12th IFAC International Symposium on
Advanced Control of Chemical Processes
July 14-17, 2024. Toronto, Canada

Design of a Database-Driven PID Controller using the Estimated Input/Output Data

Takuya Kinoshita * **Toru Yamamoto** *

* *Graduate School of Advanced Science and Engineering Hiroshima University, Higashihiroshima, Hiroshima, Japan, (e-mail: {kinoshita-takuya, okada-ryuji, yama}@hiroshima-u.ac.jp)*

Abstract: In nonlinear systems, the desired control performance is often not achieved because of changes in system characteristics. To cope with these changes, database-driven control (DD), which adjusts PID gains at each step based on data similar to current operating conditions, can be utilized. However, DD exhibits a number of problems, such as a necessity for online learning and inability to achieve the desired performance with small amounts of data. Therefore, this paper proposes methods for adjusting PID gains offline and pseudo-incrementing data from initial data. The proposed method is applicable to PID control systems over a wide range of industries. The effectiveness of the proposed method is verified through numerical example.

Keywords: database-driven controller, PID controller, offline learning

1. INTRODUCTION

Nonlinear systems are used in a wide variety of industries. However, the characteristics of these systems vary depending on operating and environmental conditions and are therefore difficult to always accurately determine. Although PID controllers are still widely used in control systems, in such nonlinear systems it is difficult to achieve the desired control performance using fixed PID gains.

In general, to determine the control parameters, it is necessary to identify the system and set appropriate parameters based on the model that is used the references [R.E. Kalman (1960); Z. Hou (2013); O. Larses (2004); F. Franco (2016)]. However, determining the model requires specialized knowledge. Furthermore, the model must be updated whenever there are changes in the operating or environmental conditions, making it a difficult problem in practice.

By contrast, data-driven control [Z. Hou (2013)], which has been actively studied in recent years, does not require system modeling and has the advantage of simplifying the control system design procedure. Among its example methods are iterative feedback tuning (IFT) [H. Hjalmarsson (1998); K. Hamamoto (2003)], virtual reference feedback tuning (VRFT) [M. C. Campi (2002)], fictitious reference iterative tuning (FRIT) [O. Kaneko (2013)], and extended FRIT (E-FRIT) [M. Kano (2010)], which have been theoretically constructed for linear systems and have been validated by experiments for their effectiveness. Database-driven PID control (DD-PID) [T. Yamamoto (2009)], which does not require system modeling and can be applied to nonlinear systems, has also been proposed. The mechanism of DD control is as follows: First, in DD control, an information vector, composed of a set of operational data and control parameters, is defined, which is stored in a database each time. The control parameters are then learned online for these parameters to follow the desired reference trajectory. Thus, DD control can cope with nonlinearity by variably adjusting control parameters each time, and its effectiveness has been experimentally confirmed.

However, two problems with DD control have been emphasized. The first is that DD control requires an enormous amount of learning time because the control parameters are learned online. The result is that the equipment must be kept in operation, which increases the associated time and personnel costs. The second problem is that control performance deteriorates when the amount of data stored in the database is small. However, increasing the size of the database requires repeated experiments, and the human and time costs associated with such experiments remain a real problem.

This paper proposes a DD scheme that addresses the aforementioned two problems. Specifically, the amount of experimental data is artificially increased offline, and PID gains are learned based on the data using the estimated response iterative tuning (ERIT) method [O. Kaneko (2018)]. In ERIT, the estimated input–output data corresponding to various control parameters can be generated using a set of closed-loop data. In this paper, the proposed scheme is referred to as “DD-ERIT.” It is also noted that the conventional ERIT method uses only two-degrees-of-freedom (2DOF) control and is not applicable to basic one-degree-of-freedom (1DOF) control, which is widely used in industries. Therefore, this paper proposes a method that is also applicable to 1DOF.

The paper is organized as follows. Section 2 provides an overview of DD-ERIT and explains the DD design method and the ERIT data generation method. Section 3 compares the numerical simulation results of the proposed DD-ERIT

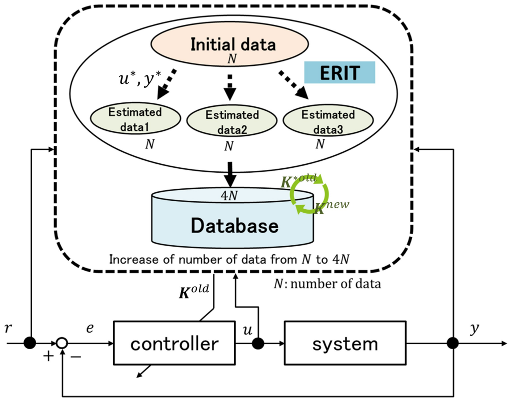

Fig. 1. Schematic figure of proposed scheme.

control with those of other methods. Finally, Section 5 presents the conclusions of the study.

