2014 IEEE Conference on Control Applications (CCA)
Part of 2014 IEEE Multi-conference on Systems and Control
October 8-10, 2014. Antibes, France

Design and Application of a Data-Driven PID Controller

Shin Wakitani[1] and Toru Yamamoto[2]

***Abstract*— Research on database utilization has received much attention in recent years. This paper presents a data-driven proportional-integral-derivative (DD-PID) controller design method that uses a large database. In this method, a large data-set comprising input/output (I/O) data and control parameters is stored in a database in advance, and current control parameters are tuned by using the database in an online manner. The DD-PID controller design method utilizes the fictitious reference iterative tuning (FRIT) approach to learn control parameters in the database in an offline manner by utilizing initial experimental I/O data. This method is called the DD-FRIT method. In the study, the DD-FRIT method was applied to a nonlinear tank system to evaluate its effectiveness. This paper is structured as follows. First, the DD-PID controller design method is presented. Next, the offline learning algorithm that combines the design method with FRIT is explained. Finally, the results of a simulation and experiment are presented to demonstrate the effectiveness of the proposed method. These results showed that control parameters in the database were learned appropriately by the proposed method, and good control results were obtained.**

I. INTRODUCTION

Database utilization has been an area of active research in recent years. In the control technology field, control theories that utilizes a database have been proposed and evaluated for their effectiveness.

Stenman et al. proposed the just-in-time (JIT) method [1]–[3] to model the nonlinear dynamics of systems. Yamamoto et al proposed the proportional-integral-derivative (PID) parameter tuning method based on the JIT approach [3]. Yamamoto et al. proposed the data-driven PID (DD-PID) controller [4] that stores data-sets composed of input/output (I/O) data and PID parameters in a database and uses them to calculate control parameters in real time. The DD-PID method has a learning mechanism in order to learn PID parameters of a database in an online manner. However, obtaining the optimal control parameters in a database takes a long time; thus, production costs may increase because of the long experiment.

In order to resolve this problem, the authors previously proposed the data-driven fictitious reference iterative tuning (DD-FRIT) method [5]. The DD-FRIT method is an offline learning approach for the DD-PID controller and based on the fictitious reference iterative tuning (FRIT) method [6]–[11]. FRIT is an implicit parameter tuning method that tunes control parameters directly without a system model by using the initial experimental data. The method introduces a fictitious reference signal generated by I/O data to tune control parameters. In the DD-FRIT approach, the fictitious reference signal is applied to the learning method of the DD-PID controller. With this approach, the control parameters of a database can be learned in an offline manner. This drastically reduces the time required for experiments.

The effectiveness of the DD-FRIT method was evaluated in several simulations. However, its viability has not been evaluated experimentally. This paper presents the experimental result for a nonlinear liquid-level control system. This paper is composed of the following sections. Section 2 presents the DD-PID control scheme. Section 3 presents the DD-FRIT approach. Section 4 presents the simulation result for a polystyrene reactor model with strong nonlinearity. Section 5 presents the experimental result of a nonlinear liquid-level control system. Finally, the conclusion summarizes the research findings and notes some outstanding issues. The results showed that the proposed method could learn the control parameters in a database in an effective manner.

II. Design of Data Driven PID Controller

A. System Description

First, it is assumed that a nonlinear system is described as the following equation:

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

where, $y(t)$ is the system output and $f(\cdot)$ expresses a nonlinear function whose output is determined by a historical data vector $\boldsymbol{\phi}(t-1)$. The historical data $\boldsymbol{\phi}(t)$ denotes as follows:

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

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

B. PID Control Law

For the nonlinear system of (1), the following velocity type of PID control law is introduced:

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

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

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

In (3), $K_P(t)$, $K_I(t)$ and $K_D(t)$ are expressed by the proportional gain, the integral gain and the derivative gain respectively. Moreover, Δ denotes the differencing operator

[1]Shin Wakitani is with the Institute of Engineering, Division of Advanced Electrical and Electronics Engineering, Tokyo University of Agriculture and Technology, Tokyo 184-8588, JAPAN wakitani@cc.tuat.ac.jp
[2]Toru Yamamoto is with the Faculty of Engineering, Division of Electrical, Systems and Mathematical Engineering, Hiroshima 739-8527, JAPAN yama@hiroshima-u.ac.jp

given by $\Delta := 1 - z^{-1}$, and z^{-1} is the backward operator which implies $z^{-1}y(t) = y(t-1)$. $r(t)$ in (4) indicates a reference signal. In the DD-PID method, these PID gains for each time are determined by utilizing datasets in a database.

