DESIGN AND EXPERIMENTAL EVALUATION OF A DATA-BASED SELF-TUNING PID CONTROLLER

Kenji Takao * Toru Yamamoto ** Takao Hinamoto *

* *Graduate School of Engineering, Hiroshima University*
** *Graduate School of Education, Hiroshima University*

Abstract: PID controllers have been widely employed for industrial processes. Although most process systems have nonlinear properties, the effective PID parameters tuning method for nonlinear systems have not been established yet. Therefore, in this paper, a new tuning method of PID parameters is proposed for nonlinear systems using the idea of a data-based modeling method. According to the proposed method, PID parameters can be adequately tuned by input/output data of the controlled object stored recursively in the data-base. Finally, this method is experimentally evaluated on a temperature control system.

Keywords: Data-Based modeling, PID control, intelligent control, nonlinear control, process control.

1. INTRODUCTION

In recent years, many complicated control algorithms such as adaptive control theory or robust control theory have been proposed and implemented. However, in industrial processes, PID controllers (Åström and Hägglund, 1988; J.G.Zieglar and N.B.Nichols, 1942; K.L.Chien *et al.*, 1972) have been widely employed for about 80% or more of control loops. The reasons are summarized as follows. (1) the control structure is quit simple; (2) the physical meaning of control parameters is clear; and (3) the operators' know-how can be easily utilized in designing controllers. Therefore, it is still attractive to design PID controllers. However, since most process systems have non-linear properties, it is difficult to obtain good control performances for such systems simply using the fixed PID parameters. Therefore, in adaptive tuning of these parameters, tuning methods using neural networks(NN)(S.Omatu *et al.*, 1995) and genetic algorithms(GA)(B.Porter and A.H.Jones, 1992) have been proposed until now. Since the learning cost is considerably large, it is not suitable very much to tune PID parameters in an on-line manner. That is, it is quite difficult to obtain good control performances using these conventional schemes for time-varying systems or nonlinear systems.

By the way, development of computers enables us to memorize, fast retrieve and read out a large number of data. By these advantages, the following method has been proposed: Whenever new data is obtained, the data is stored. Next, similar neighbors to the information requests, called '*queries*', are selected from the stored data. Furthermore, the local model is constructed using these neighbors. This data-based(DB) modeling method, is called *Just-In-Time (JIT)* method(A.Stenman *et al.*, 1996; Q.Zheng and H.Kimura, 2001*b*) , *Lazy Learning* method(J.Zhang *et al.*, 1997; G.Bontempi *et al.*, 1999) or Model-on-Demand(MoD) (A.Stenman, 1999), and these scheme have lots of attention in last decade.

In this paper, a design scheme of PID controllers based on the DB modeling method is discussed. A few PID controllers have been already proposed based on the JIT method(Q.Zheng and H.Kimura, 2001*a*) and the MoD method(J.Ohta and S.Yamamoto, 2004) which belong to the DB modeling methods. The JIT method is used as the purpose of supplementing the feedback controller with a PID structure. However, the tracking property is not guaranteed enough due to the nonlinearities in the case where reference signals are changed, because the controller does not includes any integral action in the whole control system. On the other hand, the latter method has a PID control structure. PID parameters are tuned by operators' skills, and they are stored in the data-base in advance. And also, a suitable set of PID parameters is generated using the stored data. However, the good control performance cannot be necessarily obtained in the case where nonlinearities are included in the controlled object and/or system parameters are changed, because PID parameters are not tuned in an on-line manner corresponding to characteristics of the controlled object.

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

2. PID CONTROLLER DESIGN BASED ON DATA-BASED MODELING METHOD

2.1 DB modeling method

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

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

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

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

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

2.2 Controller design based on DB method

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\tilde{\phi}(t) := [\ y(t+d), \cdots, y(t+d-n_y+1), \mathbf{K}(t), r(t), r(t-1), r(t-2), y(t), y(t-1), y(t-2), u(t-1), \cdots, u(t-n_u+1)\ ], \quad (11)$$

where $h(\cdot)$ denotes a nonlinear function. Here, for the simplicity, the orders of the output and input are respectively set as $n_y = 1$, $n_u = 1$ in eq.(11). Therefore, $\mathbf{K}(t)$ is given by following equations:

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

$$\bar{\phi}(t) := [\ y(t+d+1), y(t+d), r(t), r(t-1), r(t-2), y(t), y(t-1), y(t-2), u(t-1)\ ], \quad (13)$$

where $F(\cdot)$ denotes a nonlinear function. Moreover, $y(t+d)$ is given by the following equations:

$$\begin{aligned} y(t+d) &= f(\phi(t+d-1)) \\ &= f(y(t+d-1), u(t-1)) \\ &= f\{f(y(t+d-2), u(t-2)), u(t-1)\} \\ &\cdots \\ &= \bar{f}\{y(t-1), u(t-d-1), \cdots, u(t-1)\} \\ &= \bar{f}\{\hat{\phi}(t-1)\}, \end{aligned} \quad (14)$$

where $\bar{f}(\cdot)$ denotes a nonlinar function. $y(t+d)$ can be obtained using the information vector $\hat{\phi}(t-1)$ at time t. However, since $\hat{\phi}(t)$ associated with $y(t+d+1)$ can not be obtained at time t, $y(t+d+1)$ is replaced by $r(t+d+1)$ Moreover, it is assumed that the reference signals $r(\cdot)$ are piecewise constant. That is,

$$r(t-2) = r(t-1) = r(t) = \cdots = r(t+d+1). \quad (15)$$

From eq.(13) to eq.(15), the information vector $\bar{\phi}$ is newly rewritten as follows:

$$\bar{\phi}(t) := [\ r(t), y(t), y(t-1), y(t-2), u(t-1), u(t-2), \cdots, u(t-d-1)\ ]. \quad (16)$$

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

[STEP 1] Generate initial data-base

The DB modeling method cannot work if the past data is not saved at all. Therefore, PID parameters are firstly calculated using Zieglar & Nichols method(J.G.Zieglar and N.B.Nichols, 1942) or Chien, Hrones & Reswick(CHR) method(K.L.Chien *et al.*, 1972) based on historical data of the controlled object in order to generate the initial data-base. That is, $\mathbf{\Phi}(j)$ indicated in the following equation is generated as the initial data-base:

$$\mathbf{\Phi}(j) := [\ \bar{\phi}(j)\ ,\ \mathbf{K}(j)\ ], \quad j = 1, 2, \cdots, N(0) \quad (17)$$

where $\bar{\phi}(j)$ and $\mathbf{K}(j)$ are given by eq.(16) and eq.(9). Moreover, $N(0)$ denotes the number of information vectors stored in the initial data-base. Note that all PID parameters included in the initial information vectors are equal, that is, $\mathbf{K}(1) = \mathbf{K}(2) = \cdots = \mathbf{K}(N(0))$ in the initial stage.

[STEP 2] Calculate distance and select neighbors

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

$$dis(\bar{\phi}(t), \bar{\phi}(j)) = \sum_{l=1}^{5+d} \left| \frac{\bar{\phi}_l(t) - \bar{\phi}_l(j)}{\max_m \bar{\phi}_l(m) - \min_m \bar{\phi}_l(m)} \right|, \quad (18)$$

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

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

[STEP 3] Construct local model

Next, using k neighbors selected in STEP 2, the local model is constructed based on the following Linearly Weighted Average(LWA)(C.G.Atkeson and S.Schaal, 1997):

$$\mathbf{K}^{old}(t) = \sum_{i=1}^{k} w_i \mathbf{K}(i), \qquad \sum_{i=1}^{k} w_i = 1, \quad (19)$$

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

$$w_i = \sum_{l=1}^{5+d} \left(1 - \frac{[\bar{\phi}_l(t) - \bar{\phi}_l(i)]^2}{[\max_m \bar{\phi}_l(m) - \min_m \bar{\phi}_l(m)]^2} \right). \quad (20)$$

