Design of a Data-driven Predictive-PI Controller

Yoichiro Ashida
Graduate School of Engineering, Hiroshima University,1-4-1 Kagamiyama
Higashihiroshima city, Hiroshima, Japan

Shin Wakitani
Graduate School of Engineering, Hiroshima University,1-4-1 Kagamiyama
Higashihiroshima city, Hiroshima, Japan

Toru Yamamoto
Graduate School of Engineering, Hiroshima University,1-4-1 Kagamiyama
Higashihiroshima city, Hiroshima, Japan
E-mail: ashida-yoichiro@hiroshima-u.ac.jp, wakitani@hiroshima-u.ac.jp, yama@hiroshima-u.ac.jp
http://www.cse.hiroshima-u.ac.jp/

Abstract

PID controllers have been widely employed in real processes. However, for processes with long dead time, the control performance obtained by PID controllers are limited. The predictive PI controller can control such processes efficiently and have only three parameters. However, it is desired to determine the parameters automatically. In this paper, a data-driven design method for a predictive PI controller is proposed. The effectiveness of the proposed scheme is evaluated by a simulation example.

Keywords: Predictive control, predictive PI controller, data-driven, process control, long dead time.

1. Introduction

PID controllers[1] have been widely employed in real processes. Especially in chemical processes, 90% of the control loops are controlled by PID controllers. However, for processes with long dead time, the control performance obtained by a PID controller is limited. This is because prediction by a derivative element is not very efficient for such processes. Furthermore, the derivative part is sometimes switched off because the part amplifies the influence of noise, and it is not easy to tune the derivative gain suitably. Nevertheless, PID controllers are main controllers because PID controllers only have three parameters and can be tuned by "trial and error". To derive good control performance from the processes with long dead time, some predictive controllers have been proposed. Among them, a predictive PI controller[2-4] has only three parameters, and the controller can be tuned manually like a PID controller. Although a predictive PI controller can be tuned by "trial and error", it is desired to determine control parameters automatically. The authors have proposed a data-driven design method of PID controllers[5]. This method can be employed in designing predictive PI controllers.

In this paper, the properties of a discrete predictive PI controller are explained. In addition, a data-driven design method for a predictive PI controller is proposed. In the proposed method, control parameters of the predictive PI controller are calculated automatically from one set of operation data. The effectiveness of the proposed scheme is evaluated by a simulation example.

$$W(z^{-1}) = \frac{\{b_0(K_P + K_I)z^{-1} + b_0 K_P z^{-2}\}z^{-d}}{1 + (K_{pred} - 1 + a_1)z^{-1} + a_1(K_{pred} - 1) - \{K_{pred} - b_0(K_P + K_I)\}z^{-(d+1)} - (a_1 K_{pred} + b_0 K_P)z^{-(d+2)}} \quad (6)$$

2. Predictive PI controller

2.1. *Definition of the predictive PI controller*

A control law of a PID controller is written as follow:

$$u(t) = K_P e(t) + K_I \frac{e(t)}{\Delta} + K_D \Delta e(t), \quad (1)$$

where $u(t)$ and $e(t)$ denotes a controlled error and input, respectively. Δ denotes the differencing operator defined by $\Delta := 1 - z^{-1}$, where z^{-1} is the shift operator which means $z^{-1}e(k) = e(k-1)$. The last term is a derivative element, and it compensate a dead time. However, the derivative does not effective for a long dead time. In addition, the derivative amplifies high frequency signals such as sensor noise. Therefore, PI controllers are sometimes used.

To tackle this problem, a predictive PI controller has been proposed as follows:

$$u(t) = K_P e(t) + K_I \frac{e(t)}{\Delta} + \frac{K_{pred}}{\Delta}\{u(t-1) - u(t-d-1)\} \quad (2)$$

$$= k_c[e(t) + \frac{T_s}{T_i}\frac{e(t)}{\Delta} + \frac{T_{pred}}{T_s}\frac{1}{\Delta}\{u(t-1) - u(t-d-1)\}], \quad (3)$$

where L denotes a dead time, and a prediction term of the predictive PI controller is based on a smith predictor. This controller has only 3 parameters.

