^{-1}. \quad (17)$$

Here, the velocity-type PID control law [15] is described as

$$\Delta u(t) = \frac{k_c T_s}{T_l} e(t) - k_c \left( \Delta + \frac{T_D}{T_s} \Delta^2 \right) y(t) \quad (18)$$

where

$$e(t) := r(t) - y(t). \quad (19)$$

$r(t)$ denotes the reference signal and is given by piecewise constant components, $\Delta$ denotes the differencing operator defined as $1 - z^{-1}$. Also, $k_c$, $T_l$ and $T_D$, respectively, denote the proportional gain, the reset time and the derivative time, and $T_s$ denotes the sampling interval.

On the other hand, the cost function of the GPC [3] is given by

$$J = E \left[ \sum_{j=N_1}^{N_2} (y(t + j) - r(t))^2 + \lambda \sum_{j=1}^{N_u} (\Delta u(t + j - 1))^2 \right] \quad (20)$$

where $\lambda$ denotes the user-specified parameter, which means the weighting factor for the control input, and the period from $N_1$ thru $N_2$ denotes the prediction horizon, and $N_u$ denotes the control horizon. For simplicity, they are set as $N_1 = 1$, $N_2 = N$ and $N_u = N$, respectively, where $N$ is designed in consideration of the time constant and time-delay of the controlled object. The control law of the GPC can be obtained based on minimizing the cost function (20).

By comparing the control law of the GPC with the PID law (18), the PID parameters can be obtained by

$$k_c = f_P(\hat{\theta}, \lambda), \quad T_l = f_I(\hat{\theta}, \lambda)$$

$$T_D = f_D(\hat{\theta}, \lambda) \quad (21)$$

$$\hat{\theta} := [\hat{a}_1, \hat{a}_2, \hat{b}_0, \hat{b}_1] \quad (22)$$

where $f_P$, $f_I$ and $f_D$ denote the functions [4]. That is, the PID parameters are given as the function of the system parameter $\hat{\theta}$ and the user-specified parameter $\lambda$ included in the GPC. The detailed explanation of the above tuning algorithm should be referred in the reference [4].

2.3 Robust design

According to the GPC-PID controller, the performance of the PID controller is tuned by the user-specified parameter $\lambda$. Therefore, $\lambda$ is designed so that the close-loop system may be stabilized, that is, the PID controller which satisfy the robust stability can be designed. Using the multiplicative uncertainties $h(j\omega)$, the real controlled object $G(z^{-1})$ is expressed by

$$\tilde{G}(j\omega) = (1 + h(j\omega))G_n(j\omega). \quad (23)$$

Here $|h(j\omega)|$ satisfies the following relationship:

$$|h(j\omega)| \leq h_m(\omega) \quad (24)$$

where $h_m(\omega)$ is given by (13). Furthermore, let $T(j\omega)$ be the complementary sensitivity function constructed for the nominal model $G_n(z^{-1})$, then the necessary and sufficient condition which the close-loop system satisfies the robust stability [16] can be expressed as follows:

$$|T(j\omega)|h_m(\omega) < 1 \quad (25)$$

$$T(j\omega) = \frac{k_c}{j\omega T_l} \frac{G_n(j\omega)}{1 + G_n(j\omega)C(j\omega)} \quad (26)$$

where

$$C(j\omega) = k_c \left[ 1 + \frac{1}{j\omega T_l} + j\omega T_D \right]. \quad (27)$$

Here, PID parameters $(k_c, T_l, T_D)$ are calculated by (21). That is, since the PID parameters are given as the function of the user-specified parameter $\lambda$, the gain property of $T(j\omega)$ can be effectively and easily changed using the parameter $\lambda$. Thus, the GPC-PID controller can be designed in consideration of the robust stability, where $\lambda$ is designed so that the close-loop system may satisfy (25).
III. PD COMPENSATOR DESIGN

