# Design of a Self-tuning Predictive PI Controller for Delay Systems based on the Augmented Output

Yoichiro Ashida[1][a], Shin Wakitani[2][b] and Toru Yamamoto[2][c]
[1]*Department of Electrical Engineering and Computer Science, National Institute of Technology, Matsue College, 14-4 Nishiikumacho, Matsue, Japan*
[2]*Graduate School of Engineering, Hiroshima University, 1-4-1 Kagamiyama, Higashi-hiroshima, Japan*

Keywords: Predictive PI Controller, Predictive Control, Delay System, Data-driven, Self-tuning.

Abstract: This paper proposes an online type control parameter tuning method for a predictive PI controller. Predictive PI controller is based on a PI controller with a Smith predictor, and it is effective for a controlled object with large dead-time. Control performance of the predictive PI controller strongly depends on control parameters. Recently, some data-driven controller tuning methods have been proposed. The methods directly calculate suitable parameters from one or some sets of operating data. In addition, almost controlled processes are time-variant. In this paper, a data-driven self-tuning predictive PI controller is proposed. The effectiveness of the proposed scheme is evaluated by a simulation example.

## 1 INTRODUCTION

PID controllers(Astrom and Hagglund, 2005; Berner et al., 2018) are often applied in industries. Especially in chemical process systems, 90% or more of controllers are PID controllers. However, according to some surveys, most of the employed PID form controllers are PI controllers(Bialkowski, 1993; Sun et al., 2016). One of the reasons is the difficulty of tuning of derivative gains. To obtain good control performance for a controlled object with a long dead-time, the derivative gain should be large. However, the larger the derivative gain is, the stronger the influence of noise is. This is because a derivative element amplifies high-frequency signals. Therefore, it is difficult to tune derivative gain suitably than the other gains.

Lots of controllers which can obtain good control performance for a dead-time system are proposed. Representative examples are Smith predictors(Ingimundarson and Hagglund, 2001; Sanz et al., 2018) and model predictive controllers (MPC)(Rawlings and Mayne, 2009; Gallego et al., 2019). The Smith predictor constructs positive feedback for a conventional controller like a PI controller. By the Smith predictor, dead-time of a controlled system is removed from a closed-loop. Therefore, the conventional controller can be designed for a system without a dead-time. A MPC needs a mathematical model of the process. The input value is calculated based on an output of the mathematical model. These MPCs are sometimes employed in industries. However, both controllers predict future output, and accuracy of a mathematical model strongly affects control performance. Separately, a predictive PI controller(Hagglund, 1992; Airikka, 2014; Hassan et al., 2016) has been proposed. Although the predictive PI controller is a combination of a PI controller and a Smith predictor, a mathematical model is not required overtly and has only four parameters. Authors also propose a tuning scheme of the predictive PI controller(Ashida et al., 2019), and effectiveness for a noisy system is shown by comparing a PID controller.

Recently, data-driven controller tuning methods are proposed. According to the conventional controller parameters tuning methods, the parameters are determined based on a mathematical model of a controlled system. In contrast, data-driven methods directly compute the control parameters without any system model. Therefore, the methods can reduce the cost of the modeling and attract attention. Representative examples are Iterative Feedback Tuning (IFT)(Hjalmarsson et al., 1998; Wang and Ma, 2015), Virtual Reference Feedback Tuning (VRFT)(Campi

[a] https://orcid.org/0000-0001-7751-5270
[b] https://orcid.org/0000-0002-3850-3864
[c] https://orcid.org/0000-0001-6500-8394



Ashida, Y., Wakitani, S. and Yamamoto, T.
Design of a Self-tuning Predictive PI Controller for Delay Systems based on the Augmented Output.
DOI: 10.5220/0007919306720679
In *Proceedings of the 16th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2019)*, pages 672-679
ISBN: 978-989-758-380-3

et al., 2002; Campestrini et al., 2016) and Fictitious Reference Iterative Tuning (FRIT)(Soma et al., 2004; Kaneko, 2015). Among the data-driven methods, FRIT method requires only one set of input/output data, and some examples of actual systems are reported. However, these methods determine one set of controller parameters, and it is difficult to maintain control performance for a time-variant system. To tackle this problem, data-driven tuning methods are extended to some self-tuning controllers. Authors also propose a data-driven PID controller tuning method(Ashida et al., 2016), and the method is extended to the self-tuning PID controller(Ashida et al., 2017).

