2006 International Joint Conference on Neural Networks
Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada
July 16-21, 2006

A Design of PID Controllers with a Switching Structure by a Support Vector Machine

Kenji Takao, Toru Yamamoto *Member, IEEE*, and Takao Hinamoto, *Member, IEEE*

***Abstract*— PID control schemes have been widely employed for most process systems represented by chemical processes. However, it is a very important problem how to tune PID parameters, because these parameters have a great influence on the stability and the performance of the control system. On the other hand, lots of works for the robust control have been carried out to cope with system uncertainties. Then, some PID parameter tuning methods have been proposed based on the robust stability. However, if the range of uncertainties is very wide, the control performance becomes quite conservative. By the way, the support vector machine (SVM) has been proposed as one of the pattern recognition methods and gets lots of attention in last decades. The main motivation in this paper is to present a design scheme of controllers with the switching structure, in which some robust PID controllers are suitably switched using the SVM. Finally, the proposed control scheme is numerically evaluated on a simulation example.**

I. Introduction

Nowadays, a large variety of methods for designing controllers have been studied corresponding to development of computer technologies. However, in industrial processes, PID controllers[1]-[5], which belong to a type of classical controllers, have been widely employed about 80% or more of control loops. In designing PID controllers, it is very important to tune the PID parameters. If the tuning is not good, not only the control performances become worse but also the control system becomes unstable. Therefore, many tuning methods of PID parameters have been studied actively [5]. Among them, the PID parameter tuning scheme based on the relation between a generalized predictive controller (GPC) and a PID control law (GPC-PID) [6] is one of the effective PID parameter tuning methods [7]. Furthermore, according to GPC-PID, the robust PID controller, which can satisfy the robust stability, can be designed by tuning a user-specified parameter included in the design of a GPC [8]. However, according to robust controllers, if the range of system uncertainty is widely estimated, the control performance becomes very conservative and the tracking property becomes worse.

In this paper, the wide range of uncertainty is firstly divided into some small ranges of uncertainty using a *priori* information, and the robust GPC-PID controller is designed for each small range. Next, the designed robust controllers are adaptively switched using *a posteriori* information. And so, the control performance can be improved, because the robust controller, which is designed for small range of uncertainty, is applied. Here, there is a serious problem that the controller must be adequately switched according to the property of the controlled object. In other words, it is necessary to know the properties of the controlled object which changes gradually, and determine the suitable controller corresponding to system properties. Therefore, the support vector machine (SVM) [9], [10], [11], which is a very effective tool for the pattern recognition, is applied. The SVM is a classification method that is more highly accurate than the conventional classification techniques, because the SVM can immediately find the hyperplane to separate two classes based on the maximizing margin. Moreover, the SVM can deal with the case that the separating hyperplane is nonlinear function using the Kernel trick. Then, using the SVM, it should be able to apply to various process systems, for example nonlinear systems. In the proposed method, the SVM is firstly trained by a *priori* information. And then, using the trained SVM, the current property of the controlled object can be discriminated and the adequate controller can be selected.

Here, the input vector of the SVM becomes higher dimension, because much information of the system must be included in the vector, in order to discriminate the property of the system. However, if the order of the input vector increases, the memory and the calculation cost of the computer are exponentially increased. On the other hand, the lack of information is deteriorated of accuracy of the classification, if the information included in the input vector is casually reduced. In order to overcome this problem, lots of information is firstly compressed into a few typical components by adopting the principal component analysis (PCA).

This outline of this paper is as follows. The problem to be considered in this paper is firstly addressed. This is followed by a brief explanation of the robust GPC-PID controller design. Next, the scheme of the switching structure using the SVM is proposed. Finally, the effectiveness of the proposed method is numerically evaluated on a simulation example.