2. DESIGN OF DATABASE-DRIVEN CONTROLLER USING ESTIMATED DATA

2.1 Overview of proposed scheme

Fig. 1 shows an overview of the proposed scheme. First, initial closed-loop data are acquired. In the illustrated example, the number of initial control data points is N. In DD control, the control performance generally improves when a large number of data points are obtained and the database size is increased.

In the proposed scheme, the new data are generated offline without any additional experiments. Specifically, the number of data points is increased by applying ERIT to the set of closed-loop data. In the given example, the number of data points is increased this way from N to $4N$. Input/output data corresponding to different reference models are then stored in a database. An improvement in control performance is expected from using the aforementioned scheme.

2.2 System description

The controlled system is expressed as the following discrete-time nonlinear system:

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

where $y(t)$ and $f(\cdot)$ denote the output and the nonlinear function, respectively. In this expression, $\boldsymbol{\phi}(t-1)$ is referred to as the information vector, which is defined by the following equation:

$$\begin{aligned}\boldsymbol{\phi}(t-1) := [&y(t-1), \cdots, y(t-n_y),\\ &u(t-1), \cdots, u(t-n_u)],\end{aligned} \tag{2}$$

where $u(t)$ indicates the control input, and n_y and n_u are the orders of the output and the control input, respectively.

2.3 Design of database-driven PID controller

The following PID controller is introduced herein:

$$\Delta u(t) = K_I(t)e(t) - K_P(t)\Delta y(t) - K_D(t)\Delta^2 y(t), \tag{3}$$

where $e(t)$ is the control error, defined by

$$e(t) := r(t) - y(t), \tag{4}$$

and $K_P(t)$, $K_I(t)$, and $K_D(t)$ denote the proportional gain, integral gain, and derivative gain, respectively. Additionally, Δ denotes the differencing operator $\Delta(:= 1-z^{-1})$, and $z^{-1}y(t) = y(t-1)$. The procedure of DD-PID [T. Yamamoto (2009)] is summarized in the following steps.

[STEP1] Create initial database: To use the data-driven control scheme, historical data should first be gathered. These initial historical data are obtained using the initial PID gains, which are calculated using, for example, the Zieglar & Nichols (ZN) method [J. G. Zieglar (1942)], or Chien, Hrones, & Reswick (CHR) method [K.L. Chien (1952)]. The following vector $\boldsymbol{\Phi}$ as the initial database is composed using the input/output data and PID gains:

$$\boldsymbol{\Phi}(j) := \left[\bar{\boldsymbol{\phi}}(j), \boldsymbol{K}(j)\right], \qquad (j = 1, 2, \cdots, N) \tag{5}$$

where N denotes the initial number of data points. Meanwhile, $\bar{\boldsymbol{\phi}}(j)$ and $\boldsymbol{K}(j)$ are defined as follows:

$$\begin{aligned}\bar{\boldsymbol{\phi}}(t) := [&r(t+1), r(t), y(t), \cdots, y(t-n_y+1)\\ &u(t-1), \cdots, u(t-n_u+1)],\end{aligned} \tag{6}$$

$$\boldsymbol{K}(t) := [K_P(t), K_I(t), K_D(t)]\,. \tag{7}$$

Here, the relation $\boldsymbol{K}(1) = \boldsymbol{K}(2) = \cdots = \boldsymbol{K}(N)$ is obtained using the initial database because the initial PID gains are all the same.

In a DD-control system, the initial database is composed using the information vector defined by Eq. (6). The information vector $\bar{\boldsymbol{\phi}}(t)$, which represents the current state of the system, is called the "query." In DD control, the neighbors that are similar to the query are extracted from the database, and PID gains are calculated based on the neighbors.