In this paper, a self-tuning predictive PI controller is proposed. The proposed controller is an extent of a self-tuning PID controller proposed by the authors(Ashida et al., 2017). A recursive least squares method(Goodwin and Sin, 1984) is employed as an adaptive algorithm. In the second section, a predictive PI controller is derived from a PI controller with a Smith predictor. In addition, it is proved that the predictive PI controller can realize complete model matching for a first-order system with a dead-time. In the third section, the proposed design method of a self-tuning predictive PI controller is described. Data-driven controller parameters tuning method is firstly explained. Next, an extension of the self-tuning controller is described. At last, the proposed control scheme is evaluated by some simulation and experimental results.

## 2 DISCRETE-TIME PREDICTIVE PI CONTROLLER

In this section, a discrete-time predictive PI control law is derived. The predictive PI controller is based on a PI controller with a Smith predictor. At first, a controlled object $G(z^{-1})$ is assumed to be the following first-order system:

$$G(z^{-1}) = G_p(z^{-1})z^{-d}, \tag{1}$$

where $G_p(z^{-1})$ denotes a delay element of the system, and $d$ denotes a dead-time which is known. A PI controller with Smith predictor can be expressed as follows:

$$u(t) = \left(K_P + \frac{K_I}{\Delta}\right)\left\{e(t) - G_p(z^{-1})\left(z^{-d} - 1\right)u(t)\right\}, \tag{2}$$

where $K_P$, $K_I$ and $u(t)$ denote a proportional gain, an integral gain and an input signal. $\Delta$ denotes the differencing operator defined by $\Delta := 1 - z^{-1}$, where $z^{-1}$ is the shift operator which means $z^{-1}y(k) = y(k-1)$. In addition, $u(t)$ also denotes a controlled input, and $e(t)$ denotes a controlled error defined as follows:

$$e(t) := r(t) - y(t), \tag{3}$$

where $r(t)$ and $y(t)$ denote a reference signal and an output signal respectively. $G_p(z^{-1})$ is assumed to be the following first-order system:

$$G_p(z^{-1}) = \frac{z^{-1}b_0}{1 + a_1 z^{-1}}. \tag{4}$$

By substituting (4), (2) can be rewritten as follows:

$$\begin{aligned} u(t) = &\left(K_P + \frac{K_I}{\Delta}\right)e(t) \\ &+ \frac{z^{-1}b_0(K_P\Delta + K_I)}{\Delta(1 + a_1 z^{-1})}(z^{-d} - 1)u(t). \end{aligned} \tag{5}$$

Next, PI gains $K_P$ and $K_I$ are set as follows:

$$\begin{cases} K_P = -\dfrac{a_1}{b_0}K_{pred} \\ K_I = \dfrac{1}{b_0}K_{pred} \end{cases}, \tag{6}$$

where $K_{pred}$ is a prediction gain. By substituting (6) to (5), the following predictive PI control law is obtained:

$$u(t) = \left(K_P + \frac{K_I}{\Delta}\right)e(t) + \frac{z^{-1}K_{pred}}{\Delta}(z^{-d} - 1)u(t). \tag{7}$$

To control a first-order system with delay shown in (1) and (4) by the predictive PI controller, a closed-loop transfer function is as follows:

$$W(z^{-1}) = \frac{K_{pred}(1 + a_1 z^{-1})z^{-(d+1)}}{1 + (K_{pred} - 1 + a_1)z^{-1} + a_1(K_{pred} - 1)z^{-2}}. \tag{8}$$

The transfer function is simplified using (6). A prediction gain $K_{pred}$ is set as follows:

$$K_{pred} = 1 + p_1, \tag{9}$$

where $p_1$ is a value included in a following first-order reference model with a dead-time:

$$G_m(z^{-1}) := \frac{1 + p_1}{1 + p_1 z^{-1}}z^{-(d+1)}. \tag{10}$$

$p_1$ is an user-specified parameter and determines a rise-time. By substituting (9) to (8), the closed-loop transfer function is rewritten as follows:

$$\begin{aligned} W(z^{-1}) = &\frac{(1 + p_1)(1 + a_1 z^{-1})}{1 + (p_1 + a_1)z^{-1} + p_1 a_1 z^{-2}}z^{-(d+1)} \\ = &\frac{1 + p_1}{1 + p_1 z^{-1}}z^{-(d+1)}. \end{aligned} \tag{11}$$

Therefore, the predictive PI controller can control a first-order system with a dead-time as a first-order reference model. When a controlled object has high-order characteristics strongly, control performance of the predictive PI controller deteriorates. Therefore, the controller should be applied for a system like a first-order system.