[STEP 4] Data adjustment

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

$$\mathbf{K}^{new}(t) = \mathbf{K}^{old}(t) - \eta \frac{\partial J(t+d+1)}{\partial \mathbf{K}(t)} \quad (21)$$

$$\eta := [\eta_P, \eta_I, \eta_D], \quad (22)$$

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

$$J(t+d+1) := \frac{1}{2}\varepsilon(t+d+1)^2 \quad (23)$$

$$\varepsilon(t) := y_r(t) - y(t). \quad (24)$$

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

$$y_r(t) = \frac{z^{-1}T(1)}{T(z^{-1})}r(t) \quad (25)$$

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

Here, $T(z^{-1})$ is designed based on the reference literature(T.Yamamoto and S.L.Shah, 1998). Moreover, each partial differential of eq.(21) is developed as follows:

$$\left.\begin{aligned} \frac{\partial J(t+d+1)}{\partial K_P(t)} &= -\varepsilon(t+d+1)\Delta e(t)\frac{\partial y(t+d+1)}{\partial u(t)} \\ \frac{\partial J(t+d+1)}{\partial K_I(t)} &= -\varepsilon(t+d+1)e(t)\frac{\partial y(t+d+1)}{\partial u(t)} \\ \frac{\partial J(t+d+1)}{\partial K_D(t)} &= -\varepsilon(t+d+1)\Delta^2 e(t)\frac{\partial y(t+d+1)}{\partial u(t)}. \end{aligned}\right\} \quad (27)$$

[STEP 5] Remove redundant data

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

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

[First condition]

$$dis(\bar{\phi}(t), \bar{\phi}(i)) \leq \alpha_1, \quad i = 1, 2, \cdots, N(t) - k \quad (28)$$

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

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

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

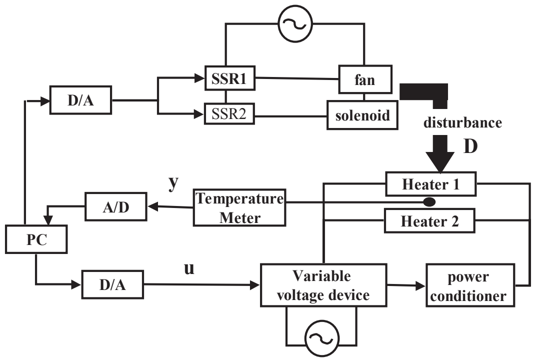

Fig. 1. The schematic figure of the experimantal temperature control system.

[Second condition]

$$\sum_{l=1}^{3} \left\{ \frac{\mathbf{K}_l(\bar{i}) - \mathbf{K}_l^{new}(t)}{\mathbf{K}_l^{new}(t)} \right\}^2 \leq \alpha_2, \quad (30)$$

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

$$\mathbf{\Phi}(\hat{i}) := \left[\bar{\phi}(\hat{i}), \ \mathbf{K}(\hat{i})\right]. \quad \hat{i} = 1, 2, \cdots \quad (31)$$

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

3. EXPERIMENTAL EVALUATION

The effectiveness of the proposed scheme is evaluated by applying for a temperature control system.

3.1 Temperature control system

The schematic figure of the experimental temperature control system is shown in Fig.1. The control objective is to regulate the heater temperature y to any desired values by manipulating tuning the input voltage u of the heater. Furthermore, let $u(t)$ be limited as 1.0[v]$\leq u(t) \leq$5.0[v]. Here, the static properties and trajectories of the system gain of this temperature control system are respectively shown in Fig.2 and Fig.3, respectively. From Fig.2 and Fig.3, it is clear that the experimental temperature control system has non-linear properties, because the static gain greatly changes at each equilibrium point. Moreover, the cyclical disturbances are given by the "*solenoid*" shown in Fig.1.

3.2 Controller Design

In order to generate the initial data-base, PID parameters are tuned by the Ziegler & Nichols

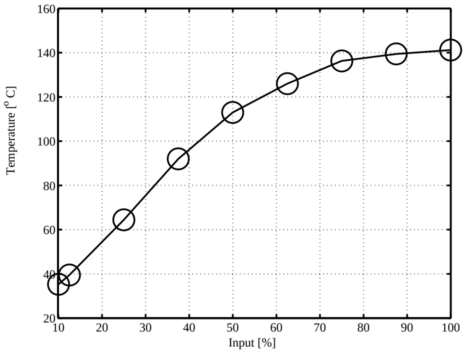

Fig. 2. Static properties of the temperature control system.

