Design and Application of a Database-Driven PID Controller with Data-Driven Updating Algorithm

Shin Wakitani,*,† Toru Yamamoto,*,† and Bhushan Gopaluni*,‡

†*Graduate School of Engineering, Hiroshima University, Hiroshima, Japan*

‡*Department of Chemical and Biological Engineering, University of British Columbia, Vancouver, BC, Canada*

E-mail: wakitani@hiroshima-u.ac.jp; yama@hiroshima-u.ac.jp; bhushan.gopaluni@ubc.ca

Abstract

This study deals with an updating algorithm for a database-driven proportional-integral-derivative (DD-PID) controller that uses a database for tuning control parameters. PID controllers are still used in many process systems including chemical processes. However, if systems exhibit nonlinearity, PID controllers with fixed PID parameters cannot achieve the desired control performance when the systems' equilibrium points are changed by setpoint changes. The DD-PID controller has been proposed to solve this problem. This controller can realize good control performance for nonlinear systems because it updates the PID parameters in its database so that the control performance around each equilibrium point has a desired characteristic. However, many experiments have to be performed for this controller to obtain a suitable database. This paper proposes a new offline database updating method called data-driven extended fictitious reference iterative tuning (DD-E-FRIT) method. In this method, the E-FRIT method, which is a direct control parameter tuning method for the linear system, is

applied to the updating rule of the DD-PID controller. The DD-E-FRIT method can update a database offline and obtain the database for realizing the desired tracking property of a closed-loop system by storing one-shot operating data into the database.

Introduction

In industrial fields, competitive manufacturing has become more serious, and the demand for products has been diversifying gradually. Moreover, production states have numerous requirements such as energy saving and improvement of quality and production efficiency. Feedback controllers are embedded in most industrial instruments, and in many scenarios, the adjustment of control parameters influences the quality of products. More than 80% of process systems use proportional-integral-derivative (PID) controllers[1–3] because of its simple structure, and because the physical meanings of the PID parameters are clear. The performance of the controllers changes dramatically depending on the combination of the set parameters. However, if the controlled objects exhibit nonlinearity, the desired control performance may not be achieved by a PID controller with fixed PID parameters when the systems' equilibrium points are changed by setpoint changes. Meanwhile, the performance of computer hardware, such as CPUs and storage functions, has drastically improved, and miniaturization and reduction of cost and power consumption of sensors have enabled the construction of mass databases. In particular, the amount of data stored is increasing rapidly today. Moreover, recently, new technologies such as big data that use large-scale databases have emerged. Currently, database technology is an essential technology in many research fields. In the field of control engineering, in order to treat nonlinear systems, databases have been applied to many control design schemes such as the just in time (JIT) method[4–6] and lazy learning.[7,8] A database-driven PID (DD-PID) controller[9] is an example of JIT controller. In the DD-PID control, the PID parameters at each equilibrium point of the system output are adaptively tuned using updated control parameters stored in a database. In the conventional DD-PID controller, an online updating algorithm was adopted for updating the control parameters in a database. However, many experiments had to be performed to obtain suitable PID parameters using this updating algorithm; hence, the controller cannot be implemented in most real process systems even if the method is effective for nonlinear systems. In order to implement the DD-PID controller, a method that can update the database in an offline manner is necessary. If detailed models or emulators are given, the conventional DD-PID controller can update the database by executing its update algorithm to a system model. However, creating a detailed nonlinear model is time consuming and involves high design cost and numerous experiments. Moreover, the obtained model may not express the characteristics of the controlled object completely. Therefore, it is important to establish an offline updating algorithm without system models. In other words, it is important to develop a direct control parameter tuning method for the DD-PID controller without a system model (data-driven tuning).

On the other hand, data-driven tuning methods for linear systems have gained much attention because they can tune control parameters in a simple manner. Typical examples of these methods are virtual reference feedback tuning (VRFT)[10–13] and fictitious reference iterative tuning (FRIT).[14–18] A data-driven self-tuning controller is also proposed.[19] One of the advantages of data-driven tuning methods is that only one-shot closed-loop data can be used to calculate control parameters. However, the main targets are linear systems, and hence, the control performance deteriorates if these methods are applied to nonlinear systems. Therefore, an offline updating algorithm for the DD-PID controller based on FRIT that is termed the data-driven fictitious reference iterative (DD-FRIT) method[20–22] has proposed. The DD-FRIT method can learn the control parameters in a database in an offline manner based on the minimization of a criterion of the FRIT method. In this method, a database can obtain suitable PID parameters only by storing one-shot experimental data given by a PID controller with fixed PID parameters. The effectiveness of the proposed method was evaluated using simulation results and experiment results. However, FRIT is a method that only based on minimization of the output response of a controlled object, thus, the method was pointed out that the criterion of FRIT is insufficient for the chemical systems which require stable operation in the closed loop.[23] To solve the above problem, an extended FRIT (E-FRIT) has been proposed by Masuda et al.[23] E-FRIT can tune control parameters considering the variance of control input by introducing a penalty term for the control input to the conventional FRIT's criterion.

Many industrial systems including chemical plants require a stable operation for a long time. Therefore, control parameter tuning considering a load of actuators is important. However, the conventional DD-FRIT method does not consider the load because the database updating was executed based on FRIT method. From the above reason, this paper proposes a more practical database updating algorithm based on E-FRIT. This paper newly derives the updating rule from the E-FRIT's criterion. The simulation demonstrated the advantage of the DD-E-FRIT method over the conventional self-tuning controller, and the conventional DD-PID controller updated by DD-FRIT method. The proposed controller was applied to a temperature control system, and the experimental results show that the proposed DD-PID controller can realize the desired control performance by using one-shot closed-loop data.

The rest of this paper is organized as follows. Section 2 presents the basic algorithm of the conventional DD-PID controller, and discusses the problem statement. Section 3 explains the E-FRIT method and the proposed DD-E-FRIT method. Section 4 elucidates the evaluation of the effectiveness of the proposed method through simulations. Section 5 presents the experimental results for the above-mentioned temperature control system obtained by applying the proposed controller. Finally, Section 6 summarizes the research findings.

Data Driven PID Controller Design[9]

This section explains the basic algorithm of the DD-PID method. Moreover, it discusses the problem of the online updating algorithm of the conventional DD-PID controller.

System Description

It is assumed that the nonlinear system is described by the following equation.

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

where t and $y(t)$ denote the step time and the system output. $f(\cdot)$ is a nonlinear function whose output is determined by a historical data vector $\boldsymbol{\phi}(t-1)$. The historical data $\boldsymbol{\phi}(t-1)$ are denoted as follows.

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

In (2), $u(t)$ is the system input, and n_y and n_u are the orders of $y(t)$ and $u(t)$, respectively.

PID Control Law

When a PID controller is applied to process systems, sometimes the derivative kick depending on the reference signals change affects the performance of the closed-loop system. This paper introduces the following velocity-type PID control law in order to avoid the derivative kick. This control law is known as the I-PD control law.