[STEP2] Calculate distance and select neighbors: In DD-PID, it is necessary to extract data similar to the query (neighborhood data) from the database to calculate the PID gains. At present, conventional methods [T. Yamamoto (2009); S. Wakitani (2019)] calculate the distance d between the query and each data point using the weighted $\mathcal{L}_1$ norm, and then extract the neighborhood data from the database, i.e., the data in the database are sorted by their d values, and n_k neighbors are selected from the smallest ones. However, there is no clear way to determine the number n_k of neighboring data points, and the user has to derive it by trial and error.

In this study, PID gains are calculated based on the similarity [A. Gramacki (2018)] shown in Eq. (8). The similarity $S(\bar{\boldsymbol{\phi}}(t), \bar{\boldsymbol{\phi}}(j))$ between the query $\bar{\boldsymbol{\phi}}(t)$ and the j-th information vector $\bar{\boldsymbol{\phi}}(j)$ stored in the database is defined by the following equation:

$$\begin{aligned}&S(\bar{\boldsymbol{\phi}}(t), \bar{\boldsymbol{\phi}}(j))\\ &= \prod_{i=1}^{n_y+n_u+1} \frac{1}{\sqrt{2\pi}h_i} \exp\left\{-\frac{1}{2}\left(\frac{\bar{\boldsymbol{\phi}}_i(t) - \bar{\boldsymbol{\phi}}_i(j)}{h_i}\right)^2\right\},\end{aligned} \tag{8}$$

where h_i is the bandwidth, and $\bar{\boldsymbol{\phi}}_i(j)$ denotes the i-th element of the j-th information vector. As described earlier, the similarity $S(\bar{\boldsymbol{\phi}}(t), \bar{\boldsymbol{\phi}}(j))$ can be computed directly using the information vectors stored in the database.

There are several methods for determining h_i, but this study uses the plugin method [C. R. Loader (1952)]. The standard deviation σ_i is defined as follows, using the number N of data points:

$$h_i = \frac{1.06\sigma_i}{N^{\frac{1}{5}}}, \tag{9}$$

$$\sigma_i = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(\bar{\boldsymbol{\phi}}_{\boldsymbol{i}}(n) - \mu_i)^2}, \tag{10}$$

where μ_i is the mean value of $\bar{\boldsymbol{\phi}}_i$.

In Eq. (8), the highest similarity occurs when the information vectors are exactly the same, and the similarity in this case is $S(\bar{\boldsymbol{\phi}}(t), \bar{\boldsymbol{\phi}}(t)) = \prod_{i=1}^{n_y+n_u+1} \frac{1}{\sqrt{2\pi}h_i}$. On the other hand, if they are not similar, then $S(\bar{\boldsymbol{\phi}}(t), \bar{\boldsymbol{\phi}}(j)) \to 0$. Therefore, in this study, the data $\bar{\boldsymbol{\phi}}(j)$ that satisfy the following equation are identified as the neighborhood data.

$$\begin{aligned} S(\bar{\boldsymbol{\phi}}(t), \bar{\boldsymbol{\phi}}(j)) &\geq T_{th} \cdot S(\bar{\boldsymbol{\phi}}(t), \bar{\boldsymbol{\phi}}(t)) \\ &\geq T_{th} \cdot \prod_{i=1}^{n_y+n_u+1} \frac{1}{\sqrt{2\pi}h_i}, \\ &(t, j = 1, 2, \ldots, N) \end{aligned} \tag{11}$$

where T_{th} is the threshold value specified by the user, with the range $0 \leq T_{th} \leq 1$. For example, setting $T_{th} = 0.99$ signifies that "information vector $\bar{\boldsymbol{\phi}}(t)$ is 99% similar to $\bar{\boldsymbol{\phi}}(j)$." It is easy for a user to set the threshold T_{th} based on the similarity. Additionally, only the data $\bar{\boldsymbol{\phi}}(j)$ satisfying Eq. (11) need to be extracted, which eliminates the need for sorting and reduces the computational cost.

[STEP3] Compute PID gains: Based on the n_k-pieces neighbors obtained in [STEP2], the PID gains are computed using the following linearly weighted average (LWA):

$$\boldsymbol{K}^{old}(t) = \sum_{i=1}^{n_k} w_i \boldsymbol{K}(i), \qquad \sum_{i=1}^{n_k} w_i = 1, \tag{12}$$

where w_i is the weight corresponding to $\boldsymbol{K}(i)$ for the selected neighbors. It is calculated using

$$w_i = \frac{S(\bar{\boldsymbol{\phi}}(t), \bar{\boldsymbol{\phi}}(i))}{\sum_{j=1}^{n_k} S(\bar{\boldsymbol{\phi}}(t), \bar{\boldsymbol{\phi}}(j))}. \tag{13}$$

The value of $\boldsymbol{K}^{old}(t)$ at each time can be calculated using the aforementioned procedure. Note that $\boldsymbol{K}^{old}(t)$ is calculated based on the initial data. If the size of the database is increased, it is easy to calculate better PID gains. Therefore, an approach to increasing the dataset is proposed in the next section. In particular, a learning method for PID gains is described in section 2.5.