It is pointed out that the desirable tracking property in the transient state cannot be obtained using the only robust PID controller. Therefore, the following PD compensator is additionally inserted in the robust PID control system:

\[ w_r(t) = \{K_P^2(t) + K_D^2(t)\} r(t) \]  (28)

where \( w_r(t) \) denotes the output of the PD compensator. \( K_P^2 \) and \( K_D^2 \) are the proportional gain and derivative gain included in the PD compensator. In this paper, in order to compensate nonlinear properties and improve the tracking properties, the PD parameters are tuned by the idea of the MBM method. According to the MBM method, the nonlinear systems can be linearized by a local linear model with respect to each query. Therefore, nonlinear properties can be compensated by tuning the PD parameters based on the idea of the MBM method.

First of all, by using (15) and (18), the transfer function \( H_{w_r \rightarrow y} \) from \( w_r \) to \( y \) is given as follows:

\[ H_{w_r \rightarrow y} = \frac{z^{-1} \hat{B}(z^{-1})k_cT_s/T_l}{\Delta \hat{A}(z^{-1}) + z^{-1} \hat{B}(z^{-1})k_c(\Delta + T_s/T_l + \Delta^2 T_D/T_s)} . \]

Here, using (29), the following relationship can be obtained:

\[ y(t + 1) = \tilde{f}(\tilde{\phi}(t + 1)) \]  (30)

\[ \tilde{\phi}(t + 1) := \{y(t), \ldots, y(t - 3), w_r(t), w_r(t - 1)\} \]  (31)

where \( \tilde{f}(. \cdot) \) denotes a linear function. Moreover, by substituting (28) into (31), the following equation can be derived:

\[ y(t + 1) = \tilde{f}(\tilde{\phi}(t + 1)) \]  (32)

\[ \tilde{\phi}(t + 1) := \{y(t), \ldots, y(t - 3), K_2(t), r(t), r(t - 1), w_r(t - 1)\} \]  (33)

\[ K_2(t) := \{K_P^2(t), K_D^2(t)\} \]  (34)

where \( \tilde{f}(. \cdot) \) denotes a linear function. Here, by transforming (32) and (33), PD gains \( K_2(t) \) are given by

\[ K_2(t) = \tilde{f}(\tilde{\phi}(t)) \]  (35)

\[ \tilde{\phi}(t) := \{y(t + 1), y(t), \ldots, y(t - 3), r(t), r(t - 1), w_r(t - 1)\} \]  (36)

where \( \tilde{f}(. \cdot) \) denotes a nonlinear function. Since the future output \( y(t + 1) \) included in (36) cannot be obtained at \( t \), \( y(t + 1) \) is replaced by \( r(t + 1) \). The purpose of the control considered in this paper is to realize \( y(t + 1) \rightarrow r(t + 1) \). Therefore, \( \tilde{\phi}(t) \) included in (36) is newly rewritten as follows:

\[ \tilde{\phi}(t) := \{r(t + 1), y(t), \ldots, y(t - 3), r(t), r(t - 1), w_r(t - 1)\} . \]  (37)

In fact, the nonlinear function \( \tilde{f}(. \cdot) \) is recursively approximated as local linear functions using the MBM method. Moreover, the data is stored in the form of the data vector \( \Phi \) expressed by the following equation:

\[ \Phi(j) = [\tilde{\phi}(j), K_2(j)] \]  (38)

The design procedure of the PD compensator using the MBM method is summarized as follows:

**Step 1. Calculate distance and select neighbors**

Distances between the information request (called “query”) \( \tilde{\phi}(t) \) and the information vector \( \tilde{\phi}(i) \) (\( i \neq t \)) stored in the database are calculated using the following \( L_1 \)-norm with some weights:

\[ \text{dis}(\tilde{\phi}(t), \tilde{\phi}(j)) = \sum_{l=1}^{8} \left| \frac{\tilde{\phi}_l(t) - \tilde{\phi}_l(j)}{\max_m \tilde{\phi}_l(m) - \min_m \tilde{\phi}_l(m)} \right| \]  (39)

where \( N_D(t) \) denotes the number of information vectors stored in the database when the query \( \tilde{\phi}(t) \) is given, and \( k \) pieces with the smallest distances are chosen from all information vectors. The data vectors \( \Phi_N(i) \) (\( i = 1, 2, \ldots, k \)) corresponding to selected information vectors are called the neighbor data.