C. Design of Data-Driven PID controller

PID gains per each unit steps is calculated by executing following 2 steps ([*STEP 1*] and [*STEP 2*]). However, in principle of the method, it needs a database thus an initial database is generated at the beginning.

[STEP 0] Generate Initial Database: In order to generate an initial database, closed-loop data is taken using a fixed PID controller. From obtained the closed-loop data, a dataset per each unit steps are generated, and they are stored in a database in series. The format of the dataset is defined as follows [4]:

$$\boldsymbol{\Phi}(j) := [\bar{\boldsymbol{\phi}}(j), \boldsymbol{K}(j)], \quad j = 1, 2, \ldots, N. \tag{5}$$

where, j indicates an index of the dataset and N is database size (it is same as number of closed-loop data). Moreover, $\bar{\boldsymbol{\phi}}(t)$ and $\boldsymbol{K}(t)$ denotes as follows:

$$\begin{aligned} \bar{\boldsymbol{\phi}}(t) \;&:=\; [r(t+1), r(t), y(t), \ldots, y(t-n_y+1), \\ &\qquad u(t-1), \ldots, u(t-n_u+1)], \quad (6) \\ \boldsymbol{K}(t) \;&=\; [K_P(t), K_I(t), K_D(t)]. \quad (7) \end{aligned}$$

If a set of fixed PID gains is chosen as being typical, then all PID gains included in the initial information vectors may be equal. Expressed numerically, that is, $\boldsymbol{K}(1) = \boldsymbol{K}(2) = \cdots = \boldsymbol{K}(N)$. In the DD-PID method, PID gains of a PID controller are adjusted in an online manor by executing the next 2 steps based on the initial database.

[STEP 1] Calculate Distance and Select Neighbors: A distance between query (which is information vector that indicates current system state) $\bar{\boldsymbol{\phi}}(t)$ and information vector $\bar{\boldsymbol{\phi}}(j)$ in a database is calculated by the following $\mathcal{L}_1$ norm with some weights:

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

In (8), $\bar{\phi}_l(j)$ expresses a l-th element in the j-th dataset, and $\bar{\phi}_l(t)$ expresses a l-th element in the query at t[step]. Moreover, $\max \bar{\phi}_l(m)$ and $\min \bar{\phi}_l(m)$ indicate a maximum value and minimum value of all l-th elements of datasets in a database. In this method, k-pieces datasets with the smallest distances among them are chosen as neighbor datasets. Where, k is set by a user.

[STEP 2] Compute PID Gains: From the selected k-pieces neighbor datasets, suitable set of PID gains at t[step] are computed by the following equation:

$$\boldsymbol{K}(t) = \sum_{i=1}^{k} w_i \boldsymbol{K(i)}, \qquad \sum_{i=1}^{k} w_i = 1, \tag{9}$$

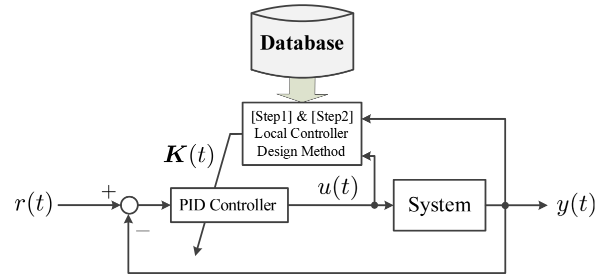

Fig. 1. Block diagram of data-driven proportional-integral-derivative (DD-PID) control system.