2.2. *Properties of the predictive PI controller*

The controlled system is assumed to be the following stable first order system with a dead time:

$$G(z^{-1}) = \frac{b_0 z^{-1}}{1 + a_1 z^{-1}} z^{-d}, \quad (4)$$

and a reference model is designed as follows:

$$G_m(z^{-1}) = \frac{(1 + p_1)z^{-1}}{1 + p_1 z^{-1}} z^{-d}. \quad (5)$$

The closed-loop transfer function obtained by the predictive PI controller is shown as e.q. (6). The influence of the dead time appears at the third and fourth terms of the denomination of e.q. (6). The control parameters should be determined to remove the influence. Therefore, K_P and K_I should be determined as follows:

$$K_P = -\frac{b_0}{a_1} K_{pred}, K_I = \frac{1}{b_0} K_{pred}. \quad (7)$$

By using these parameters, e.q. (6) can be rewritten as follows:

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

When K_{pred} is set as $K_{pred} = 1 + p_1$, the following relation is obtained because the system is stable:

$$W(z^{-1}) = \frac{(1 + p_1)(1 + a_1 z^{-1})z^{-(d+1)}}{1 + (p_1 + a_1)z^{-1} + a_1 p_1 z^{-2}} = \frac{(1 + p_1)}{1 + p_1 z^{-1}} z^{-(d+1)}. \quad (9)$$

Therefore, the predictive PI controller can control a first order and dead time system.

3. Tuning of the control parameters

3.1. *Design of the predictive PI controller*

If the system parameters are known, control parameters can be determined as discussed above. However, they are rarely known. In this section, a data-driven design method is proposed.

E. q. (2) is rewritten as follows:

$$r(t) = \frac{1}{K_P + K_I}\Delta u(t) + \frac{K_P}{K_P + K_I}\{r(t-1) - y(t-1)\} + \frac{K_{pred}}{K_P + K_I}\{u(t-1) - u(t-d-1)\} + y(t). \quad (10)$$

Augmented output $\Phi(t)$ is defined as follows:

$$\Phi(t) = a_1 \Delta u(t) + a_2\{\Phi(t-1) - y(t-1)\} + a_3\{u(t-1) - u(t-d-1)\} + y(t), \quad (11)$$

where coefficients a_1, a_2, and a_3 are defined as follows:

$$a_1 := \frac{1}{K_P + K_I}, a_2 := \frac{K_P}{K_P + K_I}, a_3 := \frac{K_{pred}}{K_P + K_I}. \quad (12)$$

From e.q. (10) to (12), the following relationship can be obtained:

$$r(t) = \Phi(t). \quad (13)$$

The evaluation function J is defined as follows:

$$J = \frac{1}{N}\sum_{j=1}^{N} \varepsilon(j)^2, \quad (14)$$

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

$$\varepsilon(t) := y(t) - G_m(z^{-1})\phi(t). \quad (15)$$

By minimizing the evaluation function J, the following relationship can be obtained:

$$G_m(z^{-1})\phi(t) \rightarrow y(t). \tag{16}$$

From e. q. (11), parameters to be optimized are a_i $(i = 1, 2, 3)$. When the minimization has been finished, it leads the following relationship:

$$y(t) \rightarrow G_m(z^{-1})\phi(t). \tag{17}$$

Therefore, the reference response is realized by using the optimized a_i. a_i can be converted to control parameters using the following equations:

$$K_P = \frac{a_2}{a_1}, K_I = \frac{1-a_2}{a_1}, K_{pred} = \frac{a_3}{a_1}, \tag{18}$$

$$k_c = \frac{a_2}{a_1}, T_i = \frac{a_2}{1-a_2}T_s, T_{pred} = \frac{a_3}{a_2}T_s. \tag{19}$$

3.2. *Extend the design method*

If the system is strictly first order and the dead time is known, the above discussions are directly applicable. However, most controlled systems are high order and have unknown dead times. Then, the design method is extended in this section.

In the previous design method, the augmented error is defined as e. q. (15). As a result, a purpose of the minimization of e. q. (14) is exact model matching. However, sometimes control parameters are calculated too large when the system is a high order or the dead time of the controller is incorrect.