## 3 DESIGN OF A DATA-DRIVEN SELF-TUNING PREDICTIVE PI CONTROLLER

According to (6) and (9), a predictive PI controller can be designed if system parameters are known. To obtain the system parameters, system modeling is required. However, it takes considerable burden to make a mathematical model. In this paper, a data-driven controller design method is employed to determine controller parameters. The method calculates the controller parameters directly from a set of input/output data. In addition, the controller parameter tuning method is extended to a self-tuning controller by employing a recursive least squares method.

The predictive PI control law shown in (7) can be rewritten as follows:

$$\begin{aligned} r(t) =& \frac{1}{K_P\Delta + K_I}\Delta u(t) + y(t) \\ &+ \frac{K_{pred}}{K_P\Delta + K_I}\{u(t-1) - u(t-d-1)\}. \end{aligned} \tag{12}$$

Augmented output $\phi(t)$ is defined as

$$\begin{aligned} \phi(t) =& \frac{1}{K_P\Delta + K_I}\Delta u(t) + y(t) \\ &+ \frac{K_{pred}}{K_P\Delta + K_I}\{u(t-1) - u(t-d-1)\}. \end{aligned} \tag{13}$$

From the definition, the following relationship is obtained;

$$\phi(t) = r(t). \tag{14}$$

In the proposed method, it is desired to make the system output $y(t)$ tracks the reference model output $y_m(t)$ which is defined as

$$y_m(t) = G_m(z^{-1})r(t), \tag{15}$$

where $G_m(z^{-1})$ has been already defined by (10), and $p_1$ is determined using the following formulation:

$$p_1 = -\exp\left(-\frac{T_s}{T_m}\right). \tag{16}$$

$T_m$ denotes a time-constant of the reference model $G_m(z^{-1})$. The evaluation function $J$ is defined as follows:

$$J = \sum_{j=0}^{N} \varepsilon(j)^2, \tag{17}$$

where $N$ denotes the number of data and $\varepsilon(t)$ is defined as follows:

$$\varepsilon(t) = y(t) - G_m(z^{-1})\phi(t), \tag{18}$$

where optimized parameters are $K_P$ and $K_I$. It is not required to search $K_{pred}$ because $K_{pred}$ depends only on $p_1$ of reference model $G_m(z^{-1})$. By minimizing the evaluation function $J$, $G_m(z^{-1})\phi(t)$ becomes identical to $y(t)$. When the minimization has been finished enough, the following relationship can be obtained:

$$G_m(z^{-1})\phi(t) = y(t). \tag{19}$$

By using a relationship as (14), the following relation can be obtained:

$$y(t) = G_m(z^{-1})r(t). \tag{20}$$

Therefore, the controlled output becomes identical to the reference model output using optimized controller parameters.

The above discussions are based on deterministic systems. However, actual systems are stochastic systems. To reduce the influence of noise, the filtered input/output signals $u_f(t)$ and $y_f(t)$ are used. A delay part of the reference model is used as a filter, thus $u_f(t)$ and $y_f(t)$ are defined as follows.

$$y_f(t) := G_{mp}(z^{-1})y(t), \tag{21}$$

$$u_f(t) := G_{mp}(z^{-1})u(t). \tag{22}$$

$G_{mp}(z^{-1})$ denotes a delay part of the reference model as follows:

$$G_m p(z^{-1}) = \frac{1+p_1}{1+p_1 z^{-1}} z^{-1}. \tag{23}$$

From a view of the frequency domain, $\phi(t)$ is band-limited by the reference model and $\varepsilon(t)$ is composed with band-limited $\phi(t)$ in the proposed tuning scheme. Therefore, the reference model filter not only reduces the influence of noise but also emphasizes the important band of the signal. Hereinafter $u_f(t)$ and $y_f(t)$ are used instead of $u(t)$ and $y(t)$.