II. Robust GPC-PID

A. System description

The following first-order continuous-time model with the time-delay is considered.

$$\tilde{G}(s) = \frac{K}{1+Ts}e^{-Ls}, \tag{1}$$

K.Takao and T.Hinamoto are with the Graduate School of Engineering, Hiroshima University, Hiroshima, 739-8527, Japan (phone: +81-82-424-7673; fax: +81-82-422-7195; email: takao17, hinamoto@hiroshima-u.ac.jp).
Toru Yamamoto is with the Graduate School of Education, Hiroshima University, Hiroshima, 739-8524, Japan (phone: +81-82-424-7160; fax: +81-82-424-3478; email: yama@hiroshima-u.ac.jp).

where K, T and L are the system gain, the time-constant and the time-delay, respectively. Next, the continuous-time model (1) is converted into the discrete-time model (2) with sampling time T_s.

$$\tilde{A}(z^{-1})y(t) = z^{-(k+1)}\tilde{B}(z^{-1})u(t) + \frac{\xi(t)}{\Delta}, \qquad (2)$$

where

$$\tilde{A}(z^{-1}) = 1 + \tilde{a}_1 z^{-1}, \quad \tilde{B}(z^{-1}) = \tilde{b}_0 + \tilde{b}_1 z^{-1}. \qquad (3)$$

$u(t)$ and $y(t)$ are the input and output signals of the system, respectively. $\xi(t)$ is the white Gaussian noise with zero mean and variance σ^2. And k is the time-delay in the discrete-time model.

B. Generalized Predictive Control (GPC)

The following second-order model, which is obtained by approximating the time-delay L included in (1), is considered.

$$G(s) \cong \frac{K}{(1+Ts)(1+Ls)}. \qquad (4)$$

Next, (4) is converted into discrete-time model (5) with sampling time T_s.

$$A(z^{-1})y(t) = z^{-1}B(z^{-1})u(t) + \frac{\xi(t)}{\Delta}, \qquad (5)$$

where

$$A(z^{-1}) = 1 + a_1 z^{-1} + a_2 z^{-2}, \ B(z^{-1}) = b_0 + b_1 z^{-1}, \qquad (6)$$

Here, (5) is considered as the design-oriented model of the GPC.

Now, the GPC law is derived by minimizing the following cost function:

$$J = E\left[\sum_{j=N_1}^{N_2}[y(t+j) - w(t)]^2 + \lambda \sum_{j=1}^{NU}[\Delta u(t+j-1)]^2\}\right], \qquad (7)$$

where $w(t)$ is the reference signal, λ is the user-specified parameter which means the weighting factor for the control input. Furthermore, the period from N_1 thru N_2 denotes the prediction horizon, and NU denotes the control horizon. For simplicity, they are respectively set as $N_1 = 1$, $N_2 = N$ and $NU = N$, where N is designed in consideration of the time-constant of the controlled object. $E[\cdot]$ means the expectation of the sigmals. Then, the control law by minimizing the cost function (7) can be immediately obtained by

$$\sum_{j=1}^{N} p_j F_j(z^{-1})y(t) + \{1 + z^{-1}\sum_{j=1}^{N} p_j S_j(z^{-1})\}\Delta u(t) - \sum_{j=1}^{N} p_j w(t) = 0, \qquad (8)$$

where $F_j(z^{-1})$ and $S_j(z^{-1})$ are obtained by solving the following Diophantine equation:

$$1 = \Delta A(z^{-1})E_j(z^{-1}) + z^{-j}F_j(z^{-1}) \qquad (9)$$
$$E_j(z^{-1})B(z^{-1}) = R_j(z^{-1}) + z^{-j}S_j(z^{-1}), \qquad (10)$$

where

$$E_j(z^{-1}) = 1 + e_1 z^{-1} + \cdots + e_{j-1}z^{-(j-1)} \qquad (11)$$
$$F_j(z^{-1}) = f_0^j + f_1^j z^{-1} + f_2^j z^{-2} \qquad (12)$$
$$R_j(z^{-1}) = r_0 + r_1 z^{-1} + \cdots + r_{j-1}z^{-(j-1)} \qquad (13)$$
$$S_j(z^{-1}) = s_0^j \qquad (14)$$

$$[p_1, p_2, \cdots, p_N] := [1, 0, \cdots, 0] \cdot (\mathbf{R}^T\mathbf{R} + \mathbf{\Lambda})^{-1}\mathbf{R}^T, \qquad (15)$$