To tackle this problem, e. q. (13) is rewritten as follows:

$$\varepsilon(t) \coloneqq y(t) - G_m(z^{-1})\phi(t) + \lambda G_m(z^{-1})\Delta u(t). \tag{20}$$

The third term denotes a variation of the input signal. λ is a weight of the variation of the input, and it is a user-specified parameter. E. q. (16) is rewritten as follows:

$$\begin{aligned}\varepsilon(t) \coloneqq\ & y(t) - (a_1 - \lambda)G_m(z^{-1}) \\ & -a_2 G_m(z^{-1})\{\Phi(t-1) - y(t-1)\} \\ & -a_3 G_m(z^{-1})\{u(t-1) - u(t-d-1)\}. \end{aligned}\tag{21}$$

By using e. q. (21), control parameters are as follows:

$$K_P = \frac{a_2}{a_1+\lambda}, K_I = \frac{1-a_2}{a_1+\lambda}, K_{pred} = \frac{a_3}{a_1+\lambda}, \tag{22}$$

$$k_c = \frac{a_2}{a_1+\lambda}, T_i = \frac{a_2}{1-a_2}T_s, T_{pred} = \frac{a_3}{a_2}T_s. \tag{23}$$

Normally, the weight λ is set as $\lambda \geq 0$. From e. q. (23), a_1 affects only k_c. This means that by tuning λ, an entire gain of a controller is tuned. When the system is a high order or dead time is unknown, λ should be large value and a conservative controller should be designed.

The influence of λ for dead time compensation is considered below. The third and fourth terms of the denomination of e. q. (6) must be zero. By using e. q. (18), the terms become zero, and the following relationships are obtained by substituting e. q. (18) to the terms of e. q. (6):

$$K_{pred} - b_0(K_P + K_I) = 0 \Leftrightarrow a_3 - b_0 = 0, \tag{24}$$

$$a_1 K_{pred} + b_0 K_P = 0 \Leftrightarrow a_1 a_3 - b_0 a_2 = 0. \tag{25}$$

When e. q. (22) is substituted instead of e. q. (18), the terms are rewritten as follows:

$$K_{pred} - b_0(K_P + K_I) = \frac{a_3 - b_0}{a_1 + \lambda}, \tag{26}$$

From e. q. (24) to (27), the third and fourth terms of the denomination of e. q. (6) are zero regardless of the value of λ.

Hence, λ does not affect the performance to compensate for the influence of the dead time.

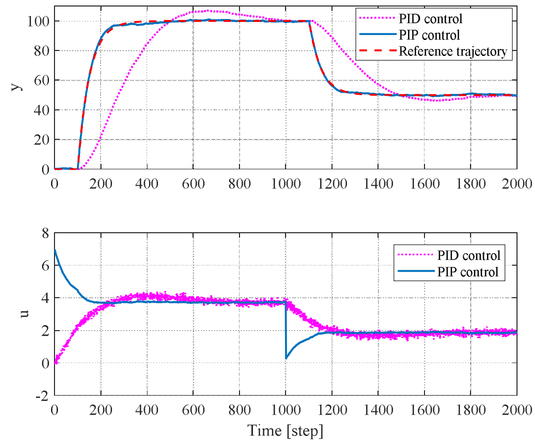

Fig. 1. Control result using PID and predictive PI controllers.

4. Simulation example

In this section, a simulation example is shown by using the following controlled system.

$$G(s) = \frac{18s + 27}{10s^3 + 30s^2 + 100s + 1}e^{-100s}. \tag{28}$$

The system was discretized by sampling time $T_s = 1$ s, and the following discrete system is obtained:

$$G(z^{-1}) = \frac{0.38z^{-1} + 0.04z^{-2} - 0.02z^{-3}}{1 - 0.57z^{-1} - 0.37z^{-2} - 0.05z^{-3}}z^{-100}. \tag{29}$$

A Gaussian white noise with zero means and 0.1^2 variance was added in the simulation.

The control result is shown as Fig. 1. There are two results using PID control with Chien, Hrones, and Reswick method and the proposed control. An initial operating data is required to apply the proposed method. The operating data obtained by PID control was used as the initial data.

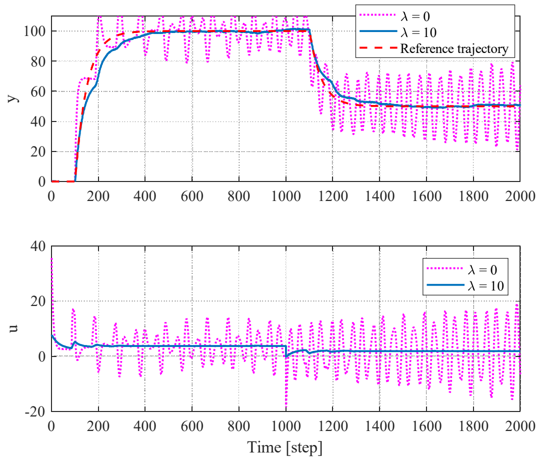

Fig. 2. Control result of predictive PI controller designed as $\lambda = 0$, and $\lambda = 10$.