In this paper, a self-tuning predictive PI controller is designed by employing recursive least squares (RLS) method(Goodwin and Sin, 1984). However, the evaluation function $J$ cannot be minimized by RLS method because $\varepsilon(t)$ is not linear to optimized parameters $K_P$ and $K_I$. By multiplying $K_P\Delta + K_I$ to (18), the following relation is obtained:

$$\begin{aligned}(K_P\Delta + K_I)\varepsilon(t) =& (K_P\Delta + K_I)y_f(t) \\ &+ G_m(z^{-1})(K_P\Delta + K_I)\phi(t). \quad (24)\end{aligned}$$

Based on (24), $\tilde{\varepsilon}(t)$ is defined as $\tilde{\varepsilon}(t) = (K_P\Delta + K_I)\varepsilon(t)$, and it can be rewritten as follows:

$$\begin{aligned}\tilde{\varepsilon}(t) =& G_m(z^{-1})[\Delta u_f(t) \\ &+ K_{pred}\{u_f(t-1) - u_f(t-d-1)\}] \\ &- a_1\{y_f(t) - G_m(z^{-1})y_f(t)\} \\ &- a_2\{G_m(z^{-1})y_f(t-1) - y_f(t-1)\}, \quad (25)\end{aligned}$$

where $a_1$ and $a_2$ are optimized parameters and defined as follows:

$$\begin{cases} a_1 := K_P + K_I \\ a_2 := K_P \end{cases}. \quad (26)$$

As a result, a new evaluation function is defined as follows:

$$\tilde{J} = \frac{1}{N}\sum_{j=0}^{N}\tilde{\varepsilon}(j)^2. \quad (27)$$

$\tilde{\varepsilon}(t)$ is linear to $a_1$ and $a_2$, then a minimization of $\tilde{J}$ is a least squares problem. $a_1$ and $a_2$ are calculated recursively using the following recursive least squares algorithm, and $a_1$ and $a_2$ are converted to $K_P$ and $K_I$ by (26).

$$\hat{\theta}(t) = \hat{\theta}(t-1) + K(t)\tilde{\varepsilon}(t), \quad (28)$$

$$K(t) = \frac{P(t-1)\varphi(t)}{\omega + \varphi^T(t)P(t-1)\varphi(t)}, \quad (29)$$

$$P(t) = \frac{1}{\omega}\left[P(t-1) - \frac{P(t-1)\varphi(t)\varphi^T(t)P(t-1)}{\omega + \varphi^T(t)P(t-1)\varphi(t)}\right], \quad (30)$$

where $\omega$ is a forgetting factor, and $\varepsilon\tilde{(t)}$, $\theta(t)$ and $\varphi(t)$ are calculated as follows:

$$\begin{aligned}\tilde{\varepsilon}(t) :=& G_m(z^{-1})[\Delta u_f(t) \\ &+ K_{pred}\{u_f(t-1) - u_f(t-d-1)\}] \\ &- \hat{\theta}^T(t-1)\varphi(t) \quad (31)\end{aligned}$$

$$\hat{\theta}(t) := [\hat{a}_1(t),\ \hat{a}_2(t)]^T, \quad (32)$$

$$\begin{aligned}\varphi(t) :=& [y_f(t) - G_m(z^{-1})y_f(t), \\ & G_m(z^{-1})y_f(t-1) - y_f(t-1)]^T. \quad (33)\end{aligned}$$

An initial value of covariance matrix $P(t)$ and $\hat{\theta}(t)$ which is estimated vector of $a_i(t)$ are determined by the following equations:

$$P(0) = \alpha I \quad (34)$$

$$\hat{\theta}(0) = [\hat{a}_1(0),\ \hat{a}_2(0)]^T, \quad (35)$$

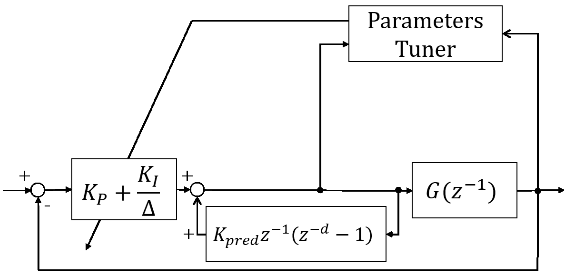


Figure 1: Block diagram of the proposed self-tuning controller.

where $\alpha$ is determined under a condition of $\alpha > 0$, and $I$ is a $2 \times 2$ matrix.