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

where $e(t)$ is the control error which is defined as follows:

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

In (3), $K_P(t)$, $K_I(t)$ and $K_D(t)$ indicate the proportional gain, the integral gain and the derivative gain, respectively. Moreover, Δ denotes the differencing operator given by $\Delta := 1 - z^{-1}$, and z^{-1} is the backward operator, which implies $z^{-1}y(t) = y(t-1)$. $r(t)$ indicates the reference signal. In the DD-PID method, these PID gains at each step are determined by utilizing a database.

Data-driven PID controller

This section explains the working principle of the DD-PID controller. In the DD-PID controller, an initial database has to be created because the controller requires a database for its function. Thus, if a database does not exist, an initial database is created by the following procedure, namely, [**STEP 1**].

[STEP 1] Generate Initial Database

Initial operating data r_0, u_0 and y_0, that is, the reference signal, the control input and the system output, are obtained by using an I-PD controller with fixed PID gains. Datasets at each step are generated by using the obtained operating data and are sequentially stored in the database. Dataset $\mathbf{\Phi}$ is defined by the following equation.

$$\mathbf{\Phi}(j) = [\bar{\boldsymbol{\phi}}^T(t_j), \boldsymbol{\theta}_{PID}^T(t_j)], \quad j = 1, 2, \ldots, \; N., \tag{5}$$

where t_j indicates the step time when the data were obtained and stored in the database, and j and N denote the index of the dataset and the total number of datasets, respectively. The dataset has two sections: $\bar{\boldsymbol{\phi}}(t_j)$ is called the information vector, and it expresses the state of the controlled object at t_j. $\boldsymbol{\theta}_{PID}(t_j)$ expresses PID gains vector applied to the controller at t_j. These vectors are given as follows:

$$\begin{aligned} \bar{\boldsymbol{\phi}}(t_j) := [&r_0(t_j+1), r_0(t_j), y_0(t_j), \ldots, y_0(t_j - n_y + 1), \\ &u_0(t_j - 1), \ldots, u_0(t_j - n_u + 1)]^T \end{aligned} \tag{6}$$

$$\boldsymbol{\theta}_{PID}(t_j) = [K_P(t_j), K_I(t_j), K_D(t_j)]^T. \tag{7}$$

After generating the initial database, PID gains at each step t while control is in process are calculated by [**STEP 2**] and [**STEP 3**].

[STEP 2] Calculate Distance and Select Neighbor Data

The distance between the query $\bar{\boldsymbol{\phi}}(t)$, which is the information vector that indicates the current system state, and an information vector $\bar{\boldsymbol{\phi}}(t_j)$ in the database is calculated by $\mathcal{L}_1$ norm with some weights.

$$d_j(\bar{\boldsymbol{\phi}}(t), \bar{\boldsymbol{\phi}}(t_j)) = \sum_{l=1}^{n_y+n_u+1} \left| \frac{\bar{\boldsymbol{\phi}}_l(t) - \bar{\boldsymbol{\phi}}_l(t_j)}{\max\limits_{m} \bar{\boldsymbol{\phi}}_l(m) - \max\limits_{m} \bar{\boldsymbol{\phi}}_l(m)} \right|, \quad j = 1, \dots, N. \tag{8}$$

In (8), $\bar{\boldsymbol{\phi}}_l(t_j)$ expresses the l-th element in the j-th dataset, and $\bar{\boldsymbol{\phi}}_l(t)$ expresses the l-th element in the query. Moreover, $\max\limits_{m} \bar{\boldsymbol{\phi}}_l(m)$ and $\min\limits_{m} \bar{\boldsymbol{\phi}}_l(m)$ indicate the maximum and minimum values of the l-th element of all the datasets in the database. In this method, the datasets in the database are sorted in the ascending order of their distance, and k-pieces of datasets with the smallest distances between them are chosen as neighbor datasets. Here k is set as per the user's discretion.

[STEP 3] Compute PID Gains

From the selected k-pieces of neighbor datasets, a suitable set of PID gains at t are computed by using the following equation.

$$\boldsymbol{\theta}_{PID}(t) = \sum_{i=1}^{k} w_i \boldsymbol{\theta}_{PID}(t_i), \qquad \sum_{i=1}^{k} w_i = 1, \tag{9}$$

where

$$w_i = \frac{\exp(-d_i)}{\sum_{i=1}^{k} \exp(-d_i)}. \tag{10}$$

The block diagram of the DD-PID controller is shown in Fig. 1. By executing [**STEP

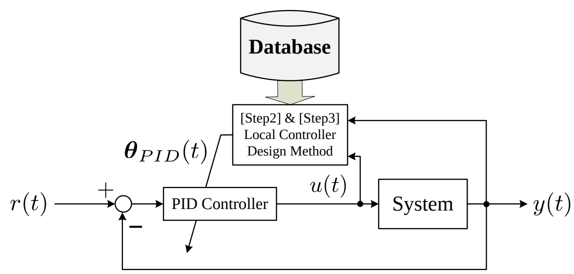

Figure 1: Block diagram of data-driven proportional-integral-derivative (DD-PID) control system.

2] and [**STEP 3**] in each sampling time interval, the PID gains are adaptively tuned if the PID gains in the database are suitably tuned in advance. However, if the result obtained by using a fixed PID controller is applied to create a database, then all PID gains included in the initial datasets may be equal. This can be expressed numerically as follows:

$$\boldsymbol{\theta}_{PID}(1) = \boldsymbol{\theta}_{PID}(2) = \cdots = \boldsymbol{\theta}_{PID}(N). \tag{11}$$

In this case, the PID gains in the initial database have to be tuned in an offline or online manner. The conventional scheme updated the database in an online manner. However, the online updating method requires many experiments to get optimal PID gains. Therefore, it is unsuitable from the viewpoint of practical use. Hence, in this research, the offline updating algorithm considers for designing an initial database by using one-shot experimental closed-loop data based on the FRIT method.

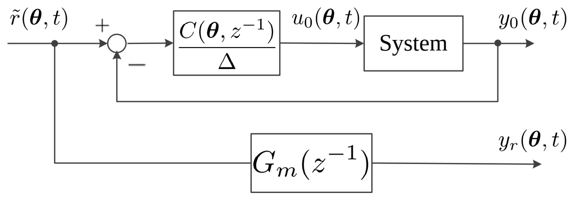

Figure 2: Block diagram of fictitious reference iterative tuning (FRIT).