2.4 ERIT

Fig. 2 shows a 2DOF control system, which will be used in this study to apply ERIT. In the diagram, $G_m(z^{-1})$, $C_{FB}(z^{-1})$, and $C_{FF}(z^{-1})$ indicate the transfer function of the reference model, the feedback controller, and the feedforward controller, respectively. By contrast, industrial processes widely utilize 1DOF control systems, where $C_{FF} = 0$. In this study, the designed system is a 1DOF control system, whereas that for data generation is a 2DOF control system. The following $y_0(t)$ is the initial output of the 1DOF control system:

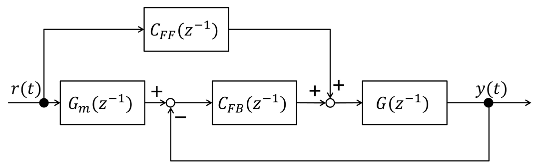

Fig. 2. Two-degree-of-freedom control system.

$$y_0(t) = \frac{G(z^{-1})C_{FB}(z^{-1})G_m(z^{-1})}{1 + G(z^{-1})C_{FB}(z^{-1})} r(t). \tag{14}$$

The following output $y(t)$ of the 2DOF control system is used to generate the data offline:

$$y(t) = \frac{G(z^{-1})C_{FF}(z^{-1}) + G(z^{-1})C_{FB}(z^{-1})G_m(z^{-1})}{1 + G(z^{-1})C_{FB}(z^{-1})} r(t). \tag{15}$$

When $r(t)$ in Eq. (14) is substituted into Eq. (15), the following relationship is obtained using the initial output $y_0(t)$:

$$y(t) = \frac{C_{FF}(z^{-1})}{C_{FB}(z^{-1})G_m(z^{-1})} y_0(t) + y_0(t). \tag{16}$$

The input $u(t)$ can also be obtained using the following equation with the initial input $u_0(t)$:

$$u(t) = \frac{C_{FF}(z^{-1})}{C_{FB}(z^{-1})G_m(z^{-1})} u_0(t) + u_0(t). \tag{17}$$

Here, a feedforward controller $C_{FF}^*(z^{-1})$ that can achieve the desired response $y^*(t)$ is introduced. Eq. (18) is derived by replacing $C_{FF}(z^{-1})$ with $C_{FF}^*(z^{-1})$ in Eq. (16). Similarly, the input $u^*(t)$ can also be obtained using Eq. (19) with $C_{FF}^*(z^{-1})$:

$$y^*(t) = \frac{C_{FF}^*(z^{-1})}{C_{FB}(z^{-1})G_m(z^{-1})} y_0(t) + y_0(t), \tag{18}$$

$$u^*(t) = \frac{C_{FF}^*(z^{-1})}{C_{FB}(z^{-1})G_m(z^{-1})} u_0(t) + u_0(t). \tag{19}$$

It is possible to estimate the desired response $y^*(t)$ and $u^*(t)$ using the initial closed-loop data $y_0(t)$ and $u_0(t)$ without any system model $G(z^{-1})$. This implies that the input/output signals can be estimated offline when $C_{FF}^*(z^{-1})$ is applied.

The desired response $y^*(t)$ can be expressed as follows using the reference model $G_m(z^{-1})$:

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

To optimize $C_{FF}(z^{-1})$, the following objective function J_1 is introduced:

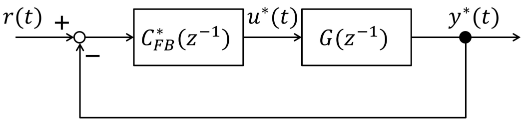

Fig. 3. One-degree-of-freedom control system.

$$J_1 = \sum_{t=0}^{N-1} \left\{ G_m(z^{-1})r(t) - \frac{C_{FF}(z^{-1})}{C_{FB}(z^{-1})G_m(z^{-1})} y_0(t) - y_0(t) \right\}^2 . \quad (21)$$

In other words, $C_{FF}(z^{-1})$ can be optimized by minimizing J_1. Furthermore, the input/output signals can be estimated offline by using Eqs. (16) and (17). Therefore, it is possible to increase the database size by storing estimated data.