A block diagram of the proposed scheme is shown in Fig. 1. A predictive PI controller can be divided into a PI controller and a predictor. $K_P$ and $K_I$ are the optimized parameters, thus only PI controller is tuned recursively and the predictor is fixed.

## 4 SIMULATION RESULT

In this section, the proposed controller is evaluated by the following time-variant system.

(i) $t \leq 3500$

$$G(s) = \frac{390s + 260}{35s^4 + 209s^3 + 418s^2 + 316s + 7}e^{-100s}, \quad (36)$$

(ii) $t > 3500$

$$G(s) = \frac{312s + 208}{70s^4 + 418s^3 + 836s^2 + 632s + 14}e^{-100s}. \quad (37)$$

The system is discretized by $T_s = 1$ s and the following discrete system can be obtained:

(i) $t \leq 3500$

$$\begin{aligned}y(t) =& 1.33y(t-1) - 0.41y(t-2) + 0.07y(t-3) \\ &- 0.003y(t-4) + 0.55u(t-101) \\ &+ 0.36u(t-102) - 0.29u(t-103) \\ &- 0.02u(t-104) + \xi(t), \quad (38)\end{aligned}$$

(ii) $t > 3500$

$$\begin{aligned}y(t) =& 1.33y(t-1) - 0.41y(t-2) + 0.07y(t-3) \\ &- 0.003y(t-4) + 0.28u(t-101) \\ &+ 0.18u(t-102) - 0.15u(t-103) \\ &- 0.01u(t-104) + \xi(t), \quad (39)\end{aligned}$$

where $\xi(t)$ denotes the Gaussian white noise sequence with zero mean and variance $0.1^2$. A reference model

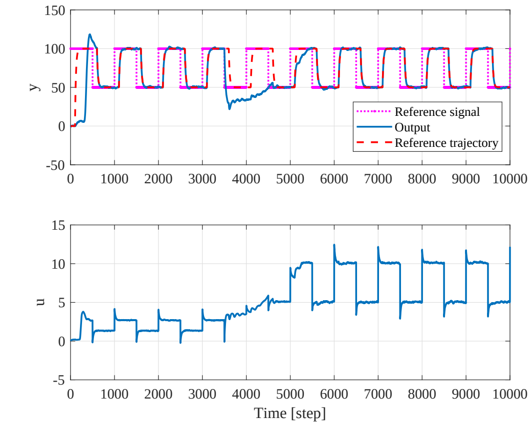


Figure 2: Control result using the proposed method.

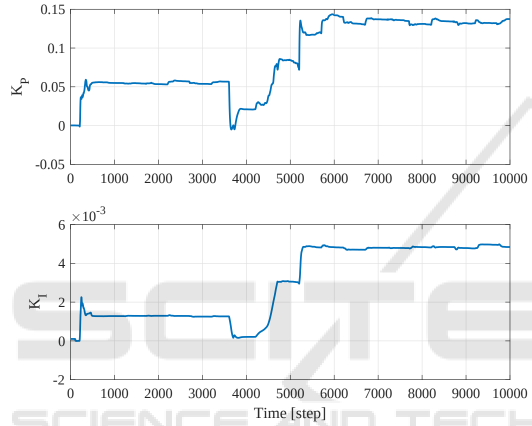


Figure 3: Trajectories of the control parameters corresponding to Fig. 2.

was determined as the following system:

$$G_m(z^{-1}) = \frac{0.05z^{-1}}{1-0.95z^{-1}}z^{-100}, \tag{40}$$

where time-constant was set as $T_m = 20$ s and dead-time was set as known. The control result using the proposed method is shown in Fig. 2, and trajectories of the controller parameters are also shown in Fig. 3. Parameters of RLS method was as follows:

$$\alpha = 1,\ \omega = 0.995. \tag{41}$$

Before and after the characteristics of the system were changed, a good control performance was obtained. From Fig. 3, the PI gains were tuned adaptively after the varying of the controlled object. However, just after the varying of the system, the PI parameters did not converge quickly, and the control performance was strongly deteriorated. It is future work to solve the problem.