The matrix $\mathbf{R}$ which consists of the coefficients of $R_j(z^{-1})$ is defined as

$$\mathbf{R} := \begin{bmatrix} r_0 & & & & \\ r_1 & r_0 & & 0 & \\ \cdot & \cdot & \cdot & & \\ \cdot & \cdot & & \cdot & \\ \cdot & \cdot & & & \cdot \\ r_{N-1} & r_{N-2} & \cdot & \cdot & \cdot & r_0 \end{bmatrix}, \qquad (16)$$

and Λ is defined as

$$\mathbf{\Lambda} := \mathrm{diag}\{\lambda\}. \qquad (17)$$

C. Design of GPC-PID

The following I-PD control law is introduced.

$$\Delta u(t) = \frac{k_c \cdot T_s}{T_I}e(t) - k_c\{\Delta + \frac{T_D}{T_s}\Delta^2\}y(t) \qquad (18)$$
$$e(t) := w(t) - y(t), \qquad (19)$$

where k_c, T_I and T_D are the proportional gain, the reset-time and the derivative time, respectively. Here, in order to derive the PID parameter tuning law based on the relationship between the PID control law and the GPC law, the second term of (8) is firstly replaced by the static gain.

$$\sum_{j=1}^{N} p_j F_j(z^{-1})y(t) + \{1 + \sum_{j=1}^{N} p_j S_j(1)\}\Delta u(t) - \sum_{j=1}^{N} p_j w(t) = 0. \qquad (20)$$

Next, the comparison of (18) with (20), yields the following PID parameter tuning law [7]:

$$\left.\begin{aligned} k_c &= -(\tilde{f}_1 + 2\tilde{f}_2) \\ T_I &= -\frac{\tilde{f}_1 + 2\tilde{f}_2}{\tilde{f}_0 + \tilde{f}_1 + \tilde{f}_2}T_s \\ T_D &= -\frac{\tilde{f}_2}{\tilde{f}_1 + 2\tilde{f}_2}T_s \end{aligned}\right\} \qquad (21)$$

where

$$\frac{1}{\nu}\sum_{j=1}^{N} p_j F_j(z^{-1}) := \tilde{f}_0 + \tilde{f}_1 z^{-1}\tilde{f}_2 z^{-2} \qquad (22)$$

$$\nu := 1 + \sum_{j=1}^{N} p_j S_j(1). \qquad (23)$$

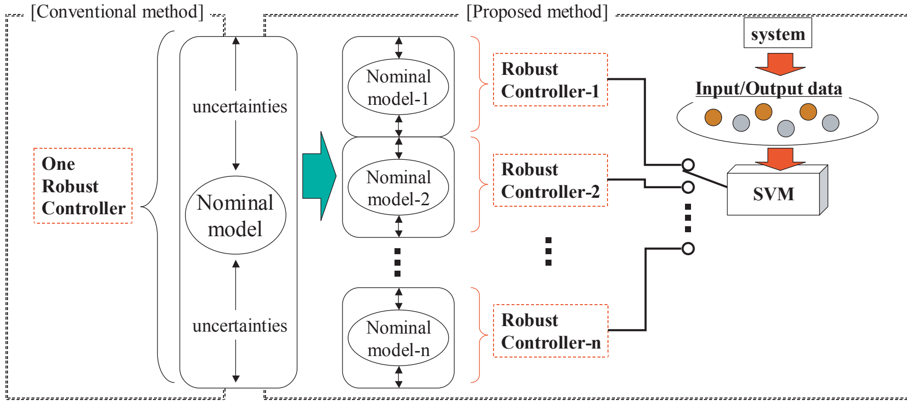

Fig. 1. Schematic figure of the proposed method.

By the way, the dynamics of the system often changes due to the changes of the operating conditions and/or various disturbances, that is, the control objects have generally uncertainty. the general system has uncertainty. In such a case, the closed-loop system, which is optimally designed based on the nominal model, may become unstable. Therefore, in the next section, the robust PID controller, which can satisfy the robust stability, is designed by tuning the user-specified parameter λ included in the GPC law.