Database Offline Updating Algorithm based on E-FRIT

Extended Fictitious Reference Iterative Tuning (E-FRIT)

The FRIT method[14] calculates the control parameters of a linear controller directly using a set of closed-loop data obtained by a stable controller. An optimal control parameter vector $\boldsymbol{\theta}^* = [c_0^*, c_1^*, \ldots, c_n^*]^T$ is calculated by using closed-loop data $u_0(\boldsymbol{\theta}, t)$ and $y_0(\boldsymbol{\theta}, t)$ obtained by a linear controller $C(\boldsymbol{\theta}, z^{-1})$ with an initial control parameters vector $\boldsymbol{\theta} = [c_0, c_1, \ldots, c_n]^T$. In other words, the fictitious reference signal $\tilde{r}(\boldsymbol{\theta}, t)$ is generated by the above closed-loop data and the control parameters, and the optimal control parameters vector $\boldsymbol{\theta}^* = [c_0^*, c_1^*, \ldots, c_n^*]^T$ is calculated using this signal. The block diagram of the FRIT method is shown in Fig. 2. $C(\boldsymbol{\theta}, z^{-1})/\Delta$ expresses a linear controller with an integrator. In addition, $C(\boldsymbol{\theta}, z^{-1})$ is given as the following polynomial.

$$C(\boldsymbol{\theta}, z^{-1}) = c_0 + c_1 z^{-1} + \cdots + c_n z^{-n}, \tag{12}$$

where n indicates the order of the controller. In the PID controller, the order of n equals to 2. From Fig. 2, the following equation is obtained as the I/O relationship of the controller.

$$u_0(\boldsymbol{\theta}, t) = \frac{C(\boldsymbol{\theta}, z^{-1})}{\Delta}\{\tilde{r}(\boldsymbol{\theta}, t) - y_0(\boldsymbol{\theta}, t)\}. \tag{13}$$

By rewriting (13), $\tilde{r}(\boldsymbol{\theta}, t)$ is given as follows.

$$\tilde{r}(\boldsymbol{\theta}, t) = C^{-1}(\boldsymbol{\theta}, z^{-1})\Delta u_0(\boldsymbol{\theta}, t) + y_0(\boldsymbol{\theta}, t). \tag{14}$$

In the FRIT method, a reference model $G_m(z^{-1})$ with the desired properties is designed by the user in advance, and the output from the model is defined as $y_r(t)$. Moreover this method solves the following optimization problem and derives the optimal control parameters. Thus, the controller with optimal control parameters can be obtained by a set of closed-loop data.

$$\boldsymbol{\theta}^* = \arg \min_{\boldsymbol{\theta}} J_{\text{FRIT}}(\boldsymbol{\theta}) \tag{15}$$

$$J_{\text{FRIT}}(\boldsymbol{\theta}) = \frac{1}{N}\sum_{t=1}^{N} \{y_0(\boldsymbol{\theta}, t) - y_r(\boldsymbol{\theta}, t)\}^2 \tag{16}$$

However, the method was pointed out that the criterion (16) is not enough in process systems that give emphasis to the stability of a closed loop system because the criterion only focuses on the minimization of the output error. Therefore, Masuda et al have proposed the following new criterion that is introduced a penalty term of the differential system input $\Delta\tilde{u}(t)$ to the previous criterion.[23]

$$\boldsymbol{\theta}^* = \arg \min_{\boldsymbol{\theta}} J_{\text{E-FRIT}}(\boldsymbol{\theta}) \tag{17}$$

$$J_{\text{E-FRIT}}(\boldsymbol{\theta}) = \frac{1}{N}\sum_{t=1}^{N} \left[\{y_0(\boldsymbol{\theta}, t) - y_r(\boldsymbol{\theta}, t)\}^2 + \lambda f_s \Delta\tilde{u}(\boldsymbol{\theta}, t)^2\right] \tag{18}$$

$$\Delta\tilde{u}(\boldsymbol{\theta}, t) = C(\boldsymbol{\theta})\{\tilde{r}(\boldsymbol{\theta}, t) - y_r(\boldsymbol{\theta}, t)\} \tag{19}$$

$$f_s = \sqrt{\frac{\text{Var}\left[y_r(\boldsymbol{\theta}, t) - y_0(t)\right]}{\text{Var}\left[\Delta\tilde{u}(\boldsymbol{\theta}, t)\right]}} \tag{20}$$

Where f_s is a scaling parameter. This method is called Extended-FRIT (E-FRIT) method. The original FRIT and E-FRIT methods can be applied to only linear systems. Thus, in this work, the concept of the E-FRIT method is introduced to the offline updating algorithm of the DD-PID controller. The specific algorithm of the updating method is presented in the next section.

Derivation of Updating Algorithm based on E-FRIT

In this research, the PID gains of the datasets stored in the initial database are updated using closed-loop data. In order to calculate the PID gains, k neighbor datasets around a query $\bar{\boldsymbol{\phi}}_0(t)$ is chosen by using (8). Where the query $\bar{\boldsymbol{\phi}}_0(t)$ is calculated by using closed-loop data $r_0(t)$, $y_0(t)$ and $u_0(t)$ as follows.

$$\begin{aligned}\bar{\boldsymbol{\phi}}_0(t) := [&r_0(t+1), r_0(t), y_0(t), \dots, y_0(t-n_y+1),\\ &u_0(t-1), \dots, u_0(t-n_u+1)]^T.\end{aligned} \tag{21}$$

Next, PID gains $\boldsymbol{\theta}_{PID}(t)$ are calculated by (9). Furthermore, the calculated PID gains $\boldsymbol{\theta}_{PID}(t)$ are updated by the following steepest descent method.

$$\boldsymbol{\theta}^T_{PID}(t) - \boldsymbol{\eta}\frac{\partial J(t+1)}{\partial \boldsymbol{\theta}^T_{PID}(t)}, \tag{22}$$

where $J(t)$ is the criterion of the method defined as follows:

$$J(t) = \frac{1}{2}\left\{\varepsilon(\boldsymbol{\theta}_{PID}, t)^2 + \lambda f_s \Delta\tilde{u}(\boldsymbol{\theta}_{PID}, t-1)^2\right\}, \tag{23}$$

$$\varepsilon(\boldsymbol{\theta}_{PID}, t) = y_0(t) - y_r(\boldsymbol{\theta}_{PID}, t), \tag{24}$$

$$\Delta\tilde{u}(\boldsymbol{\theta}_{PID}, t) = K_I(t)\{\tilde{r}(t) - y_r(t)\} - K_P(t)\Delta y_r(t) - K_D(t)\Delta^2 y_r(t). \tag{25}$$

From (3) and (14), $\tilde{r}(t)$ is calculated as follows.

$$\tilde{r}(t) := y_0(t) + \frac{1}{K_I(t)}\left\{\Delta u_0(t) + K_P(t)\Delta y_0(t) + K_D(t)\Delta^2 y_0(t)\right\}. \tag{26}$$

$y_r(t)$ can be calculated by the fictitious reference signal $\tilde{r}(t)$ and the reference model $G_m(z^{-1})$ as follows.

$$y_r(t) = G_m(z^{-1})\tilde{r}(t). \tag{27}$$

Moreover, $G_m(z^{-1})$ is given as follows.[9]

$$G_m(z^{-1}) = \frac{z^{-1}P(1)}{P(z^{-1})}, \tag{28}$$

$$P(z^{-1}) = 1 + p_1 z^{-1} + p_2 z^{-2}. \tag{29}$$

$P(z^{-1})$ is the characteristic polynomial. The coefficients of $P(z^{-1})$ are designed as follows.[24]

$$\left.\begin{aligned} p_1 &= -2\exp\left(-\tfrac{\rho}{2\mu}\right)\cos\left(\tfrac{\sqrt{4\mu-1}}{2\mu}\rho\right) \\ p_2 &= \exp\left(-\tfrac{\rho}{\mu}\right) \\ \rho &:= T_s/\sigma \\ \mu &:= 0.25(1-\delta) + 0.51\delta \end{aligned}\right\}. \tag{30}$$

In (30), T_s is the sampling interval. Further, σ denotes the rise time in which the system output attains about 60% of the final value of a step reference signal. The damping property δ is generally set within $0 \leq \delta \leq 2.0$. According to the above equations, $P(z^{-1})$ can be obtained by setting σ and δ.