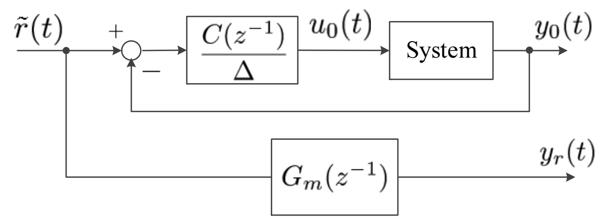

Fig. 2. Block diagram of fictitious reference iterative tuning.

where

$$w_i = \frac{\dfrac{1}{d_i}}{\displaystyle\sum_{i=1}^{k} \frac{1}{d_i}}. \tag{10}$$

By these procedures, PID gains are adaptively-tuned at every step. The block diagram of the DD-PID controller is shown in Fig. 1. However, an initial database obtained by utilizing a fixed PID controller is not able to tune PID gains suitably in operation because all PID gains in a dataset are same. Thus learning these parameters in advance is required. In Section III, the data-driven fictitious reference iterative tuning (DD-FRIT) method for learning an initial database is presented.

III. Data-Driven Fictitious Reference Iterative Tuning Method

A. Fictitious Reference Iterative Tuning (FRIT)

FRIT is a method calculating control parameters directly using a set of closed-loop data. A block diagram of the FRIT method is shown in Fig.2. In this figure, $u_0(t)$ is the control input and $y_0(t)$ is the system output of the closed-loop data and $C(z^{-1})$ is a polynomial equation of a controller. Moreover, $\tilde{r}(t)$ is the fictitious reference signal which is given by:

$$\tilde{r}(j) = C^{-1}(z^{-1})\Delta u_0(t) + y_0(t). \tag{11}$$

In the FRIT method, a criterion is defined as (12) and calculates the optimal control parameters so that minimizing the criterion:

$$J(t) = \frac{1}{2} \left\{ y_0(t) - y_r(t) \right\}^2. \tag{12}$$

Here, $y_r(t)$ is the output of $G_m(z^{-1})$ which is the desired response model given by an operator and it is denoted as follows:

$$G_m(z^{-1}) := \frac{z^{-1}P(1)}{P(z^{-1})}, \tag{13}$$

where

$$P(z^{-1}) = 1 + p_1 z^{-1} + p_2 z^{-2}, \tag{14}$$

$$\left.\begin{aligned} p_1 &= -2\exp\left(-\frac{\rho}{2\mu}\right)\cos\left(\frac{\sqrt{4\mu-1}}{2\mu}\rho\right) \\ p_2 &= \exp\left(-\frac{\rho}{\mu}\right) \\ \rho &:= T_s/\sigma \\ \mu &:= 0.25(1-\delta) + 0.51\delta \end{aligned}\right\}. \tag{15}$$

In (15), T_s is the sampling interval. Moreover, σ denotes the rise time that the system output attains about 60% of a finale value of a step reference signal. The damping property δ is generally set within $0 \leq \delta \leq 2.0$. In particular, it reflects the binomial response when $\delta = 0$ and the Butterworth model response when $\delta = 1.0$.

B. Initial Database Offline Learning Method by Utilizing FRIT

In this research, PID gains in the initial database are learned using the closed-loop data which composes the database. First, in order to calculate PID gains, the neighbor datasets around a query $\bar{\phi}_0(t)$ (which is a query at t[step] in the closed-loop data) are chosen by (8). Next, PID gains $\boldsymbol{K}^{old}(t)$ are calculated by (9). Furthermore, the calculated PID gains $\boldsymbol{K}^{old}(t)$ is learned based on the following modified law:

$$\boldsymbol{K}^{new}(t) = \boldsymbol{K}^{old}(t) - \boldsymbol{\eta}\frac{\partial J(t+1)}{\partial \boldsymbol{K}(t)} \tag{16}$$