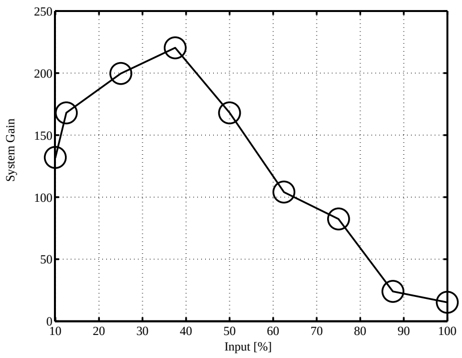

Fig. 3. Trajectories of the system gain at each equilibrium point.

(ZN) method. As the designing model, following continuous-time model $G(s)$ is considered.

$$G(s) = \frac{K}{1+Ts}e^{-LS}, \tag{32}$$

where K, T and L denote the system gain, the time-constant and the time-delay, respectively. Here, by examining the properties of the system at high temperature, these parameters are given as follows:

$$K = 80, \quad T = 100, \quad L = 10. \tag{33}$$

Moreover, PID parameters which tuned by the ZN method are given as follows:

$$K_{PC} = 0.15, \quad K_{IC} = 0.0075, \quad K_{DC} = 0.75, \tag{34}$$

where the sampling time T_s is set as 1.0[s]. According to the fixed PID method (ZN method), there is a problem that the control result becomes oscillatory using PID parameters tuned at high temperature.

3.3 Control result

In order to confirm the effectiveness of the proposed method, some control results are indicated.

First, the reference tracking properties are examined using the fixed PID controller tuned by the ZN method and the proposed method. Here the reference temperature was set as 40[°C].

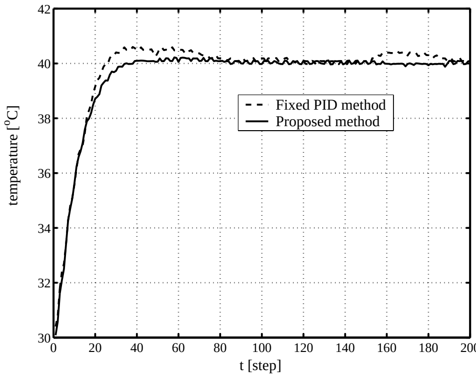

Fig. 4. Control results using the proposed method (solid line) and the fixed PID method (broken line).

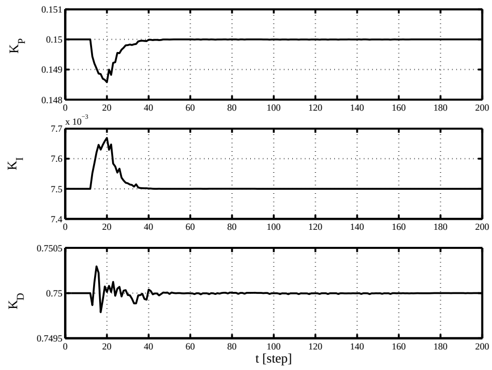

Fig. 5. Trajectories of PID parameters corresponding to Fig.4.

The control results using the fixed PID method is shown in Fig.4(broken line). From Fig.4(broken line), it can be seen that the response became oscillatory. On the other hand, the control result using the proposed method is shown in Fig.4(solid line), and trajectories of PID parameters are shown in Fig.5. From Fig.4(solid line) and Fig.5, the good response can be obtained without overshoot and oscillation because PID parameters are adequately adjusted. That is, the problem of the conventional method can be solved using the proposed method.

Next, the influence of the cyclical disturbances is examined. The cyclical disturbances $D(t)$ are given by the following equation.

$$D(t) = \begin{cases} \text{Off} & (\quad 0 \leq \text{t} < 150) \\ \text{On} & (150 \leq \text{t} < 200) \\ \text{Off} & (200 \leq \text{t} < 300) \\ \text{On} & (300 \leq \text{t} < 350) \\ \text{Off} & (350 \leq \text{t}). \end{cases} \tag{35}$$

Here, it is assumed that the temperature can be controlled adequately at 80[°C]. In this case, PID parameters given in eq.(34) are similarly used .