The second term of the right side (22) is developed as follows.

$$
\left.
\begin{aligned}
\frac{\partial J(t+1)}{\partial K_P(t)} &= \frac{1}{2}\frac{\partial \varepsilon(t+1)}{\partial y_r(t+1)}\frac{\partial y_r(t+1)}{\partial \tilde{r}(t)}\frac{\partial \tilde{r}(t)}{\partial K_P(t)} + \frac{\lambda f_s}{2}\frac{\partial \Delta\tilde{u}(t)^2}{\partial \Delta\tilde{u}(t)}\frac{\partial \Delta\tilde{u}(t)}{\partial K_P(t)} \\
&= -\varepsilon(t+1)P(1)\frac{\Delta y_0(t)}{K_I(t)} + \lambda f_s \Delta\tilde{u}(t)\{\Delta y_0(t) - \Delta y_r(t)\} \\
\frac{\partial J(t+1)}{\partial K_I(t)} &= \frac{1}{2}\frac{\partial \varepsilon(t+1)}{\partial y_r(t+1)}\frac{\partial y_r(t+1)}{\partial \tilde{r}(t)}\frac{\partial \tilde{r}(t)}{\partial K_I(t)} + \frac{\lambda f_s}{2}\frac{\partial \Delta\tilde{u}(t)^2}{\partial \Delta\tilde{u}(t)}\frac{\partial \Delta\tilde{u}(t)}{\partial K_I(t)} \\
&= \varepsilon(t+1)P(1)\frac{x_0(t)}{K_I(t)^2} + \lambda f_s \Delta\tilde{u}(t)\{y_0(t) - y_r(t)\} \\
\frac{\partial J(t+1)}{\partial K_D(t)} &= \frac{1}{2}\frac{\partial \varepsilon(t+1)}{\partial y_r(t+1)}\frac{\partial y_r(t+1)}{\partial \tilde{r}(t)}\frac{\partial \tilde{r}(t)}{\partial K_D(t)} + \frac{\lambda f_s}{2}\frac{\partial \Delta\tilde{u}(t)^2}{\partial \Delta\tilde{u}(t)}\frac{\partial \Delta\tilde{u}(t)}{\partial K_D(t)} \\
&= -\varepsilon(t+1)P(1)\frac{\Delta^2 y_0(t)}{K_I(t)} + \lambda f_s \Delta\tilde{u}(t)\{\Delta^2 y_0(t) - \Delta^2 y_r(t)\}
\end{aligned}
\right\} . \tag{31}
$$

Where

$$
x_0(t) = \Delta u_0(t) + K_P(t)\Delta y_0(t) + K_D(t)\Delta^2 y_0(t). \tag{32}
$$

After calculating the above correction terms, the PID gains of the datasets are updated according to the following equation.

$$
\boldsymbol{\Phi}(j) \leftarrow \left[\bar{\boldsymbol{\phi}}^T(t_j), \boldsymbol{\theta}^T_{PID}(t_j) - \boldsymbol{\eta}\frac{\partial J(t+1)}{\partial \boldsymbol{\theta}^T_{PID}(t)}\right], \quad j = 1, \ \ldots, \ k. \tag{33}
$$

This procedure is executed iteratively until the amount of correction in (23) becomes sufficiently small. Eventually, the database is updated completely and the control performance of the closed-loop system is improved to get closer to the desired closed-loop property. The database updating algorithm of the proposed method is summarized as follows and a block diagram is shown in Fig. 3.

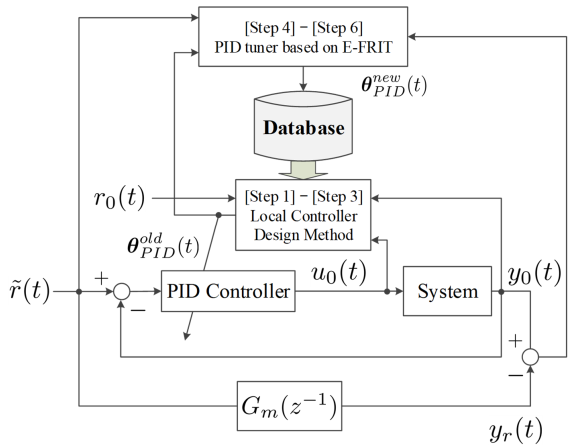

Figure 3: Brock diagram of data-driven extended fictitious reference iterative tuning (DD-E-FRIT) method.

Algorithm

The proposed algorithm is summed up as follows:

Step 1: Create a query $\bar{\boldsymbol{\phi}}_0(t)$ from the operating data, and calculate the distance between $\bar{\boldsymbol{\phi}}_0(t)$ and all of the $\bar{\boldsymbol{\phi}}(t_j)$ by using (8).

Step 2: Sort the database in ascending order of their distance, and extract k neighbor data.

Step 3: Calculate local PID gains $\boldsymbol{\theta}^T_{PID}(t)$ using the neighbor data by using (9).

Step 4: Calculate correction terms by (31).

Step 5: Update PID gains in the database by using (33).

Step 6: Repeat from Step1 to Step 5 until value of (23) at each step becomes sufficiently small.

Numerical Examples

This simulation deals with a polystyrene reactor model shown in Fig. 4. The control objective is to control the reactor temperature by manipulating the jacket temperature. A mathematical relationship between the reactor temperature and jacket temperature is given

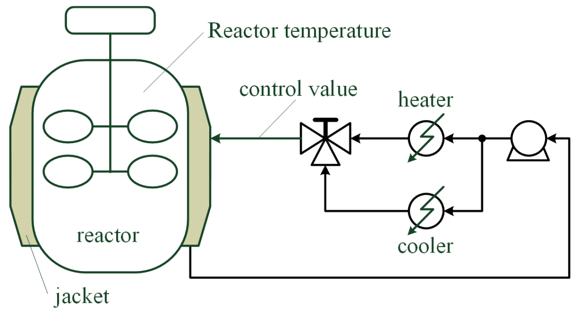

Figure 4: Schematic of polystyrene reactor.

as follows:[25] In this simulation, the sampling interval is set to $T_s = 1.0$ s.

$$y(t) = 0.804y(t-1) + 5.739 \times 10^{15} \cdot \exp\{-E_a/R(y(t-1)+273)\} + 0.148u(t-1) + \frac{\xi(t)}{\Delta}. \tag{34}$$

where $\xi(t)$ indicates the modeling error of the plant. In this simulation, the error is described as the white Gaussian noise with zero mean and variance of 0.05^2.