\[ \Phi_N(i) = [\tilde{\phi}_N(i), K_2N(i)] \]  (40)

**Step 2. Construct local model**

Next, using selected neighbor data \( \Phi_N \), the local model is constructed based on the linear

© 2008 John Wiley and Sons Asia Pte Ltd and Chinese Automatic Control Society
weighted average (LWA) [17].

\[
K_2^{old}(t) = \sum_{i=1}^{k} w_i K_{2N}(i), \quad \sum_{i=1}^{k} w_i = 1 \quad (41)
\]

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

\[
w_i = \frac{8}{\sum_{i=1}^{8} \left( 1 - \frac{[\hat{\phi}_i(t) - \hat{\phi}_{1N}(i)]^2}{[\max \phi_{1N}(m) - \min \phi_{1N}(m)]^2} \right)}.
\]

(42)

**Step 3. Data adjustment**

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

\[
K_2^{new}(t) = K_2^{old}(t) - \eta \frac{\partial J(t + d + 1)}{\partial K_2^{old}(t)} \quad (43)
\]

\[
\eta := [\eta_P, \eta_D] \quad (44)
\]

where \(\eta\) denotes the learning rate, and the following \(J(t + d + 1)\) denotes the error criterion:

\[
J(t + d + 1) := \frac{1}{2} \varepsilon(t + d + 1)^2 \quad (45)
\]

\[
\varepsilon(t) := y_r(t) - y(t) \quad (46)
\]

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

\[
y_r(t) = \frac{z^{-(d+1)} T(1)}{T(z^{-1})} r(t) \quad (47)
\]

\[
T(z^{-1}) := 1 + t_1 z^{-1} + t_2 z^{-2} \quad (48)
\]

\(T(z^{-1})\) is designed based on the reference literature [18]. That is, \(T(z^{-1})\) is designed in consideration of the desired rise-time and the damping property.

Here, for the purpose of generalizing robust tracking for the reference signal, it is compulsory to replace \(K_{p2}(t)\) with 1.0 when the control error intermittently falls into the \(\beta[\%]\) neighborhood corresponding to the reference signal.

**Step 4. Remove redundant data**

The storing of redundant data in the database needs excessive computational time. Therefore, an algorithm to avoid the excessive increase of stored data is required. The procedure is carried out in the following two steps.

First, the data vectors \(\Phi_F(\tilde{i})\), which satisfy the following first condition, are extracted from the database without the neighbor data \(\Phi_N(i)\) \((i = 1, 2, \ldots, k)\).

**First condition**

\[
dis(\tilde{\phi}(t), \hat{\phi}(i)) \leq \alpha_1, \quad 1 \leq i \leq N_{D}(t) \quad (49)
\]

\[
\hat{\phi}(i) \notin \bar{\Phi}_N(j), \quad (i \neq j) \quad (50)
\]

where \(\Phi_F(\tilde{i})\) is defined by

\[
\Phi_F(\tilde{i}) := [\Phi_F(\tilde{i}), K_{2F}(\tilde{i})], \quad 1 \leq \tilde{i} \leq N_{D}(t). \quad (51)
\]

Moreover, the data vectors \(\Phi_S(\tilde{i})\) which satisfy the following second condition, are further chosen from the extracted \(\Phi_F(\tilde{i})\).