Note that the reference model $G_m(z^{-1})$ is given by the following equation [T. Yamamoto (2004)]:

$$G_m(z^{-1}) = \frac{z^{-1}T(1)}{T(z^{-1})}, \quad (22)$$

where $T(z^{-1})$ is the characteristic polynomial of the reference model and is expressed by the following equation:

$$T(z^{-1}) := 1 + t_1 z^{-1} + t_2 z^{-2} \quad (23)$$

$$\left. \begin{array}{l} t_1 = -2\exp\left(-\dfrac{\rho}{2\mu}\right)\cos\left(\dfrac{\sqrt{4\mu-1}}{2\mu}\rho\right) \\ t_2 = \exp\left(-\dfrac{\rho}{\mu}\right) \\ \rho := T_s/\sigma \\ \mu := 0.25(1-\delta) + 0.51\delta \end{array} \right\}, \quad (24)$$

where T_s is the sampling time; σ denotes the rise time; μ is the damping coefficient and is adjusted by changing δ; and $\delta = 0$ and $\delta = 1$ are the responses of the Binominal and Butterworth models, respectively.

ERIT is a method that is applicable only to 2DOF control systems, and thus, it cannot estimate $y^*(t)$ and $u^*(t)$ in 1DOF control systems. However, it would better if the aforementioned scheme were applicable to 1DOF control systems because 1DOF systems are widely utilized in industries. In the next section, a method for calculating the control parameters of a 1DOF control system using the data estimated by ERIT is discussed.

2.5 Database-driven 1DOF controller using estimated data

In this part of the study, the 1DOF controller, shown in Fig. 3, is designed using the $u^*(t)$ and $y^*(t)$ derived earlier via ERIT. In the diagram, $C^*_{FB}(z^{-1})$ is a feedback controller that achieves $u^*(t)$ and $y^*(t)$. First, the estimated error $e^*(t)$ is calculated using

$$e^*(t) = r(t) - y^*(t). \quad (25)$$

Based on Eq. (3), the PID control law for the 1DOF control system is expressed as

$$\Delta u^*(t) = K^*_I(t)e^*(t) - K^*_P(t)\Delta y^*(t) - K^*_D(t)\Delta^2 y^*(t) \quad (26)$$

where $K^*_P(t)$, $K^*_I(t)$, and $K^*_D(t)$ denote the proportional gain, integral gain, and derivative gain in $C^*_{FB}(z^{-1})$, respectively.

The following is the objective function J_2 for the purpose of optimizing $C^*_{FB}(z^{-1})$ in Eq. (26):

$$J_2(t) = \frac{1}{2}\left[u^*(t) - \left\{\frac{1}{\Delta}K_I(t)e^*(t) - K_P(t)y^*(t) - K^*_D(t)\Delta y^*(t)\right\}\right]^2 . \quad (27)$$

Specifically, the PID gains are updated from $\boldsymbol{K}^{*old}(t)$ to $\boldsymbol{K}^{new}(t)$ by minimizing $J_2(t)$ using the following steepest descent method:

$$\boldsymbol{K}^{new}(t) = \boldsymbol{K}^{*old}(t) - \boldsymbol{\eta}\frac{\partial J_2(t)}{\partial \boldsymbol{K}(t)} \quad (28)$$

$$\boldsymbol{\eta} := [\eta_P, \eta_I, \eta_D] \quad (29)$$

$$\left\{ \begin{array}{ll} \dfrac{\partial J_2(t+1)}{\partial K_P(t)} & = \dfrac{\partial J_2(t+1)}{\partial \epsilon(t+1)}\dfrac{\partial \epsilon(t+1)}{\partial K_P(t)} \\ & = \epsilon(t+1)\Delta y^*(t) \\ \dfrac{\partial J_2(t+1)}{\partial K_I(t)} & = \dfrac{\partial J_2(t+1)}{\partial \epsilon(t+1)}\dfrac{\partial \epsilon(t+1)}{\partial K_I(t)} \\ & = -\epsilon(t+1)e^*(t) \\ \dfrac{\partial J_2(t+1)}{\partial K_D(t)} & = \dfrac{\partial J_2(t+1)}{\partial \epsilon(t+1)}\dfrac{\partial \epsilon(t+1)}{\partial K_D(t)} \\ & = \epsilon(t+1)\Delta^2 y^*(t) \end{array} \right. \quad (30)$$

$$\epsilon(t) = u^*(t) - \left\{\frac{1}{\Delta}K_I(t)e^*(t) - K_P(t)y^*(t) - K^*_D(t)\Delta y^*(t)\right\}, \quad (31)$$

where $\boldsymbol{\eta}$ is the learning rate.