Firstly, a PID controller with fixed PID gains was applied. The reference signals were set as follows:

$$r(t) = \begin{cases} 60 & (0 \leq t < 300) \\ 70 & (300 \leq t < 600) \\ 85 & (600 \leq t < 900) \end{cases}. \tag{35}$$

The fixed PID gains were calculated based on the Chien, Hrones, and Reswick (CHR) method[1] as follows:

$$K_P = 9.0, \quad K_I = 0.5 \quad K_D = 1.0. \tag{36}$$

The control result obtained by fixed PID controller is as shown in Fig. 5. This result shows that the transient properties of the system output at different reference signals are not the same. It implies the difficulty of obtaining good control performance at each equilibrium

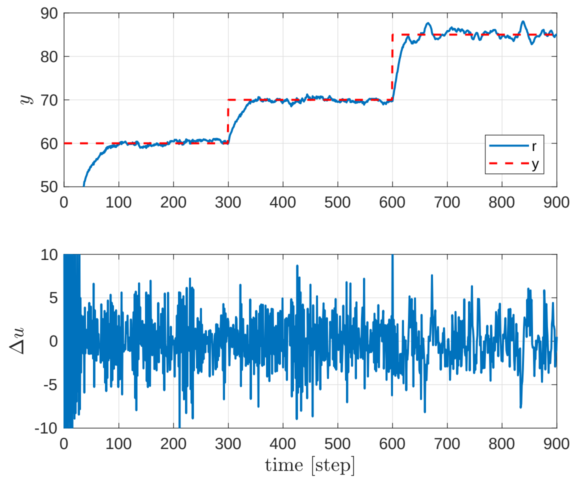

Figure 5: Control result by fixed PID controller.

point using the fixed PID controller because of the nonlinearity of the system. All the obtained closed-loop data and fixed PID gains were transformed according to the format of the datasets in eq. (5), and an initial database was created from these datasets.

The simulations validated three cases: (1) the data-driven self-tuning PID controller,[19] (2) the database-driven PID controller updated by FRIT,[20] and (3) the data-driven PID controller with the proposed method. In the simulations, the characteristic polynomial of the reference model $P(z^{-1})$ is expressed as

$$P(z^{-1}) = 1 - 1.34z^{-1} + 0.449z^{-2}. \tag{37}$$

The above polynomial was utilized in all of the cases.

Case 1: Data-driven self-tuning PID controller (conventional method)

Self-tuning control is one of the effective controller design approaches for nonlinear systems. The data-driven self-tuning PID controller is proposed and the effectiveness of the method has been verified in conventional research.[19] The conventional data-driven PID controller design approach was applied to the controlled object. The control results are shown in Fig.

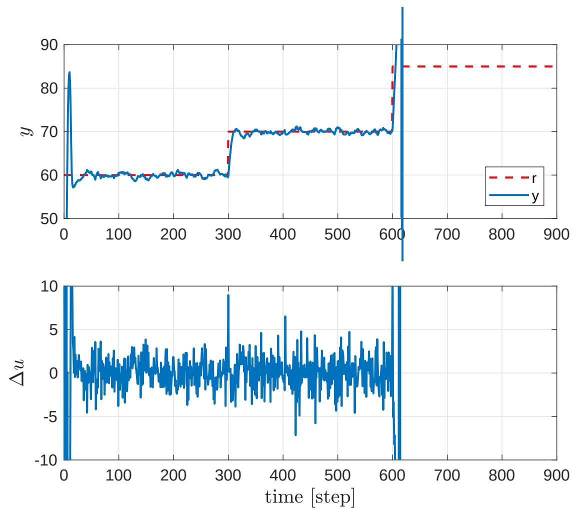

Figure 6: Control result obtained by data-driven self-tuning PID controller.

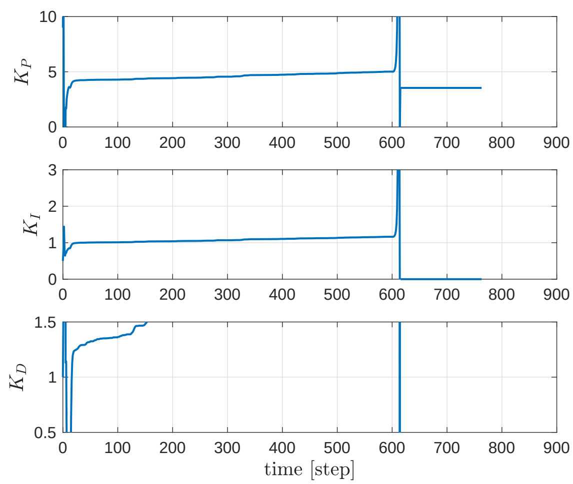

Figure 7: Trajectories of PID gains corresponding to Fig. 6.

6 and Fig. 7.

These results show that overshoot occurred at approximately 10 steps because the controller used the recursive least squares for estimating its PID gains. Moreover, although it can achieve good control performance from 50 to 600 steps, the system falls into an unstable state after 600 steps. This indicates that the self-tuning mechanism could be unable to deal with the strong nonlinearity of the plant.

Table 1: Design parameters used in simulations.

Variable	Value	Description
k	30	Number of neighbor data
n_y	3	Order of output variables
n_u	2	Order of input variables
η_P	0.1	Learning rate of P gain
η_I	0.01	Learning rate of I gain
η_D	0.01	Learning rate of D gain
σ	5.0	Rise time of reference model
δ	0	Damping property of reference model

Case 2: Database-driven PID controller with DD-FRIT method (conventional method)

The conventional DD-PID controller updated by conventional DD-FRIT method[20] was applied to the controlled object. The database generated by closed-loop data in Fig. 5 obtained by the fixed PID controller. The design parameters for online database updating are listed in Table 1.

After setting the parameters, offline updating based on FRIT was executed by utilizing the closed-loop data that was used for generating an optimal database. As stated above, this method requires a number of iterations to complete the updating. Thus, in order to confirm whether updating has been completed, the following error function was introduced and monitored:

$$J(epoch) = \frac{1}{N}\sum_{t=1}^{N}\varepsilon(t)^2 \tag{38}$$

The error behaviors of offline updating are shown in Fig. 8. Note that, the learning rates $\eta_{P,I,D}$ are adjusted heuristically. In particular, the sufficiently small values are set as initial

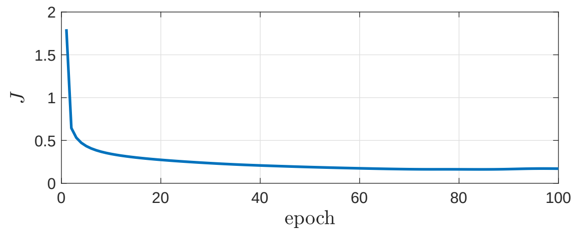

Figure 8: Error behaviors using offline updating method based on DD-FRIT.

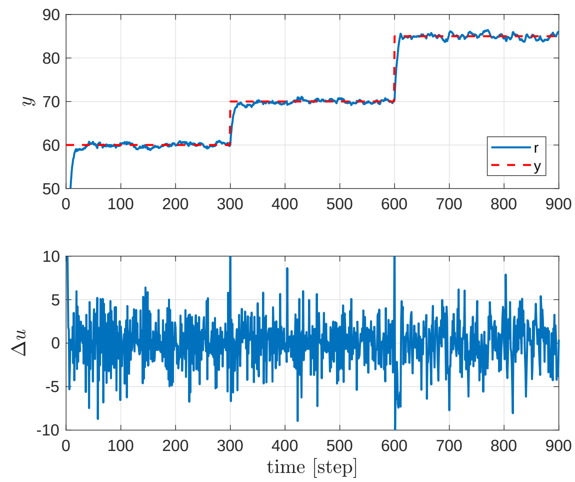

Figure 9: Control result of DD-PID controller updated by DD-FRIT.

values of the rates, and these values are gradually changed to larger values while confirming with the curve in Fig. 8, eventually, the ratios are set to the values shown in Table 1. The above strategy is also adopted in machine learning algorithms.[26] The figure shows that updating progressed significantly at approximately 10 epochs, and convergence was almost achieved at 80 epochs. In this case, the database updated by 100 epochs that is considered satisfactorily updated is adopted.