$$\left.\begin{aligned} \frac{\partial J(t+1)}{\partial K_P(t)} &= \frac{\partial J(t+1)}{\partial y_r(t+1)}\frac{\partial y_r(t+1)}{\partial \tilde{r}(t)}\frac{\partial \tilde{r}(t)}{\partial K_P(t)} \\ &= -\frac{\varepsilon(t+1)G_m(1)\Delta y_0(t)}{K_I^{old}(t)} \\ \frac{\partial J(t+1)}{\partial K_I(t)} &= \frac{\partial J(t+1)}{\partial y_r(t+1)}\frac{\partial y_r(t+1)}{\partial \tilde{r}(t)}\frac{\partial \tilde{r}(t)}{\partial K_I(t)} \\ &= -\frac{\varepsilon(t+1)G_m(1)\Gamma(t)}{K_I^{old}(t)^2} \\ \frac{\partial J(t+1)}{\partial K_D(t)} &= \frac{\partial J(t+1)}{\partial y_r(t+1)}\frac{\partial y_r(t+1)}{\partial \tilde{r}(t)}\frac{\partial \tilde{r}(t)}{\partial K_D(t)} \\ &= -\frac{\varepsilon(t+1)G_m(1)\Delta^2 y_0(t)}{K_I^{old}(t)} \end{aligned}\right\} \tag{17}$$

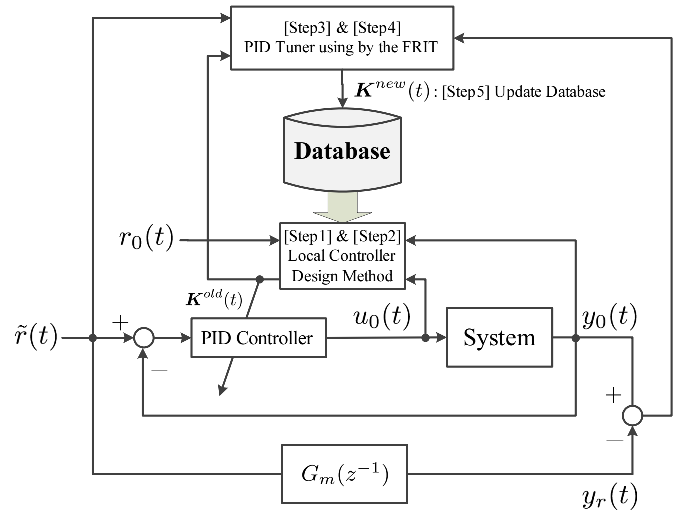

Fig. 3. Brock diagram of data-driven fictitious reference iterative tuning (DD-FRIT) method.

here,

$$\begin{aligned}\Gamma(t) &= \Delta u_0(t) + \{K_P^{old}(t) + K_D^{old}(t)\}y_0(t) \\ &\quad - \{K_P^{old}(t) + 2K_D^{old}(t)\}y_0(t-1) + K_D^{old}(t)y_0(t-2).\end{aligned} \tag{18}$$

In (16), $\boldsymbol{\eta}$ is a learning coefficient vector given by

$$\boldsymbol{\eta} = [\eta_P, \eta_I, \eta_D], \tag{19}$$

here, η_P, η_I and $,\eta_D$ indicate leaning coefficients for each amount of correction of PID gains. (16) and (17) shows that a fictitious reference signal is included into the PID modified law. Finally, the database is learned by using the learned PID gains. This procedure is iteratively executed until the amount of correction in (16) becomes small enough. The database update algorithm of the proposed method is summarized as follows and the block diagram is shown in Fig.3.

[Learning algorithm]

- Step1: Calculate the distances between $\bar{\phi}_0(t)$ and $\bar{\phi}(j)$ by (8), and k-pieces neighbor data are chosen.
- Step2: PID gains ($\boldsymbol{K}^{old}(t)$) are computed by (9).
- Step3: A fictitious reference signal at t[step] is calculated by (11).
- Step4: The correction term $\boldsymbol{\eta}\frac{\partial J(t+1)}{\partial \boldsymbol{K}(t)}$ is calculated by (17) and the learned PID gains are computed by (16).
- Step5: The initial database is learned by using learned PID gains ($\boldsymbol{K}^{new}(t)$).
- Step6: Repeat from Step1 to Step5 until the amount of correction in (16) becomes small enough.