D. Design of robust GPC-PID

Let $\tilde{G}(j\omega)$ and $G_o(j\omega)$ be the frequency transfer function of the nominal controlled object and that of the perturbed controlled object, respectively. Then $G_o(j\omega)$ is expressed as

$$G_o(j\omega) = \{1 + h(j\omega)\}\,\tilde{G}(j\omega) \tag{24}$$

$h(j\omega)$ denotes the multiplicative uncertainties, and it is assumed that $h_m(\omega)$ satisfies the following relationship:

$$|h(j\omega)| \leq h_o(\omega) \tag{25}$$

Here, $T(j\omega)$ is defined as the complementary sensitivity function constructed for the nominal model $\tilde{G}(j\omega)$. Then, the following necessary and sufficient condition is required in order to guarantee the robust stability of the closed-loop system:

$$|T(j\omega)|h_o(\omega) < 1 \tag{26}$$

Moreover, the following equation can be obtained using the relation $z = e^{j\omega T_s}$.

$$T(z^{-1}) = \frac{z^{-(k+1)}\tilde{B}(z^{-1})\tilde{F}(z^{-1})}{\Delta\tilde{A}(z^{-1}) + z^{-(k+1)}\tilde{B}(z^{-1})\tilde{F}(z^{-1})} \tag{27}$$

Further, it is found that the gain property of the complementary sensitivity function, $|T(j\omega)|$, can be effectively and easily changed by the user-specified parameter λ. That is, the robust GPC-PID controller can be constructed by choosing λ so that the closed-loop system satisfies (26).

III. Switching structure using SVM

According to the robust controller, the stability of the closed-loop system is guaranteed in the range of uncertainty estimated beforehand. However, if the range of uncertainty is estimated widely more than necessity, the control performance becomes very conservative, and it is difficult to obtain the desirable tracking property. Therefore, the wide range of uncertainty is firstly divided into some small ranges, and the robust GPC-PID controller is designed for each range. And then, the suitable controller is selected corresponding to the result of the discrimination of the system property. The outline figure of the proposed method is shown in Fig.1.

A. Support Vector Machine (SVM)

The SVM can find the separating hyperplane using the historical data, which is called training samples. Now, if the training sample $\mathbf{x}_i \in R$ belongs to the class $y_i \in \{-1, 1\}$ and the each class can be linearly separated, the discrimination function is described as follows:

$$f(\mathbf{x}_i) = \mathbf{w}^T\mathbf{x}_i + b, \tag{28}$$

where $\mathbf{w}$ is an adjustable weight vector, and b is the bias. The separating hyperplane is the plane with $f(\mathbf{x}) = 0$, and needs to satisfy the following constraints so that the hyperplane may be uniquely determined for all training data.

$$y_i \cdot ((\mathbf{w}^T\mathbf{x}_i) + b) \geq 1 \quad (i = 1, \cdots, l). \tag{29}$$

The distance between the hyperplane and the nearest samples (called support vectors), which is called margin, is always $1/\|\mathbf{w}\|$. If the $\mathbf{w}$ to maximize the margin is selected, the discrimination function with the high accuracy generalization ability is obtained. That is, the problem of maximizing the margin is replaced with the problem of minimizing $\|\mathbf{w}\|/2$ subject to the constraints (29). Here, of course, all data cannot be linearly separated. Moreover, there may be some cases where the discrimination function becomes wrong. For such cases, the slack variables $\zeta_i \geq 0$ are introduced, and then the following problems is considered:

$$\min \qquad Q(\mathbf{w},\zeta) = \frac{1}{2}\|\mathbf{w}\|^2 + \mu\sum_{i=1}^{n}\zeta_i \tag{30}$$

$$\text{subject to} \quad y_i \cdot ((\mathbf{w}^T\mathbf{x}_i) + b) - (1-\zeta_i) \geq 0 \tag{31}$$

$$\zeta_i \geq 0 \quad \forall i, \tag{32}$$

where μ is the user-specified parameter which is determined by the trade-off between the maximum margin and the minimum classification error. Here, the Lagrange multipliers are introduced as follows:

$$\max \qquad \sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{l}\alpha_i\alpha_j y_i y_j \mathbf{x}_i\mathbf{x}_j \tag{33}$$

$$\text{subject to} \quad 0 \leq \alpha_i \leq \mu (i = 1, \cdots, l) \tag{34}$$

$$\sum_{i=1}^{l}\alpha_i y_i = 0. \tag{35}$$

One of the characteristics of the SVM is to use the technique called kernel trick. Using this kernel trick, the nonlinear separating hyperplane can be obtained. As one of the general Kernel functions, the Gaussian kernel function is given as the following equation.