In the proposed method, the reference model was made as $p_1 = -0.98$ with known dead time, and $\lambda = 0$. The optimization method is *fminserch* function of MATLAB R2017b. Control parameters were as follows:

$$\text{PID}: K_P = 0.02, K_I = 0.0002, K_D = 1.11, \quad (30)$$

$$\text{Predictive PI}: K_P = 0.07, K_I = 0.0007, K_{pred} = 0.02. \quad (31)$$

By using the proposed predictive PI control, the more aggressive response was realized than PID control and tracked the reference trajectory. In addition, the predictive PI controller did not amplify the influence of noise in contrast to the PID control. This is because there are no derivative elements in the predictive PI controller.

Next, the dead time used in designing predictive PI controller was set as 85. The other parameters were set as the same as the previous result, and the initial data was also the same. The control results by setting $\lambda = 0$, and $\lambda = 10$ are shown as Fig. 2. When $\lambda = 0$, the closed loop transfer function became unstable. In contrast, when $\lambda = 10$, the transfer function was stable while the control performance was worse than the reference model.

The calculated control parameters were as follows:

$$\lambda = 0: K_P = 0.35, K_I = 0.003, K_{pred} = 0.14, \quad (32)$$

$$\lambda = 10: K_P = 0.08, K_I = 0.0008, K_{pred} = 0.03. \quad (33)$$

The parameters are rewritten to the form of e.q. (19) as follows:

$$\lambda = 0: k_c = 0.35, T_i = 102.5, K_{pred} = 0.39, \quad (34)$$

$$\lambda = 10: k_c = 0.07, T_i = 102.5, K_{pred} = 0.39, \quad (35)$$

It is clear that only k_c was varied by changing λ. This result confirms the discussion at **3.2**.

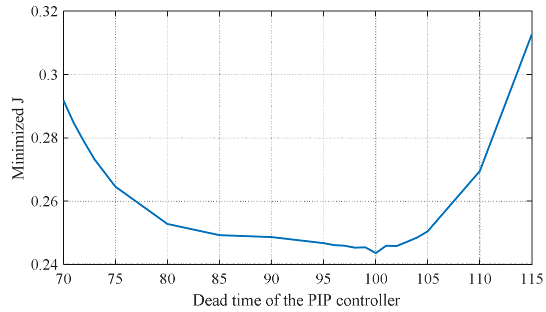

Fig. 3. The minimized evaluation values J corresponding to the dead time of the predictive PI controller when $\lambda = 0$.

At last, the minimized evaluation values J corresponding to the dead time of the predictive PI controller when $\lambda = 0$ are shown as Fig. 3.

The minimum evaluation value is calculated when dead time is 100 steps, and 100 steps are the same as the true dead time of the system. Then, the dead time can be determined based on the evaluation function. However, in the actual systems, it is considered that exact dead time cannot be calculated because of various reasons like nonlinearity. Therefore, it is recommended that λ is set as a nonzero value.

5. Conclusions

In this paper, a consideration of a predictive PI controller is presented, and a data-driven design method is proposed. The controller is effective for systems with long dead time. The effectiveness of the proposed method is confirmed by a simulation example.

References

1. K. J. Åström and T. Hägglund, *Advanced PID Control*, (International Society of Automation, North Carolina, 2006).
2. T. Hagglund, A predictive PI controller for processes with long dead times, *IEEE Control Systems Magazine* **12**(1) (1992) 57–60.
3. T. Hagglund, An industrial dead-time compensating PI controller, *Control Engineering Practice* **4**(6) (1996) 749–756.
4. P. Airikka, Robust Predictive PI Controller Tuning, *IFAC Proceedings Volumes* **47**(3) (2014) 9301–9306.
5. Yoichiro Ashida, Kayoko Hayashi, Shin Wakitani, and Toru Yamamoto, A novel approach in designing PID controllers using closed-loop data, *Proc. of the American Control Conference 2016* (2016) 5308–5313.