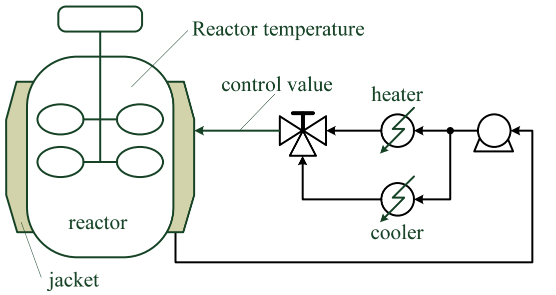

Fig. 4. The schematic of polystyrene reactor.

IV. Simulation Example

The effectiveness of the proposed method is evaluated by polystyrene reactor model. A schematic diagram of the model is shown in Fig.4. The relational expression between the jacket temperature $u(t)$ and the reaction temperature $y(t)$ is expressed as follows:

$$\begin{aligned} y(t) &= 0.804y(t-1) + 5.739 \times 10^{15} \\ &\cdot \exp\{-E_a/R(y(t-1)+273)\} \\ &+0.148u(t-1) + \xi(t). \end{aligned} \tag{20}$$

Here, $E_a = 240$, $R = 0.01986$ and $\xi(t)$ is white Gaussian noise with zero mean and variance 1.0×10^{-3}. The reference signal values for each instant of time are set as:

$$r(t) = \begin{cases} 60 \ (0 \le t < 200) \\ 70 \ (200 \le t < 400) \\ 85 \ (400 \le t < 600) \\ 75 \ (600 \le t < 800) \end{cases}. \tag{21}$$

In order to take initial data, the fixed PID controller is first applied. However, its PID gains cannot be determined by using simple tuning method such as the Ziegler-Nichols (ZN) method and the Chien, Hrones, and Reswick (CHR) method (the closed-loop system becomes unstable), thus these parameters are determined by trial and error so that a closed-loop system may be stabilized:

$$K_P = 8.0, \quad K_I = 0.2, \quad K_D = 0.1. \tag{22}$$

A control result using the fixed PID controller is shown in Fig.5. The figure shows that a rise time of the system output (especially between 0[step] and 200[step]) becomes slow because the PID gains are set with an emphasis on stability.

Next, an initial database is generated by using the I/O data in Fig.5 and its PID gains are learned by the DD-FRIT method. The setting parameters to learn the database are shown in Table.I, and the denominator polynomial of $G_m(z^{-1})$ is designed as follows:

$$P(z^{-1}) = 1 - 1.64z^{-1} + 0.670z^{-2} \tag{23}$$

Where, it is necessary to determine when the database has learned enough. Thus, in this simulation, the following

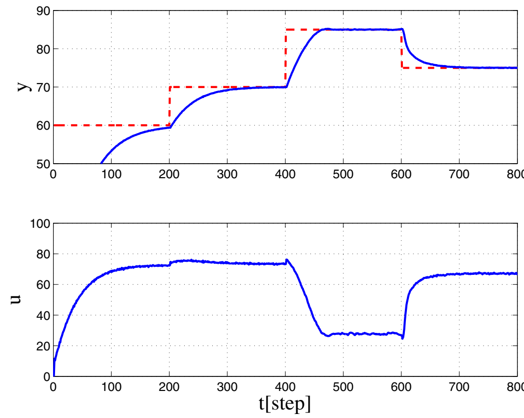

Fig. 5. Initial operating data obtained using fixed proportional-integral-derivative (PID) controller.

TABLE I
SETTING PARAMETERS OF PROPOSED METHOD.