The control result obtained by the DD-PID controller is shown in Fig. 9. The trajectories of PID gains are shown in Fig. 10. From these results, the conventional DD-PID controller can obtain good control results even if the controlled object has strong nonlinearity. In this case, the variance of $\Delta u(t)$ is 10.78.

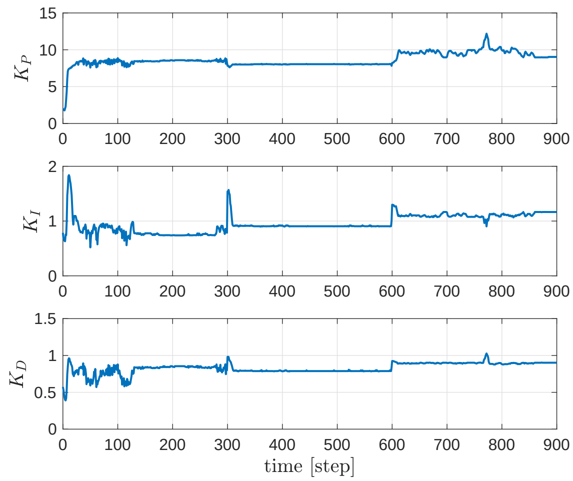

Figure 10: Trajectories of PID gains corresponding to Fig. 9.

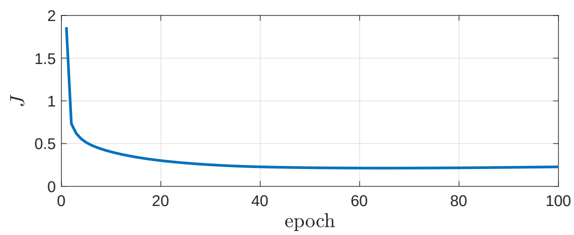

Figure 11: Criterion behaviors using proposed offline updating method.

Case 3: Database-driven PID controller with proposed method

The same initial database applied in case 2 was utilized as an initial database. The same parameters in Table 1 were also used to update the database. In this case, in order to confirm whether updating has been completed, the following error function was introduced and monitored:

$$J(epoch) = \frac{1}{N}\sum_{t=1}^{N}\left\{\varepsilon(t)^2 + \lambda f_s \Delta\tilde{u}(t)^2\right\} \tag{39}$$

Where the weight coefficient λ is set to 0.15. The criterion behaviors of offline updating are shown in Fig. 11. In this case, the database updated by 100 epochs that is considered satisfactorily updated is adopted.

The control result is shown in Fig. 12, and trajectories of the PID gains are shown in

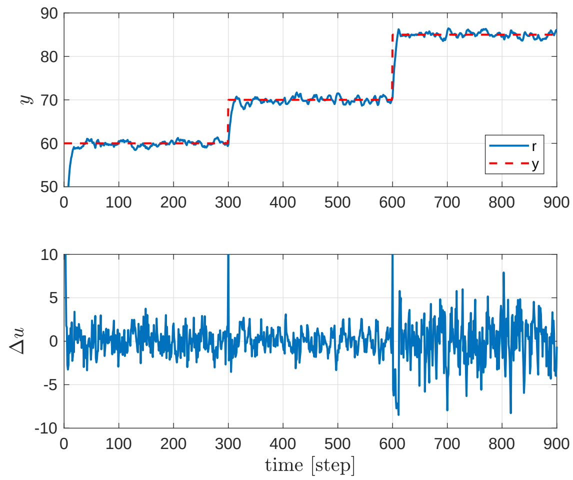

Figure 12: Control result of DD-PID controller updated by the proposed method.

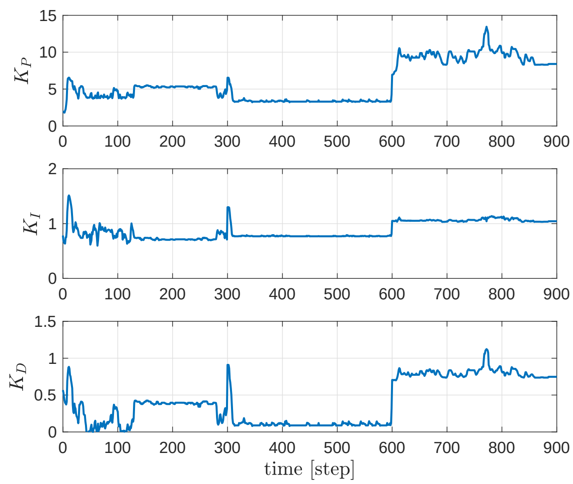

Figure 13: Trajectories of PID gains corresponding to Fig. 12.

Fig. 13. The figures show that the desired control results were obtained by tuning PID gains adaptively. From the results, the variance of the control input $\Delta u(t)$ is obviously suppressed compared with the previous DD-FRIT method. The variance of $\Delta u(t)$ is 7.24, therefore, the proposed method can suppress the variance more than 30%.

The effects of each criterion (DD-FRIT and DD-E-FRIT) on the calculated PID parameters are considered. By comparing Fig. 10 with Fig. 13, these figures show that all the PID gains are averagely smaller than the PID gains obtained by the conventional DD-FRIT. The reason can be considered that the sensitivity of the controller is tuned small values because

Figure 14: Appearance of controlled object.

of the penalty term to the difference of the input signal in eq. (18), that is λ. Therefore, the result shows that the proposed method can achieve good control performance considering actuators load.

Experimental Result

The usefulness of the proposed method is evaluated in this section. The appearance of a controlled object is shown in Fig. 14, and a schematic of the system is illustrated as Fig. 15. This section deals with temperature control of drained water $y(t)$ [°C] by manipulating the valve position for hot water $u(t)$ [%] (the opening ratio is between 0 and 100%). The sampling interval of the control systems is set to $T_s = 10$ s.

To obtain the initial closed-loop data, a PID controller with fixed PID gains was applied. The reference values were set as follows:

$$r(t) = \begin{cases} 30 & (0 \le t < 500) \\ 40 & (500 \le t < 1000) \end{cases} . \tag{40}$$

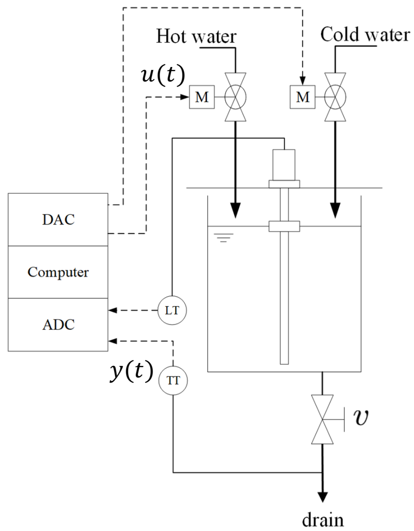

Figure 15: Schematic of controlled object.