$$K(\mathbf{x},\mathbf{x}') = \exp\left(-\frac{\|\mathbf{x}-\mathbf{x}'\|^2}{2\sigma^2}\right). \tag{36}$$

Using the kernel function (36) enables us to rewrite (33) as

$$\max \sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{l}\alpha_i\alpha_j y_i y_j K(\mathbf{x}_i,\mathbf{x}_j). \tag{37}$$

By solving the optimizing problem (37), the following separating function can be obtained.

$$f(\mathbf{x}) = \sum_{i,j=1}^{l}\alpha_i y_i K(\mathbf{x},\mathbf{x}_i) + b. \tag{38}$$

B. Switching structure using SVM

1) Divide the range of uncertainty: Firstly, the nominal model $G(j\omega)$ with the wide range of uncertainty is divided into some nominal models $G_i(j\omega)$ $(i = 1, 2, \cdots, n)$ with the small range of uncertainty. Next, for each nominal model $G_i(j\omega)$, the perturbed model $\tilde{G}_i(j\omega)$ is given as follows:

$$\left.\begin{array}{l}\tilde{G}_i(j\omega) = \{1 + h_i(j\omega)\}G_i(j\omega)\\ |h_i(j\omega)| \leq h_{m_i}(\omega)\\ (i = 1, 2, \cdots, n)\end{array}\right\} \tag{39}$$

Moreover, the necessary and sufficient condition for the closed-loop system in order to satisly the robust stability is introduced as follows:

$$|T_i(j\omega)|h_{m_i}(\omega) < 1. \tag{40}$$

Here, $T_i(j\omega)$ denotes the complementary sensitivity function for each nominal model $G_i(j\omega)$. Therefore, PID controller associated with each nominal model $G_i(j\omega)$ is designed so that the robust stability (40) is satisfied.

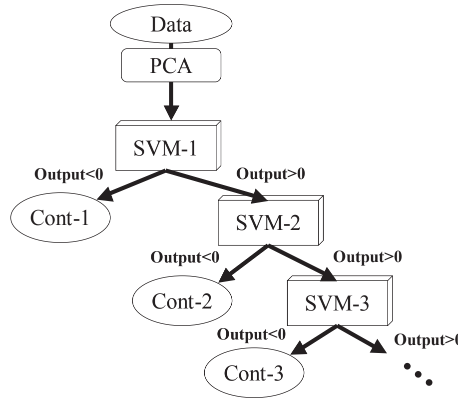

Fig. 2. Decision tree structure.

2) Input vector of SVM: In order to discriminate system property, it is necessary to prepare the input vector ($\boldsymbol{t_{in}} = [t_1, \ \cdots, \ t_{n_s}]$) with higher dimension. However, if the dimension (n_s) of the input vector becomes large, the memory and the calculation cost of the computer are exponentially increased. On the other hand, the accuracy of the classification is deteriorated due to the information lack, if the dimension of the input vector is carelessly reduced. Then, lots of information is firstly compressed into a few typical components by adopting the PCA [12].

Here, if the input/output data of the system, $((y_i, u_i), i = 1, 2, \cdots, m)$, is given in advance, the data vector $\mathbf{X}$, which represents the property of the system, is expressed by some components which are combined input/output data $\{y_i, u_i\}$. (For example, the forms such as (43) and (44)). The data vector $\mathbf{X}$ is compressed by the PCA and the relative PCA points are obtained. Then, the input vector $\boldsymbol{t_{in}}$ of the SVM is composed by these PCA points.

3) Discrimination of the property using SVM: If the number of division, n, is three or more, the usual SVM cannot be simply applied for the multi-classification problem[13], [14], because the conventional SVM is applied only for two classification problem. Therefore, the decision tree structure, which is composed using some SVMs, is introduced in this paper. And the decision tree structure is shown in Fig.2. Using the dicision tree structure, multi-classification problem can be solved.