Sampling interval	$T_s = 1.0$s
Orders of the information vector	$n_y = 2$
	$n_u = 2$
Rise time	$\sigma = 10$s
Damping property	$\delta = 0$
Number of neighbors	$k = 5$
Learning rates	$\eta_P = 0.1$
	$\eta_I = 0.05$
	$\eta_D = 0.1$

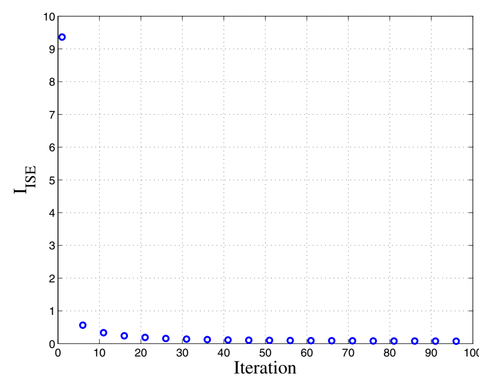

Fig. 6. Trajectory of integrated-squares-error (ISE) index at offline learning.

integrated squared error (ISE) index is defined:

$$I_{ISE} := \frac{1}{N}\sum_{t=1}^{M}\{y_0(t) - y_r(t)\}^2, \tag{24}$$

and it is considered that the learning process has completed sufficiently when the index is converged on a minimum value. The trajectory of the ISE index value is shown in Fig.6. The Fig.6 shows that the index value are rapidly converged around 10[iteration].

Finally, a control result by using the DD-PID controller is shown in Fig.7 and the trajectories of PID gains are shown in Fig.8. These figures shows that PID gains are adaptively-

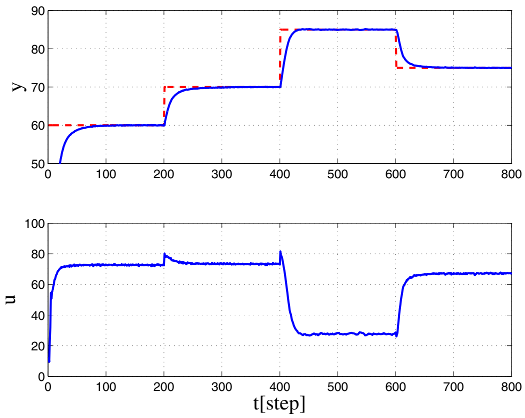

Fig. 7. Control result obtained using proposed method.

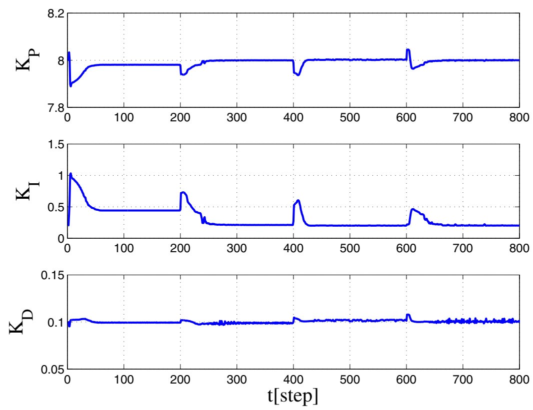

Fig. 8. Trajectories of proportional-integral-derivative (PID) gains corresponding to Fig.7.

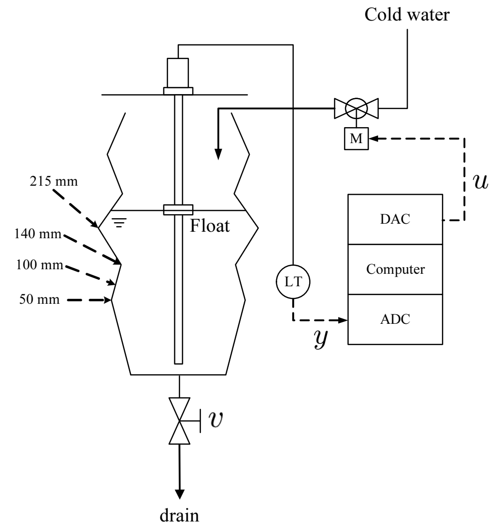

Fig. 9. Schematic of liquid-level control system.