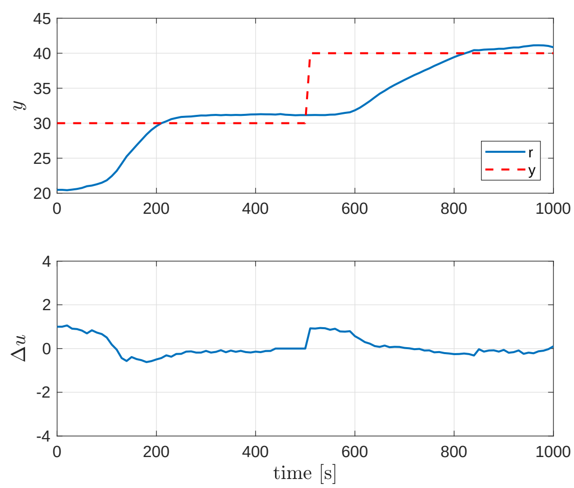

Figure 16: Control result by fixed PID controller.

The initial PID parameters determined by CHR method were set as follows:

$$K_P = 1.42, \quad K_I = 0.143, \quad K_D = 1.40. \tag{41}$$

The control results obtained by the fixed PID controller are shown in Fig. 16. The temperature control of a tank system is usually treated as a linear control problem. However, the results show that the tracking properties of the output temperature are different. Thus, it

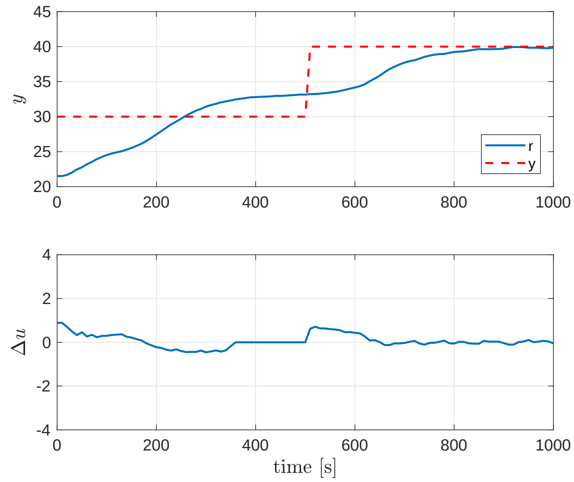

Figure 17: Control result by fixed PID controller obtained by conventional E-FRIT.

is inferred that the system may have some nonlinearity.

In this section, the conventional E-FRIT controller and the proposed DD-PID controller are applied to the controlled object. Firstly, the conventional E-FRIT method[23] is applied. The closed-loop data in Fig. 16 is utilized to calculate PID parameters. The characteristic polynomial of the reference model $P(z^{-1})$ is shown below.

$$P(z^{-1}) = 1 - 1.69z^{-1} + 0.717z^{-2}. \tag{42}$$

These coefficients are obtained by parameters $\sigma = 120$ and $\delta = 0$ in (30) set by the user, respectively. Moreover, λ in eq. (18) is set to $\lambda = 0.1$. The minimization problem given by eq. (17) is solved by using 'fmincon' command supported by MATLAB R2017b. The lower limit of the solutions are set to 0 in the function, that is the obtained PID gains are constrained to positive values. The obtained PID parameters are shown below.

$$K_P = 1.149, \quad K_I = 0.120, \quad K_D = 6.94 \times 10^{-8}. \tag{43}$$

The control result obtained by the conventional E-FRIT method is shown in 17. lthough, good control result is obtained when the reference value is set $r(t) = 40$, control performance strongly deteriorated when the reference value is $r(t) = 30$. This result indicates that although the conventional E-FRIT method works well to linear systems, the obtained parameters cannot get good control performance at each equilibrium points if the control object has nonlinearity.

Next, the DD-PID controller with propose method is applied. The initial database was generated by the obtained closed-loop data in Fig. 16. Moreover, the design parameters for offline database updating are listed in Table 2. The conditions of σ, δ, and λ are the same parameters that the conventional E-FRIT method used.

Table 2: Design parameters used in experiment.

.

Variable	Value	Description
k	10	Number of neighbor data
n_y	2	Order of output variables
n_u	5	Order of input variables
η_P	1.0×10^{-5}	Learning rate of P gain
η_I	1.0×10^{-6}	Learning rate of I gain
η_D	1.0×10^{-5}	Learning rate of D gain
σ	120	Rise time of reference model
δ	0	Damping property of reference model
λ	0.1	Weight coefficient for control input in E-FRIT

The criterion behavior calculated by (39) is shown in Fig. 18. The updating sufficiently converged at approximately 300 epochs. Thus, the database updated at 300 epochs was utilized in this experiment. The control results obtained by the proposed DD-PID controller are shown in Fig. 19 and Fig. 20. These results show that good control performance was achieved at each equilibrium point by adjusting the PID gains adaptively. The results demonstrate the usefulness of the proposed method. From these results, the conventional E-FRIT can only deal with linear systems because it determines a fixed control parameter.

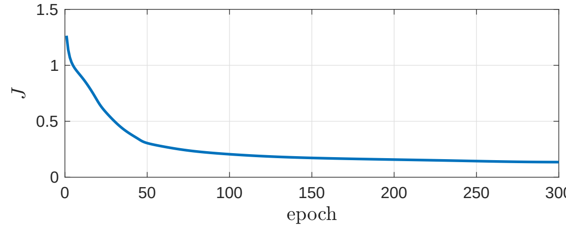

Figure 18: Criterion behaviors using proposed offline updating method.

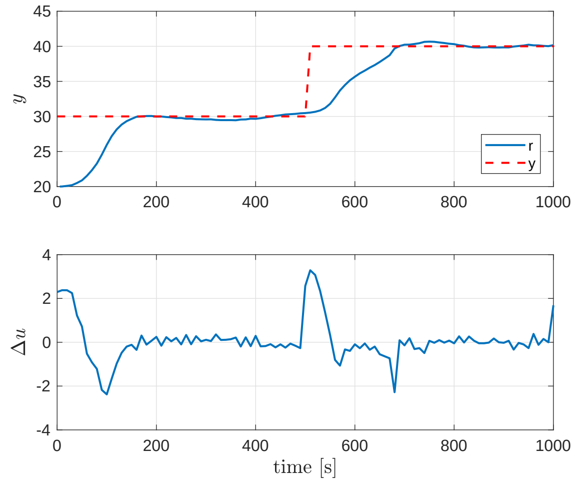

Figure 19: Control result DD-PID controller with the database updated by the proposed method.