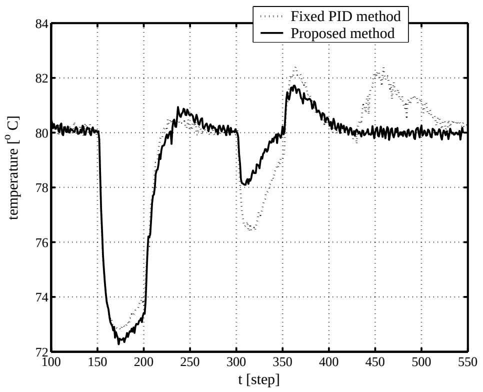

Fig. 6. Control results using the proposed method (solid line) and the fixed PID method (dotted line) for the system with cyclical disturbances.

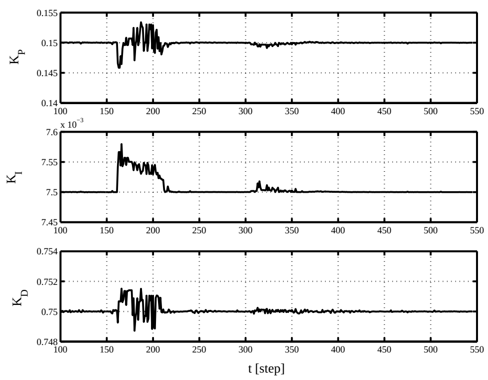

Fig. 7. Trajectories of PID parameters corresponding to Fig.6.

The control results using the fixed PID method is shown in Fig.6(dotted line). From Fig.6(dotted line), it can be seen that the response became oscillatory after 350[step]. On the other hand, the control result using the proposed method is shown in Fig.6(solid line), and trajectories of PID parameters are shown in Fig.7. From Fig.6(solid line) and Fig.7, it is clear that the influence of the disturbance can be reduced more than the fixed PID method, and the response can be obtained without oscillation.

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 verified by a experimental temperature control system.

The authors wish to thank Mr.Takaaki Yamada of OMRON Corporation for helping with the experimental evaluation.

REFERENCES

A.Stenman (1999). *Model on Demand*. Algorithms, analysis and applications, PhD thesis, Department of Electrical Engineering Linkoping University.

A.Stenman, F.Gustafsson and L.Ljung (1996). Just in time models for dynamical systems. *35th IEEE Conference on Decision and Control* pp. 1115–1120.

B.Porter and A.H.Jones (1992). Genetic tuning of digital pid controllers. *Electronics Letter* **28**, 843–844.

C.G.Atkeson, A.W.Moore and S.Schaal (1997). Locally weighted learning for control. *Artificial Intelligence Review* **11**, 75–114.

E.Nakanisi and Y.Hanakuma (1992). *The foundation and the practice of a process control(in Japanese)*. Asakura Publishing.

G.Bontempi, M.Birattari and H.Bersini (1999). Lazy learning for local modeling and control design. *International Journal of Control* **72**, 643–658.

J.G.Zieglar and N.B.Nichols (1942). Optimum settings for automatic controllers. *Trans. ASME* **64**, 759–768.

J.Ohta and S.Yamamoto (2004). Detabase-driven tuning of pid controllers (in japanese). *Trans. on SICE* **40**, 664–669.

J.Zhang, Y.Yim and J.Yang (1997). Intelligent selection of instances for prediction functions in lazy learning algorithms. *Artificial Intelligence Review* **11**, 175–191.

K.L.Chien, J.A.Hrones and J.B.Reswick (1972). On the automatic control of generalized passive systems. *Trans. ASME* **74**, 175–185.

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

Q.Zheng and H.Kimura (2001*a*). Just-in-time pid control. *The 44th Japan Joint Automatic Control Conference* pp. 336–339.

Q.Zheng and H.Kimura (2001*b*). A new just-in-time modeling method and its applications to rolling set-up modeling (in japanese). *Trans. on SICE* **37**, 640–646.

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

T.Yamamoto and S.L.Shah (1998). Design and experimental evaluation of a multivariable self-tuning pid controller. *Proc. of IEEE Conference on Control Applications* pp. 1230–1234.