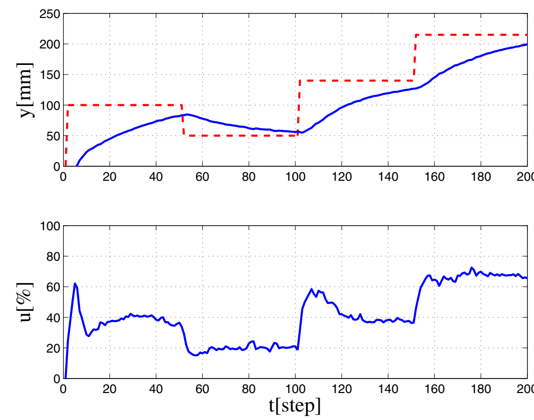

Fig. 10. Control result obtained using fixed PI controller.

tuned for each reference values and good tracking properties are obtained with maintaining of a stability.

V. Application to Liquid-Level Control System

The viability of the DD-FRIT method is evaluated by an experiment. The method is applied to the nonlinear tank system whose appearance is shown in Fig.**??**. A simple illustration of this system is shown in Fig.9. In this system, cold water enter the tank, and the liquid discharge is adjusted by manipulating the exit valve position v. In this experiment, the control objective is to regulate the water level in the tank y by manipulating the control valves u, where, the exit valve position v is stabilized.

The each reference signal value is set as:

$$r(t) = \begin{cases} 100 \text{ mm } (0 \leq t < 50) \\ 50 \text{ mm } (50 \leq t < 100) \\ 140 \text{ mm } (150 \leq t < 200) \\ 215 \text{ mm } (250 \leq t < 300) \end{cases} . \tag{25}$$

The sampling interval is set as 10s. Furthermore, in this experiment, a PI controller is applied by considering the stability of the closed-loop system

At first, the fixed PI controller is applied and initial database is composed by the control result. PI gains are determined by ZN method. The calculated PI gains is shown as follows:

$$K_P = 3.37, \quad K_I = 0.13 \tag{26}$$

The control result is shown in Fig.10. An initial database is generated by using the I/O data in Fig.10 and its PID gains are learned by the DD-FRIT method. The setting parameters to learn the database are shown in Table.II, and the denominator polynomial of $G_m(z^{-1})$ is designed as follows:

$$P(z^{-1}) = 1 - 1.56z^{-1} + 0.61z^{-2} \tag{27}$$

The trajectory of the ISE index is shown in Fig.11. From Fig.11, the index is almost converged around 100 iterations. Therefore the database learned 100 iterations is applied to the system.

The control result by utilizing the DD-FRIT method is shown in Fig.12 and the trajectories of PI gains are shown in Fig.13. These figures show that a tracking property is

TABLE II

SETTING PARAMETERS OF PROPOSED METHOD.

Sampling interval	$T_s = 10$s
Orders of the information vector	$n_y = 3$
	$n_u = 2$
Rise time	$\sigma = 80$s
Damping property	$\delta = 0$
Number of neighbors	$k = 5$
Learning rates	$\eta_P = 1.0 \times 10^{-2}$
	$\eta_I = 1.0 \times 10^{-4}$

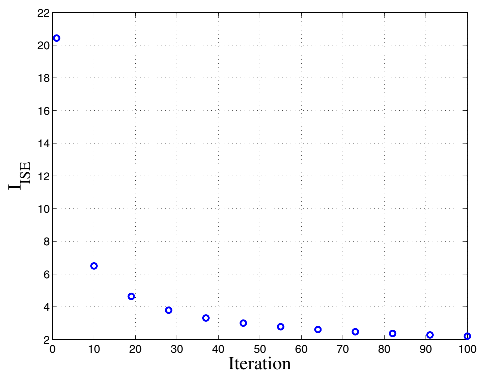

Fig. 11. Trajectory of ISE index at offline learning.