Here, $\boldsymbol{K}^{*old}(t)$ is obtained based on the similarity $S(\bar{\boldsymbol{\phi}}^*(t), \bar{\boldsymbol{\phi}}^*(j))$ using the following estimated data vector $\bar{\boldsymbol{\phi}}^*(t)$:

$$\bar{\boldsymbol{\phi}}^*(t) := [r(t+1), r(t), y^*(t), \cdots, y^*(t-n_y+1) \\ u^*(t-1), \cdots, u^*(t-n_u+1)]. \quad (32)$$

Based on its calculation using Eq. (32), $\boldsymbol{K}^{*old}(t)$ is obtained via the same approach as that used in [STEP3]. Finally, by storing $\boldsymbol{K}^{new}(t)$, the database becomes smarter to follow the desired response.

3. NUMERICAL EXAMPLE

3.1 Controlled plant and parameter setting

The controlled plant follows a Hammerstein model. The static characteristics of this model are shown in Fig. 4.

$$\left. \begin{array}{rl} y(t) & = 0.6y(t-1) - 0.1y(t-2) \\ & \quad +1.2x(t-1) - 0.1x(t-2) \\ x(t) & = 1.5u(t) - 1.5u^2(t) + 0.5u^3(t) \end{array} \right\} \quad (33)$$

The reference signal at each time is set as follows:

$$r(t) = \left\{ \begin{array}{l} 0.5 \ (0 < t \le 50) \\ 1.0 \ (50 < t \le 100) \\ 2.0 \ (100 < t \le 150) \\ 1.5 \ (150 < t \le 200) \end{array} \right. . \quad (34)$$

The polynomial of the reference model $T(z^{-1})$ is set as follows:

$$T(z^{-1}) = 1 - 0.271z^{-1} + 0.0183z^{-2}. \quad (35)$$

The user-specified parameters are summarized in Table 1.

3.2 Control results

In the proposed scheme, the initial database required is obtained using the initial PID gains. The PID gains are calculated via the CHR method [K.L. Chien (1952)] and are expressed as follows:

$$K_P = 0.243, K_I = 0.113, K_D = 0.061. \quad (36)$$

Fig. 5 shows the control results obtained via the fixed PID control in Eq. (36) and DD-FRIT [S. Wakitani (2019)]. DD-FRIT, which is an integration of DD and FRIT, is similar to DD-ERIT in that it can also learn control parameters offline. However, unlike DD-ERIT, DD-FRIT cannot generate estimated data offline. This numerical example provides a comparison of the control results for when estimated data are generated and for when they are not.

In Fig. 5, $y_0(t), u_0(t)$ indicate the initial control result, whereas $y(t), u(t)$ denote the control result for DD-FRIT. Based on a comparison with other reference responses $y^*(t)$, it can be confirmed that the rise performance is particularly poor between 100 [step] and 130 [step], and that $y_0(t)$ does not follow the desired response $y^*(t)$. By contrast, $y(t)$ follows the desired response $y^*(t)$ except for in the period between 100 [step] and 130 [step]. However, there is a large oscillation in the period of the reference signal $r = 2$. This is because the PID gains were calculated to be high as a result of offline learning by FRIT. It can also be inferred that appropriate parameters cannot be calculated from the neighborhood data. This can be observed from the trajectories of the PID gains of DD-FRIT shown in Fig. 6. Herein, K_I increases between 100 [step] and 150 [step], indicating that the output $y(t)$ is oscillating.

Subsequently, the control result and trajectories of PID gains for DD-ERIT are shown in Figs. 7 and 8, respectively. In this simulation, the nearest neighbor data are selected using a similarity threshold of $T_{th} = 95\%$. Compared with that in Fig. 5, the output $y(t)$ in Fig. 7 follows $y^*(t)$ well for all reference signals. From the trajectories of PID gains for DD-ERIT shown in Fig. 8, it is confirmed that the high gain of K_I is suppressed compared to that observed in Fig. 6. This indicates that the DD-ERIT scheme can improve control performance by increasing the amount of data in advance using Eqs. (16) and (17).