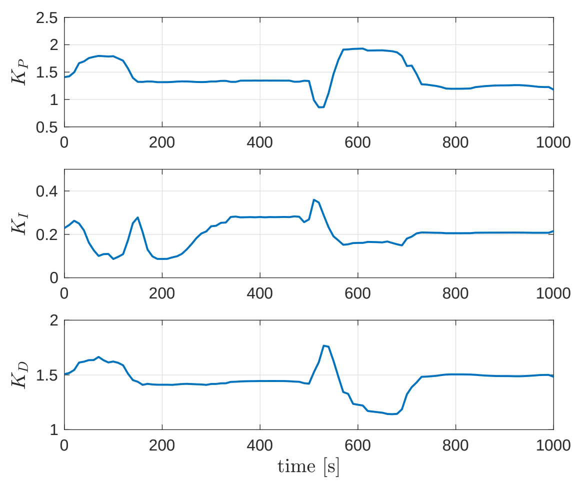

Figure 20: Trajectories of PID gains corresponding to Fig. 19.

On the other hands, the proposed method can deal with more wide systems such as nonlinear systems because it applies E-FRIT method to a piecewise linear system by introducing a concept of the local linearization of JIT modeling. Therefore, the proposed method is considered a modification of the E-FRIT method. However, by expanding to DD-PID control scheme, the proposed method has a drawback that the required design parameters are increased than the conventional E-FRIT.

Conclusion

In this work, an E-FRIT-based updating algorithm of the DD-PID controller was considered. An optimal initial database can be created using only one-shot operating data by introducing a fictitious reference signal in the E-FRIT method. Moreover, the DD-PID controller with an updated database can be obtained to achieve the desired control performance for nonlinear systems. The simulation results show that the proposed controller can achieve good control results considering the variances of control input, by using only one set of closed-loop data. Moreover, the experimental results show that the proposed method improved the control performance greatly. Thus, the application range of the proposed controller in real industrial systems is considered as wider than that of the conventional DD-PID controller. However, this method cannot be applied on systems with time delays longer than the sampling time because the method computes updated PID gains by the steepest decent method based on the one-step ahead system output. Thus, if a system has a long time delay, the updating may not be efficient. Solving this problem will be point of our future work.

Acknowledgement

This work was supported by JSPS KAKENHI Grant Number JP16H04384.

References

(1) Chien, K. L.; Hrones, J. A.; Reswick, J. B. On the automatic Control of Generalized Passice Systems. *Trans. ASME* **1952**, *74*, 175–185.

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

(3) Vilanova, R.; Visioli, A. *PID control in the Third Millennium : lessons learned and new approaches*; Springer, 2012.

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

(5) Zheng, Q.; Kimura, H. Just-in-time PID Control. *The 44th Japan Joint Automatic Control Conference* **2001**, 336–339.

(6) Nakpong, N.; Yamamoto, S. Just-In-Time Predictive Control for a Two-Wheeled Robot. *Proc. ICT-KE2012* **2012**, 95–98.

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

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

(9) Yamamoto, T.; Takao, K.; Yamada, T. Design of a Data-Driven PID Controller. *IEEE Trans. Control Systems Technology* **2009**, *17*, 29–39.

(10) A.Lecchini,; M.C.Campi,; S.M.Savaresi, Virtual Reference Feedback Tuning for Two Degree of Freedom Controllers. *International Journal of Adaptive Control and Signal Processing* **2002**, *16*, 355–371.

(11) M.C.Campi,; A.Lecchini,; S.M.Savaresi, An application of the Virtual Reference Feedback Tuning (VRFT) method to a benchmark active suspension system. *European Journal of Control* **2003**, *9*, 66–76.

(12) M.C.Campi,; A.Lecchini,; S.M.Savaresi, An application of the Virtual Reference Feedback Tuning Method to Benchmark Problem. *European Journal of Control* **2003**, *9*, 66–76.

(13) F.Previdi,; T.Schauer,; S.M.Savaresi,; K.J.Hunt, Data-Driven Control Design for Neuroprotheses : A Virtual Reference Feedback Tuning (VRFT) Approach. *IEEE Trans. on Control Systems Technology* **2004**, *12*, 176–182.

(14) Kaneko, O.; Souma, S.; Fujii, T. A fictitious reference iterative tuning (FRIT) in the two-degree of freedom control scheme and its application to closed loop system identification. *Proc. of 16th IFAC World congress (CD-ROM)* **2005**,

(15) Wakasa, Y.; Kanagawa, S.; Kanya, T.; Nishimura, Y. FRIT with dead-zone compensation and its application to ultrasonic motors. *10th IFAC International Workshop on the Adaptation and Learning in Control and Signal Processing* **2010**, 156–161.

(16) Masuda, S.; Kano, M.; Yasuda, Y. A Fictitious Reference Iterative Tuning Method with Simultaneous Delay Parameter Tuning of the Reference Model. *International Journal of Innovative Computing, Information and Control* **2010**, *6*, 1–13.

(17) Shigemasa, T.; Negishi, Y.; Baba, Y. From FRIT of a PD Feedback Loop to Process Modelling and Control System Design. *Proc. of 11th IFAC International Workshop on Adaptation and Learning in Control and Signal Processing* **2013**, 337–342.

(18) Julkananusar, A.; Nilkhamhang, I.; Vanijjirattikhan, R.; Takahashi, A. Quadrotor tuning for attitude control based on PID controller using fictitious reference iterative tuning (FRIT). *Proc. of 6th International Conference of Information and Communication Technology for Embedded Systems* **2015**, 1–5.

(19) Ashida, Y.; Wakitani, S.; Yamamoto, T. Design of an Implicit Self-Tuning PID Controller Based on the Generalized Output. *Preprints of the 20th World Congress The International Federation of Automatic Control (IFAC 2017)* **2017**, 14511–14516.

(20) Wakitani, S.; Deng, M.; Yamamoto, T. Design of a Data-Driven Controller for a Spiral Heat Exchanger. *Proc. 11th IFAC Symposium on Dynamics and Control of Process Systems, including Biosystems (DYCOPS-CAB 2016)* **2016**, 342–346.

(21) Wakitani, S.; Yamamoto, T. A learning algorithm for a Data-driven Controller based on Fictitious Reference Iterative Tuning. *Proc. of 2016 International Joint Conference on Neural Networks (IJCNN 2016)* **2016**, 4908–4913.

(22) Wakitani, S.; Yamauchi, Y.; Kinoshita, T.; Yamamoto, T.; Miyakoshi, M.; Harada, S.; Yano, Y. Design of a Vehicle Driver Model Based on Database-Driven Control Approach. *Proc. 2018 IEEE Conference on Control Technology and Applications (CCTA2018)* **2018**, 840–845.

(23) Kano, M.; Tasaka, K.; Ogawa, M.; Takinami, A.; Takahashi, S.; Yoshii, S. Extended Fictitious Reference Iterative Tuning and Its Application to Chemical Processes. *Proc. of the 2011 4th International Symposium on Advanced Control of Industrial Processes (ADCONIP)* **2011**, 379–384.

(24) Yamamoto, T.; Shah, S. L. Design and Experimental Evaluation of a Multivariable Self-Tuning PID Controller. *IEE Proc. of Control Theory and Applications* **2004**, *151*, 645–652.

(25) Nakanishi, E.; Hanakuma, Y. *Fundamentals and practice of process control (in Japanese)*; Asakura Publishing Co., Ltd., 1992.

(26) Smith, L. N. Cyclical learning rates for training neural networks. *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)* **2017**, 464–472.

Graphical TOC Entry