IV. Simulation Example

In order to evaluate the effectiveness of the newly proposed scheme, a simulation example is considered. In (1), the fluctuation ranges of T, K and L are given as follows:

$$50 \leq K \leq 200, \quad 5 \leq T \leq 35, \quad L = 5. \tag{41}$$

And the nominal model for (41) has the following parameters:

$$K_n = 125, \quad T_n = 20, \quad L_n = 5. \tag{42}$$

In this paper, the range of uncertainty (all-range) was divided into three small ranges (sub-range), and nominal model

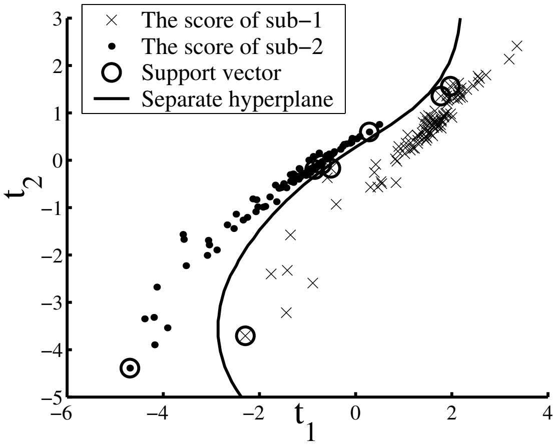

Fig. 3. The result of classifying sub-1 with sub-2.

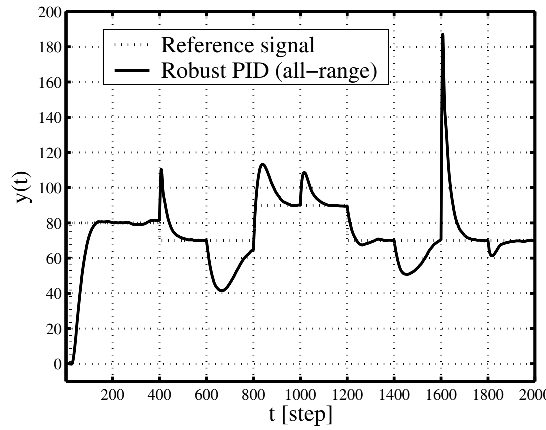

Fig. 4. Control results using only one robust PID controller.

was designed for each range. The nominal models to be designed are summarized in Table 1. Moreover, the robust PID controllers, which were designed for these nominal models, are shown in Table 2.

TABLE I

DIVISION RANGE OF THE UNCERTAINTY

	nominal		Perturbation	
	T_{ni}	K_{ni}	T	K
all-range	20	125	$5 \sim 35$	$50 \sim 200$
sub-range①	10	175	$5 \sim 15$	$150 \sim 200$
sub-range②	20	125	$15 \sim 25$	$100 \sim 150$
sub-range③	30	75	$25 \sim 35$	$50 \sim 100$

TABLE II

USER-SPECIFIED PARAMETERS λ, CONTROLLER NUMBER AND PID PARAMETER IN EACH SUB-RANGE

	sub-1	sub-2	sub-3	all-range
Controller No.	1	2	3	0
λ	5.8×10^5	4.1×10^4	9.4×10^3	9.4×10^6
k_c	0.0073	0.0269	0.0646	0.0051
T_I	8.5921	10.1835	11.5269	17.4945
T_D	1.5527	1.6279	1.6751	1.7970

Now, the number of the training sample data was 100. And the form of the data vector, $\mathbf{X_1}$ and $\mathbf{X_2}$, were given as follows:

$$\boldsymbol{X_{1i}} = \left[y(i),\ u(i-k-1),\ \frac{y(i)}{1+u(i-k-1)} \right] \quad (43)$$

$$\boldsymbol{X_{2i}} = \left[y(i),\ \frac{y(i)}{1+y(i-1)} \right], \quad (i = 1, \cdots, 100). \quad (44)$$

Next, using the data compressed by the PCA, the result of classifying "sub-1" with "sub-2" is shown in Fig.3 From Fig.3, it is clear that the good discrimination result can be obtained using the SVM combined with the PCA.