**Second condition**

\[
\sum_{i=1}^{3} \left( \frac{K_{F1}(\tilde{i}) - K_{new}(t)}{K_{F1}(\tilde{i})} \right)^2 \leq \alpha_2 \quad (52)
\]

where \(\Phi_S(\tilde{i})\) is defined by

\[
\Phi_S(\tilde{i}) := [\Phi_S(\tilde{i}), K_{2S}(\tilde{i})], \quad 1 \leq \tilde{i} \leq N. \quad (53)
\]

If there exist plural \(\Phi_S(\tilde{i})\), the information vector with the smallest value in the second condition among all \(\Phi_S(\tilde{i})\) is removed. By the above procedure, the redundant data can be removed from the database.

The block diagram of the proposed control system mentioned above is shown in Fig. 1. Moreover, the stability of the proposed method is considered. First of all, the transfer function \(H_{r \rightarrow y}\) from \(r\) to \(y\) is given by

\[
H_{r \rightarrow y}(s) = \frac{K_{P2}(s) + K_{D2}(s)s}{T(s)}. \quad (54)
\]

The closed-loop controller (PID controller) is designed so that the closed-loop system may satisfy condition...
(25) of the robust stability. Therefore, it is safe to say that the proposed system is stable as long as the system uncertainties are included in the maximum estimated uncertainty \( h_m(s) \). That is, the proposed controller can be applied only to particular systems, which can be guaranteed robust stability by the closed-loop controller (i.e. the robust PID controller).

**IV. SIMULATION EXAMPLE**

In order to investigate the behavior of the proposed control method, a simulation example is discussed. As a controlled object, the following Hammerstein model [19] is considered:

\[
y(t) = 0.6y(t-1) - 0.1y(t-2) + 1.2x(t-2) - 0.1x(t-3) + \zeta
\]

\[
x(t) = 1.5u(t) - 1.5u^2(t) + 0.5u^3(t)
\]

where \( \zeta \) denotes the white Gaussian noise with zero mean and variance 0.01².

First, the robust PID controller is designed. The Hammerstein model is linearized at some equilibrium points. Among these linearized models, “Max model” (T:lowest, K, L: highest) and “Min model” (T: highest, K, L: lowest) are chosen. Here, static properties of the Hammerstein model, “Min model” and “Max model” are shown in Fig. 2.

Using Min and Max models, the nominal model and the maximum perturbed model are designed, and the robust PID parameters can be calculated. The calculated PID parameters are given by

\[
k_c = 0.039, \quad T_I = 0.7179, \quad T_D = 0.1229. \quad (56)
\]

The reference model \( T(z^{-1}) \) is designed as

\[
T(z^{-1}) = 1 - 0.7767z^{-1} + 0.1508z^{-2}.
\]

Furthermore, the user-specified parameters included in the proposed method are determined as shown in Table 1. The reference signal \( r(t) \) is given by

\[
r(t) = \begin{cases} 
0.5 & (0 \leq t < 100) \\
1.0 & (100 \leq t < 200) \\
2.0 & (200 \leq t < 300) \\
0.8 & (300 \leq t < 400) \\
0.0 & (400 \leq t \leq 500).
\end{cases} \quad (58)
\]

First, for the purpose of comparison, the control result using the only robust PID controller, whose PID parameters are given by (56), is shown in Fig. 3 (broken line). In Fig. 3 (broken line), it can be seen that the set-point tracking property is worse around \( y = 1.2 \) because the robust PID control system is quite conservative due to the strong nonlinear properties.