At last, influence of the time-constant of the reference model is discussed. Figure 4 shows the control

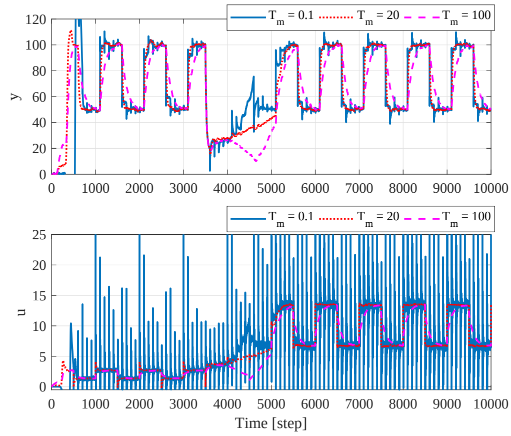


Figure 4: Controlled outputs corresponding to the time-constant of the reference model.

outputs when $T_m$ were set as $0.1, 20,$ and 100. The reference signal and designed parameters except the reference model were the same as Figure 2. When $T_m = 20$, the result is likely the same as 2, and when $T_m = 100$, there were no overshoots. However, the output sometimes became spike-like when $T_m = 0.1$. In this simulation, there is a mismatch. The reason of the mismatch is that the controlled object is high-order although the predictive PI controller is designed for a first-order system. From Figure 4, the influence of the mismatch tends to be larger as $T_m$ becomes smaller. Therefore, it is recommended that a reference model with the slow response should be applied firstly, and $T_m$ should become smaller gradually in an actual usage.

## 5 EXPERIMENTAL RESULT

In this section, a pilot-scale tank system is used to evaluate the proposed method under the condition more realistic than the simulation example. Fig. 5 shows an appearance of the system, and the corresponding schematic figure is illustrated in Fig. 6. The controlled object is to regulate the temperature of the mixed water. There are two valves which regulate the flow of the cold and hot water respectively. In this result, the opening ratio of the cold water's valve $u_c(t)$ was constant as 40%. A predictive PI controller determined the opening ratio of the hot water's valve $u_h(t)$ to regulate the temperature. In this section, a sampling time $T_s$ is 5 s. The experimental system has a transfer delay as a dead-time. The delay is about 30 seconds. In addition, a time-constant is about 30 seconds. Therefore, ratio of dead-time and time-constant is about 1. A control result by using the proposed method is shown in Fig. 7, and the trajectories of the

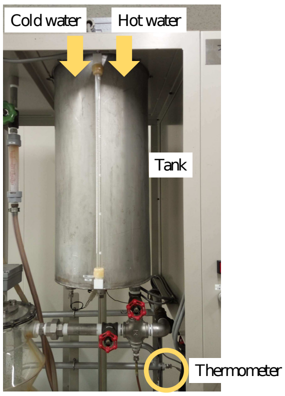


Figure 5: Appearance of the experimental temperature control system.

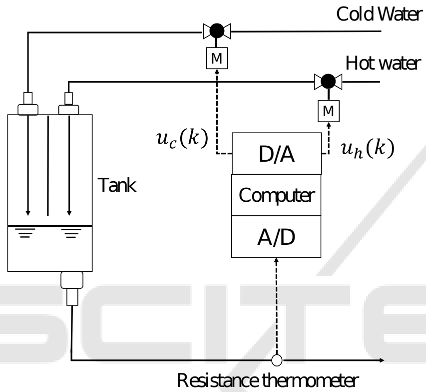


Figure 6: Schematic figure of the experimental temperature control system.

PI gains are also shown in Fig. 8. In this result, parameters of the proposed method were set as follows: $\alpha = 100, \omega = 0.995$. A reference model was determined as the following system:

$$G_m(z^{-1}) = \frac{0.22z^{-1}}{1-0.78z^{-1}}z^{-7}, \tag{42}$$

where time-constant was set as 20 s and dead-time was set as $d = 7$. It is difficult to realize a trajectory of the reference model without dead-time compensation because the dead-time is longer than the time-constant in the reference model. The initial value of $\hat{\theta}(t)$ was determined as follows:

$$\hat{\theta}(0) = [2, 1]^T. \tag{43}$$

This led the following initial PID gains:

$$K_P(0) = 1, K_I(0) = 1. \tag{44}$$

From the control results, likely desired response was obtained as time went on. However, control performance sometimes deteriorated in the transient state.