Next, the change of the controlled object was given by follows:

$$\left.\begin{array}{l}
K = 125,\ T = 20,\ L = 5 \quad (0 \le t < 400) \\
K = 200,\ T = 8,\ L = 5 \quad (400 \le t < 600) \\
K = 50,\ T = 35,\ L = 5 \quad (600 \le t < 800) \\
K = 100,\ T = 25,\ L = 5 \quad (800 \le t < 1000) \\
K = 150,\ T = 15,\ L = 5 \quad (1000 \le t < 1200) \\
K = 100,\ T = 25,\ L = 5 \quad (1200 \le t < 1400) \\
K = 50,\ T = 35,\ L = 5 \quad (1400 \le t < 1600) \\
K = 200,\ T = 8,\ L = 5 \quad (1600 \le t < 1800) \\
K = 150,\ T = 15,\ L = 5 \quad (1800 \le t \le 2000)
\end{array}\right\} \quad (45)$$

Moreover, the reference signal $w(t)$ was shown as follows:

$$w(t) = \begin{cases} 80 & (20 \le t \le 400) \\ 70 & (401 \le t \le 800) \\ 90 & (801 \le t \le 1200) \\ 70 & (1201 \le t \le 2000) \end{cases} \quad (46)$$

[Conventional method-1: Robust GPC-PID [8]**]**

The conventional robust GPC-PID controller was firstly employed. In this method, the nominal model and the fluctuation range were given as (42) and (41), respectively. Moreover, the PID parameters were given in Table 2. The control result is shown in Fig.4. From Fig.4, it is clear that the control performance becomes very conservative because the fluctuation range is very large.

[Conventional method-2: Self-tuning GPC-PID [7]**]**

Next, the conventional self tuning GPC-PID controller was employed. The control result is shown in Fig.5. From Fig.5, it is clear that the control result becomes worse in the case where the system changes from a certain condition (K:small, T:big) into another condition (K:big, T:small) (400steps, 1000steps and 1600steps). The reasons are that the system parameters were not adequately identified.

[Proposed method]

Finally, the proposed method was employed. The control result is shown in Fig.6 and the selection result of the controllers is shown in Fig.7. From Fig.6 and Fig.7, it is

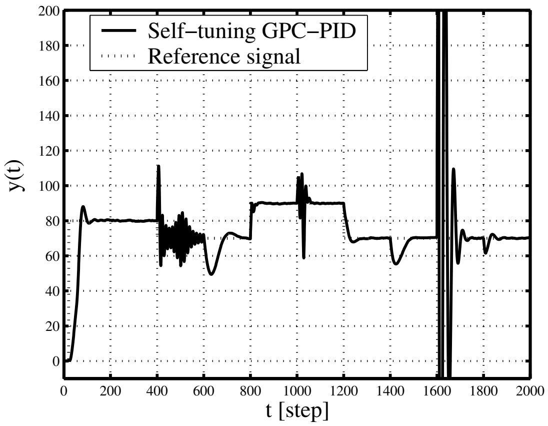

Fig. 5. Control results using the self-tuning GPC-PID controller.

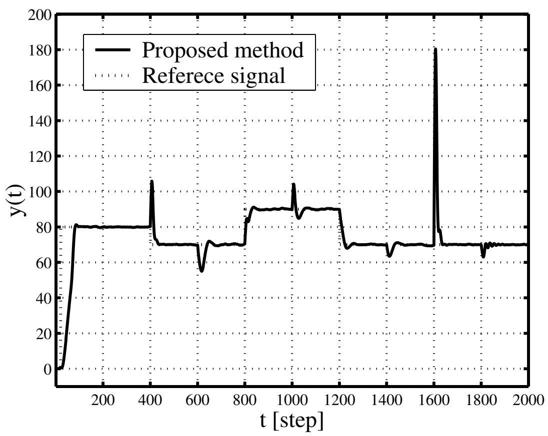

Fig. 6. Control results using the proposed method.

clear that the control performance is more improved than these conventional methods because the controller can be suitably selected corresponding to the system property using the proposed method. In addition, the arrows shown in Fig.7 indicate the fluctuation range when the robust controllers were designed. From Fig.7, it is clear that the current system is included in the corresponding fluctuation range. Therefore, the selected controller always has robust stability. That is, using the proposed method, the tracking property can be improved and the robust stability can be guaranteed simultaneously.