4. CONCLUSION

In this paper, a database driven-control method that uses estimated data (DD-ERIT) is proposed. The proposed scheme does not require system modeling and is applicable to nonlinear systems.

In the conventional method, the control parameters are learned online, which requires a large amount of time and manpower. By comparison, the proposed scheme has the following advantages:

Table 1. User-specified parameters in numerical example.

Orders of information vector	$n_y = 2$ $n_u = 2$
Learning rates	$\eta_P = 0.10$ $\eta_I = 0.10$ $\eta_D = 0.10$
Initial number of data points	$N(0) = 200$
Rise time	$\sigma = 1.0$
Damping coefficient	$\delta = 0.0$

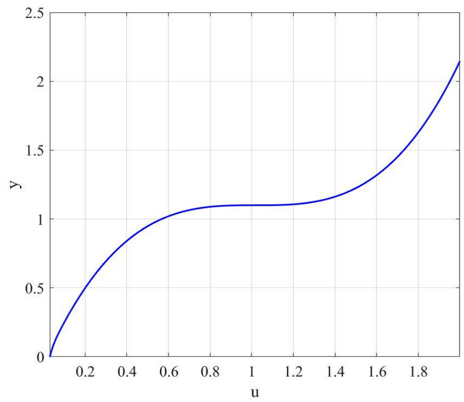

Fig. 4. Static properties of Hammerstein model.

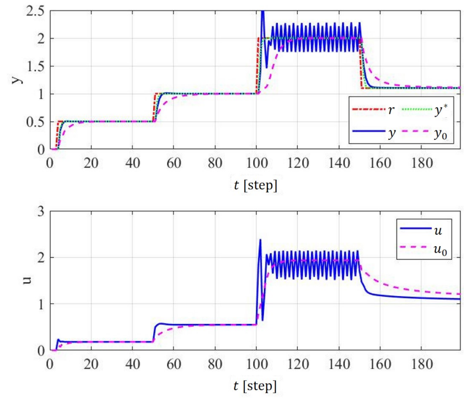

Fig. 5. Control result for DD-FRIT.

(i) The control parameters can be learned offline.
(ii) The experimental data can be artificially increased offline.

The aforementioned features are useful for solving implementation problems in industries.

In recent years, DX and cyber-physical systems have been developed. The proposed method has a very high affinity with these cyber-physical systems because it is a database-based method. Therefore, through the offline analysis and learning of data obtained in the field, the desired control performance (including energy saving) can be immediately reflected in the system.

REFERENCES

Kalman RE. A new approach to linear filtering and prediction problems. J Basic Eng 1960;82(1):34–45. https://doi.org/10.1115/1.3662552.

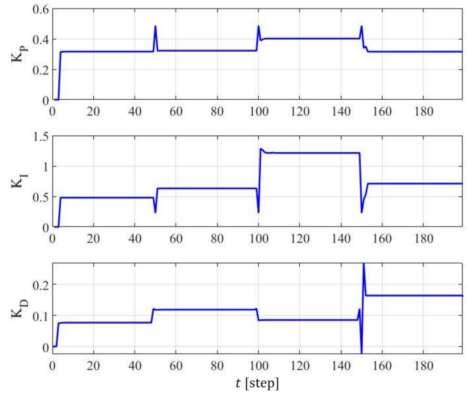

Fig. 6. Trajectories of PID gains corresponding to Fig. 5.

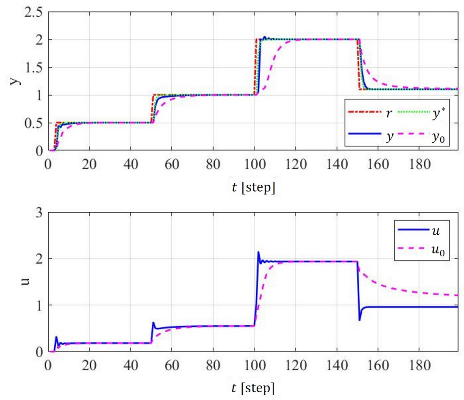

Fig. 7. Control result for DD-ERIT.