Next, the self-tuning GPC-PID controller [4] was applied to the system. Here, the PID parameters utilized in this method are tuned by (21), whose system parameters are identified by the recursive least square method. The control result in Fig. 3 (solid line) shows that the control result is oscillatory around \( y = 2.0 \) and the control system falls in unstable state after 400 steps owing to the nonlinear properties of the system.
Next, the 2DOF-PID controller, whose PD compensator is fixed (called fixed 2DOF-PID), was applied to the system. The PD parameters were designed by trial and error (note that $K_{P2}$ must be set as 1.0). The control results using the fixed 2DOF-PID are shown in Fig. 4 ($K_{D2} = 8.0$) and Fig. 5 ($K_{D2} = 10.0$). In Fig. 4, it can be seen that although the tracking properties can be improved in the case where the controlled object has weak nonlinear properties, the tracking property cannot be improved enough in the case where the controlled object has strong nonlinear properties. Fig. 5 shows that if the $K_{D2}$ is designed with a bigger value than Fig. 4, the tracking properties cannot be improved around $t = 300$, and the overshoot arises around $t = 400$. Therefore, even if the PD parameters are adequately tuned in off-line, the nonlinear properties cannot be compensated without tuning the PD parameter in online.

Next, the control result using the proposed method is shown in Fig. 6 (solid line), the trajectories of PD parameters are illustrated in Fig. 7, and the output of the PD compensator is shown in Fig. 8. From these figures, it is clear that the set-point tracking property is superior in comparison with the only robust PID controller, because parameters included in the PD compensator are adequately adjusted by the MBM method, and the output of the PD compensator is significantly changed corresponding to the set-points in order to cope with the nonlinear properties. Moreover, the error behavior by the proposed method is shown in Fig. 9. Here, the number of 1 epoc is 500 steps, and ISE denotes

$$ISE(i + 1) := \sum_{t=1}^{500} (y_r(t) - y(t))^2,$$

$i$ : the number of iteration

$$i :$$ the number of iteration

$$(59)$$
where ISE(1) means ISE by only the robust PID controller. From Fig. 9, the proposed method gives us good learning efficiency. For confirmation, the result at 4 epoc is shown in Fig. 10. Moreover, the behavior is shown more clearly in Fig. 10, where this figure shows the control results around $t = 300$. The trajectories of PD parameters are illustrated in Fig. 12. From these figures, it is clear that the response was improved in comparison with 1 epoc.

Moreover, * in Fig. 13 and Fig. 14 shows the point where an information vector is newly stored in the database. Fig. 13 shows that lots of data is newly stored in the database because the number of the stored data is few at 1 epoc. Then, the number of stored data was 141 at 1 epoc. Using the algorithm (Step 4) to remove needless data, the number of stored data can be reduced from 500 to 141. Moreover, the number of stored data can be also effectively reduced from 2500 to 241 at 4 epoc in Fig. 14. It is clear that the number of data newly stored in the database decreases as the learning advances, and the redundant data is adequately neglected. This means that Step 4 in the proposed algorithm works effectively.

Next, the 2DOF-PID controller, whose feedforward compensator is designed using the neural network (2DOF-NN-PID), was applied to this system. The block diagram of the 2DOF-NN-PID is shown in Fig. 15. The 2DOF-NN-PID is composed of the PID controller and the PD compensator. The robust GPC-PID controller is utilized as the closed-loop controller. The PD compensator is designed by the multi-layered neural-network (NN). The weights included in the NN are learned by the back propagation method based on minimizing the error.
K. Takao et al.: Design of a Memory-Based Prefilter Supplementing a Robust PID Control System

Fig. 11. Enlarged figure around \( t = 300 \) in Fig. 10.

Fig. 12. Trajectories of PD parameters corresponding to Fig. 10 (4 epoc).

Fig. 13. 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 (iteration = 4 epoc).

Fig. 14. 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 (iteration = 4 epoc).

Fig. 15. Block diagram of the 2DOF-NN-PID.

Fig. 16. Error behaviors by the 2DOF-NN-PID and the proposed method.

Fig. 17. Control results using the 2DOF-NN-PID (1 epoc).

Fig. 18. Control results using the 2DOF-NN-PID (500 epoc).

Firstly, the step disturbance is added to the system:

\[
d_{\text{step}}(t) = \begin{cases} 
0 & t < 200 \\
1.0 & t \geq 200.
\end{cases}
\]
The control results are shown in Fig. 19, and the trajectories of PD parameters are shown in Fig. 20. From Figs. 19 and 20, it is clear that the disturbance response is improved using the proposed method, because PD parameters are adequately adjusted corresponding to the step disturbance.