V. CONCLUSIONS

In this paper, the robust GPC-PID controller with the switching structure using the SVM has been proposed. The design of the robust controller has been widely studied in recent years. However, it has been problem that the tracking property may become worse in the case where the range of uncertainty is very large. Therefore, the controller design scheme has been newly proposed which plural robust controllers are switched corresponding to the system conditions in an online manner. According to the proposed control scheme, even if the discrimination results have some error margins, the stability of the control system is guaranteed,

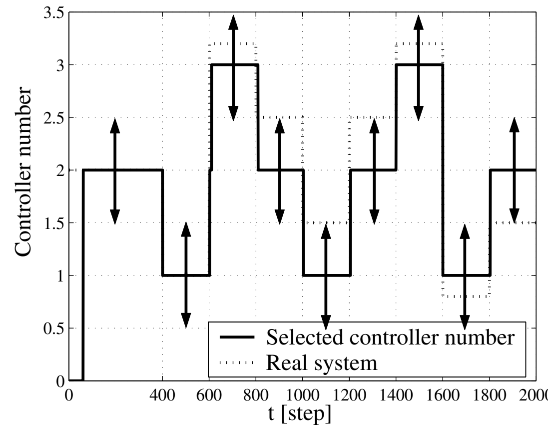

Fig. 7. Controller selection result (Proposed method).

because the selected controller is designed in consideration of uncertainty. In addition, the input vector of the SVM utilized in switching the controllers is composed by the components compressed by the PCA. A lot of information with respect to the controlled object can be expressed by few components by introducing the PCA, and it yields the suitable switching of plural controllers. Finally, the effectiveness of the proposed scheme have been evaluated on a numerical simulation example. The application for the real systems is in our future work.

REFERENCES

[1] J.G.Zieglar and N.B.Nichols, “Optimum settings for automatic controllers,” *Trans. ASME*, Vol. 64, no. 8, pp. 759–768, 1942.

[2] 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, 1952.

[3] K.J.Astrom and T.Hagglund, “PID controllers, Theory design and tuning,” *Research Triangle Park*, N.C. Instrument Society of America, 1995.

[4] K.J.Astrom and T.Hagglund, “The future of PID control,” *Control Engineering Practice*, Vol.9, pp.1163–1175, 2001.

[5] A.O'Dwyer, “Handbook of PI and PID Controller Tuning Rules,” *Imperial College Press*, 2003.

[6] D.W.Clarke, C.Mohtadi and P.S.Tuffs, “Generalized predictive control,” *Automatica*, Vol.23, pp859–875, 1987.

[7] M.Katayama, T.Yamamoto and Y.Mada, “A design of multiloop predictive self-tuning PID controllers,” *Asian Journal of Control*, Vol.4, No.4, pp.472–481, 2002.

[8] M.Katayama, T.Yamamoto and Y.Mada, “A Practical Design of Multiloop Robust PID Control Systems,” *IEEJ(Institute of Electrical Engineering in japan) Trans. on Electronics, Information, and Systems*, Vol.123-C, No.2, pp.330-337, 2003.

[9] N.Cristianini and J.S.Taylor, “An Introduction to Support Vector Machines and other kernel-based learning methods,” Cambridge Univ., 2000.

[10] V.N.Vapnik, “The Nature of Statiscal Learning Theory,” Springer, 1995.

[11] Simon S. Haykin, “Neural Networks: A Comprehensive Foundation,” Prentice Hall, 1998.

[12] I.T.Jolliffe, “Principal Component Analysis,” Springer-Verlag, 2002.

[13] K.Crammer, Y.Singer, “On the learnability and design of output codes for multiclass problems,” *Proc. Computational Learning Theory*, pp.35-46, 2000.

[14] F.Takahashi, S.Abe, “Decision-Tree-Based Support Vector Machines,” *Proc. Systems Control and Information*, Vol.16, No.3, pp. 125-132, 2003.