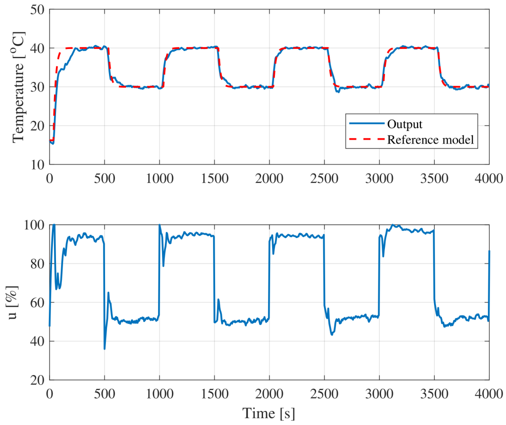


Figure 7: Experimental result by using the proposed method.

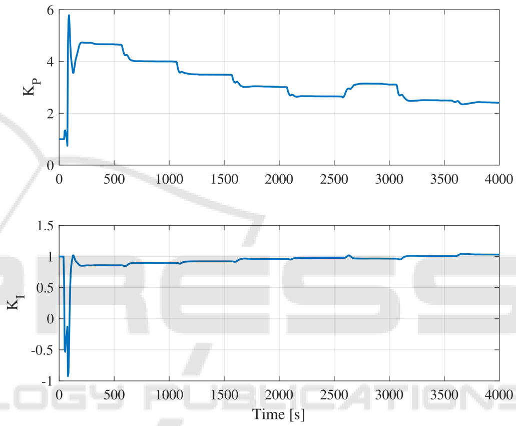


Figure 8: Trajectories of the PI gains corresponding to Fig. 7.

A mismatch between the controller and the controlled object is considered to cause deterioration. Although the experimental system is a high-order system, the controller is designed for a first-order controlled object.

At last, the role of the predictive term is considered. the predictive PI control low is as follows:

$$u(t) = \left(K_P + \frac{K_I}{\Delta}\right)e(t) + \frac{z^{-1}K_{pred}}{\Delta}(z^{-d}-1)u(t). \tag{45}$$

PID control low is also shown in follows:

$$u_{PID}(t) = \left(K_P + \frac{K_I}{\Delta}\right)e(t) + K_D\Delta e(t), \tag{46}$$

where the third term is a derivative term and $K_D$ denotes a derivative gain. It is well-known that the derivative term works like a brake. When PI gains are large, sometimes large input is calculated by PI

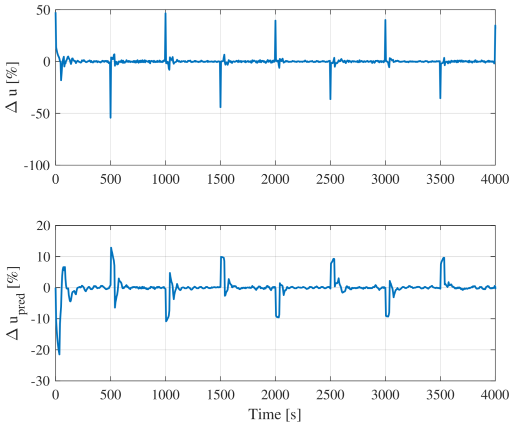


Figure 9: Comparison between $\Delta u(t)$ and $\Delta u_{pred}(t)$.

terms and an overshoot is occurred. In contrast, the derivative term predicts a future error and calculate an input which reduces the overshoot based on the estimated error. The derivative term predicts a future error under the assumption that the error is a proportional function. However, the assumption is unrealistic, and a large derivative gain sometimes causes problems. In contrast, the predictive term of the predictive PI controller is based on a Smith predictor when a controlled object is first order. Therefore, the predictive PI controller employs more advanced prediction than the PID controller although the predictive and derivative terms have the same role.

Next, the consideration above is checked using the experimental result. The input signal calculated by the predictive term is defined as:

$$u_{pred}(t) = \frac{z^{-1}K_{pred}}{\Delta}(z^{-d} - 1)u(t). \quad (47)$$

Fig. 9 shows a comparison between $\Delta u(t)$ and $\Delta u_{pred}(t)$. According to Fig. 9, the prediction term worked like a brake. For example, $\Delta u(t)$ was about $-50$ at 1505 second to lower the temperature. In contrast, $\Delta u_{pred}(t)$ was about 10 between 1505 and 1535 seconds. The 30 seconds between 1505 and 1535 mean the estimated dead-time, and the predictive term tried to suppress a large input signal in the period. Thus, the control result shows that the predictive term has the same role as the derivative term.