Next, the following sinusoidal disturbance is added to the system:

\[
  d_{\text{step}}(t) = \begin{cases} 
  0 & t < 200 \\
  0.5 \sin(\pi \cdot t/50) & t \geq 200. 
  \end{cases} \tag{62}
\]

The control results are shown in Fig. 21, and the trajectories of PD parameters are shown in Fig. 22. The
V. CONCLUSIONS

According to the conventional robust controllers, the control performance may become quite conservative in the case where the system uncertainties are large. Though the 2DOF controller has been proposed in order to solve this problem, most of the 2DOF controllers are designed for the linear systems, which makes it impossible to effectively improve the tracking property for nonlinear systems.

Therefore, a new design scheme of a 2DOF controller, which can compensate the nonlinear properties, has been proposed in this paper. Here, the PD compensator has been inserted by the form of the prefilter. Moreover, the PD parameters included in the PD compensator have been tuned by the idea of the MBM method. According to the MBM method, the nonlinear function can be easily and recursively approximated by the local linear models. Therefore, the proposed prefilter can compensate the nonlinear properties and improve the set-point tracking properties, even if the system has nonlinear properties.

In addition, the effectiveness of the proposed method has been verified by a numerical simulation example.
REFERENCES


Kenji Takao received his M.Eng. degree from Hiroshima University in 2004. He is currently pursuing the Ph.D. degree in system engineering at Hiroshima University. He is also a research fellow of the Japan Society for the Promotion of Science. His current research interests are in the area of PID control, intelligent control, process control, and memory-based modeling/control.

Mr. Takao is a student member of the Society of Instrumentation and Control Engineers in Japan (SICE) and the Institute of Systems, Control and Information Engineers in Japan (ISCIE).

Toru Yamamoto received his B.Eng. and M.Eng. degrees from University of Tokushima in 1984 and 1987, respectively, and his D.Eng. degree from Osaka University in 1994. He has been a professor in the Department of Technology and Information Education at Hiroshima University. He was a visiting researcher in the Department of Mathematical Engineering and Information Physics at University of Tokyo in 1991, and he was a visiting professor in the Department of Chemical and Materials Engineering at University of Alberta,
Edmonton, Canada in 1996 and 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 received Prize of Progress from the Institute of Electrical Engineers in Japan (IEEJ) and Technology Award from the Society of Instrumentation and Control Engineers in Japan (SICE) in 2003.

Takao Hinamoto received the B.E. degree in electrical engineering from Okayama University, Okayama, Japan, in 1969, the M.E. degree in electrical engineering from Kobe University, Kobe, Japan, in 1971, and the Dr. Eng. degree in electrical engineering from Osaka University, Osaka, Japan, in 1977. From 1972 to 1988, he was with the Faculty of Engineering Kobe University, Kobe, Japan. From September 1979 to March 1981, he was on leave from Kobe University as a visiting member of staff in the Department of Electrical Engineering, Queen’s University, Kingston, Ontario, Canada. During 1988–1991, he was Professor of Electrical Circuits in the Faculty of Engineering, Tottori University, Tottori, Japan. Since January 1992, he has been Professor of Electronic Control in the Department of Electrical Engineering, Hiroshima University, Higashi-Hiroshima, Japan. His teaching and research interests include digital signal processing, system theory, and control systems engineering. Dr. Hinamoto was an associate editor of the IEEE Transactions on Circuits and Systems II during 1993–1995, and an associate editor of the IEEE Transactions on Circuits and Systems I during 2002–2003. He served as the General Chair of the 47th IEEE International Midwest Symposium on Circuits and Systems to be held in Hiroshima, Japan, in July 2004. He is a recipient of the IEEE Third Millennium Medal and is a Fellow member of IEEE.