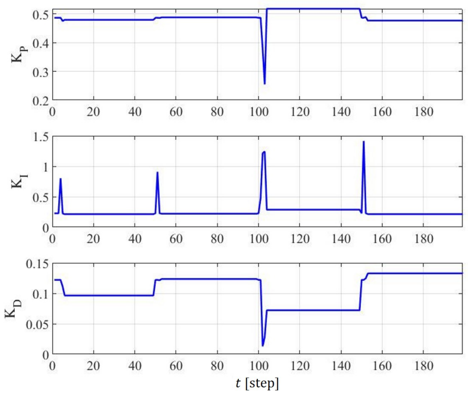

Fig. 8. Trajectories of PID gains corresponding to Fig. 7.

Hou ZS, Wang Z. From model-based control to data-driven control: survey, classification and perspective. Inf Sci 2013;235:3–35. https://doi.org/10.1016/j.ins.2012.07.014.

Larses O, Adamsson N. Drivers for model based development of mechatronic systems. In: Proceedings of DESIGN 2004, the 8th International Design Conference. Scotland: The Design Society; 2004, p. 865–70.

Franco F, Stella GD, Neme JH, Santos MMD, Stevan SL, da Rosa JNH. Teaching model-in-the loop: a case study for controller of distributed dashboard in a road vehicle. In: Proceedings of the 2016 IEEE 25th International Symposium on Industrial Electronics (ISIE). Piscataway, NJ: IEEE; 2016, p. 863–8. https://doi.org/10.1109/ISIE.2016.7745003.

Hjalmarsson H, Gevers M, Gunnarsson S, Lequi O. Iterative feedback tuning: theory and applications. IEEE Control Syst 1998;18(4):26–41. https://doi.org/10.1109/37.710876.

Hamamoto K, Fukuda T, Sugie T. Iterative feedback tuning of controllers for a two-mass-spring system with friction. Control Eng Pract 2003;11(9):1061–8. https://doi.org/10.1016/S0967-0661(02)00229-0.

Campi MC, Lecchini A, Savaresi SM. Virtual reference feedback tuning: a direct method for design of feedback controllers. Automatica 2002;38(8):1337–46. https://doi.org/10.1016/S0005-1098(02)00032-8.

Kaneko O. Data-driven controller tuning: FRIT Approach. In: Proceedings of the 11th IFAC International Workshop on Adaptation and Learning in Control and Signal Processing. Amsterdam: Elsevier; 2013, p. 326–36. https://doi.org/10.3182/20130703-3-FR-4038.00122.

Kano M, Ogawa M. The state of the art in chemical process control in Japan: good practice and questionnaire survey. J Process Control 2010;20(9):969–82. https://doi.org/10.1016/j.jprocont.2010.06.013.

Yamamoto T, Takao K, Yamada T. Design of a data-driven PID controller. IEEE Trans Control Syst Technol 2009;17(1):29–39. https://doi.org/10.1109/TCST.2008.921808.

Kaneko O, Nakamura T, Ikezaki T. A new approach to update of feedfoward controller in the two-degree-of-freedom control system — a proposal of estimated response iterative tuning (ERIT) [in Japanese]. Trans Soc Instrum Control Eng 2018;54(12):857–64. https://doi.org/10.9746/sicetr.54.857.

Zieglar JG, Nichols NB. Optimum settings for automatic controllers. Trans Am Soc Mech Eng 1942;64(8):759–68. https://doi.org/10.1115/1.4019264.

Chien KL, Hrones JA, Reswick JB. On the automatic control of generalized passive systems. Trans Am Soc Mech Eng 1952;74(2):175–183. https://doi.org/10.1115/1.4015724.

Wakitani S, Yamamoto T, Gopaluni B. Design and application of a database-driven PID controller with data-driven updating algorithm. Ind Eng Chem Res 2019;58(26):11419–29. https://doi.org/10.1021/acs.iecr.9b00704.

Gramacki A. Nonparametric kernel density estimation and its computational aspects. Cham, Switzerland: Springer; 2018. https://doi.org/10.1007/978-3-319-71688-6.

Loader CR. Bandwidth selection: classical or plug-in? Ann Statist 1999;27(2):415–38. https://doi.org/10.1214/aos/1018031201.

Yamamoto T, Shah SL. Design and experimental evaluation of a multivariable self-tuning PID controller. IEE Proc Control Theory Appl 2004;151(5):645–52. https://doi.org/10.1049/ip-cta:20040853.