## 6 CONCLUSIONS

In this paper, a discrete-time predictive PI controller has been discussed, and a data-driven self-tuning design method has also been proposed. Features of the proposed controller are summarized as follows:

- The proposed controller can realize fast rise time for a system with long dead-time.
- Any system model is not required to calculate PI gains.
- Only PI gains are tuned in an on-line manner.

The proposed control scheme has been evaluated by some numerical and experimental examples. In particular, the role of the predictive term included in the proposed controller has been mentioned using the experimental example.

## REFERENCES

Airikka, P. (2014). Robust predictive pi controller tuning. *IFAC Proceedings Volumes*, 47(3):9301 – 9306. 19th IFAC World Congress.

Ashida, Y., Hayashi, K., Wakitani, S., and Yamamoto, T. (2016). A novel approach in designing pid controllers using closed-loop data. In *2016 American Control Conference (ACC)*, pages 5308–5313.

Ashida, Y., Wakitani, S., and Yamamoto, T. (2017). Design of an implicit self-tuning pid controller based on the generalized output. In *Proc. of the 20th IFAC World Congress*, pages 14511 – 14516.

Ashida, Y., Wakitani, S., and Yamamoto, T. (2019). Design of a data-driven pi controller. In *Proc. of the 2020 International Conference on Artificial Life and Robotics*.

Astrom, K. and Hagglund, T. (2005). *Advanced PID Control*. International Society of Automation, North Carolina.

Berner, J., Soltesz, K., Hagglund, T., and Astrom, K. (2018). An experimental comparison of pid autotuners. *Control Engineering Practice*, 73:124–1133.

Bialkowski, W. (1993). Dreams versus reality: a view from both sides of the gap: manufacturing excellence with come only through engineering excellence. *Pulp & Paper Canada*, 94:19–27.

Campestrini, L., Eckhard, D., Chia, L. A., and Boeira, E. (2016). Unbiased mimo vrft with application to process control. *Journal of Process Control*, 39:35 – 49.

Campi, M., Lecchini, A., and Savaresi, S. (2002). Virtual reference feedback tuning: a direct method for the design of feedback controllers. *Automatica*, 38(8):1337 – 1346.

Gallego, A. J., Merello, G. M., Berenguel, M., and Camacho, E. F. (2019). Gain-scheduling model predictive control of a fresnel collector field. *Control Engineering Practice*, 82:1–13.

Goodwin, G. C. and Sin, K. S. (1984). *Adaptive Filtering Prediction and Control*. Prentice-Hall, Upper Saddle River.

Hagglund, T. (1992). A predictive pi controller for processes with long dead times. *IEEE Control Systems Magazine*, 12(1):57–60.

Hassan, S. M., Ibrahim, R., Saad, N., Asirvadam, V. S., and Chung, T. D. (2016). Predictive pi controller for wireless control system with variable network delay and disturbance. In *2016 2nd IEEE International Symposium on Robotics and Manufacturing Automation (ROMA)*, pages 1–6.

Hjalmarsson, H., Gevers, M., Gunnarsson, S., and Lequin, O. (1998). Iterative feedback tuning: theory and applications. *IEEE Control Systems Magazine*, 18(4):26–41.

Ingimundarson, A. and Hagglund, T. (2001). Robust tuning procedures of dead-time compensating controllers. *Control Engineering Practice*, 9:1195–1208.

Kaneko, O. (2015). Fictitious reference iterative tuning of internal model controllers for a class of nonlinear systems. In *2015 IEEE Conference on Control Applications (CCA)*, pages 88–94.

Rawlings, J. B. and Mayne, D. Q. (2009). *Model Predictive Control Theory and Design*. Nob Hill Publishing, Madison.

Sanz, R., Garcia, P., and Albertos, P. (2018). A generalized smith predictor for unstable time-delay siso systems. *ISA Transactions*, 72:197–204.

Soma, S., Kaneko, O., and Fujii, T. (2004). A new approach to parameter tuning of controllers by using one-shot experimental data (in japanese). *Transactions of the Institute of Systems, Control and Information Engineers*, 17(12):528–536.

Sun, L., Li, D., and Lee, K. Y. (2016). Optimal disturbance rejection for pi controller with constraints on relative delay margin. *ISA Transactions*, 63:103–111.

Wang, Y. and Ma, G. (2015). Fault-tolerant control of linear multivariable controllers using iterative feedback tuning. *International Journal of Adaptive Control and Signal Processing*, 29(4):457–472.