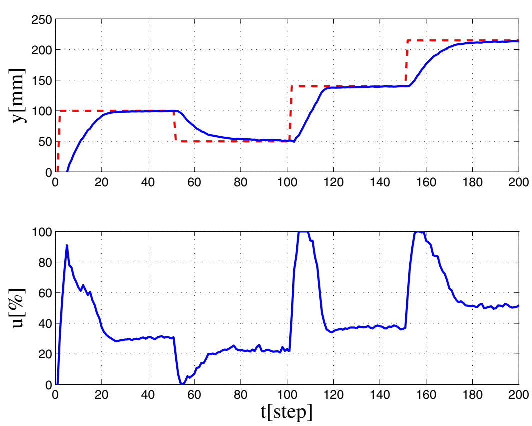

Fig. 12. Control result obtained using proposed method.

drastically improved and the rise-time of the closed loop system also becomes 80s.

VI. CONCLUSIONS

This paper presents a nonlinear controller design method with database utilization. In the method, initial experimental I/O data and control parameters are stored in advance. The control parameters in the constructed database are learned in an offline manner with the DD-FRIT method using the initial experimental I/O data. During operation, the control parameters are tuned according to the DD-PID approach by utilizing the learned database. The effectiveness of the control system was evaluated through the simulation and the experiment. These results showed that the controller learned

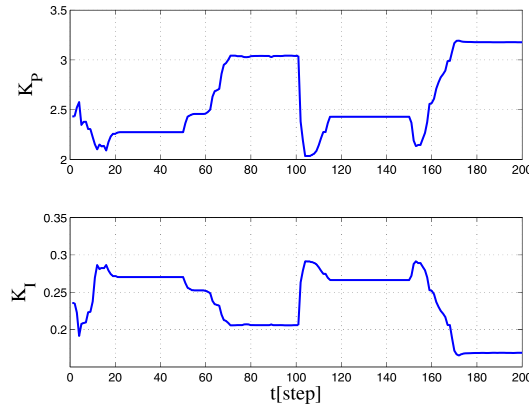

Fig. 13. Trajectories of proportional-integral (PI) gains corresponding to Figure 12.

and tuned its parameters appropriately; thus, good control results can be obtained for nonlinear systems. However, this learning scheme does not consider time delay. Control systems become unstable if there is a long time delay. Therefore, future work will involve extending the learning scheme to systems with a long time delay.

REFERENCES

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

[2] Q.Zheng and H.Kimura, "Just-in-time pid control," *Proc. of the 44th Japan Joint Automatic Control Conference*, pp. 336–339, 2001.

[3] N.Nakpong and S.Yamamoto, "Just-in-time predictive control for a two-wheeled robot," *Proc. ICT-KE2012*, pp. 95–98, 2012.

[4] T.Yamamoto, K.Takao, and T.Yamada, "Design of a data-driven pid controller," *IEEE Trans. Control Systems Technology*, vol. 17, no. 1, pp. 29–39, 2009.

[5] S.Wakitani, K.Nishida, M.Nakamoto, and T.Yamamoto, "Design of a data-driven pid controller using operating data," *Proc. 11th IFAC International Workshop on Adaptation and Learning in Control and Signal Processing*, pp. 587–592, 2013.

[6] O.Kaneko, S.Souma, and T.Fujii, "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.

[7] O.Kaneko, S.Souma, and T.Fujii, "Fictitious reference iterative tuning in the two-degree of freedom control scheme and its application to a facile closed loop system identification," *Trans. of SICE*, vol. 42, no. 1, pp. 17–25, 2006.

[8] I.Mizumoto and H.Tanaka, "Model free design of parallel feedforward compensator for adaptive output feedback control via frit with t-s fuzzy like model," *10th IFAC International Workshop on the Adaptation and Learning in Control and Signal Processing*, pp. 139–144, 2010.

[9] Y.Wakasa, S.Kanagawa, T.Kanya, and Y.Nishimura, "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*, pp. 156–161, 2010.

[10] Y.Matsui, S.Akamatsu, T.Kimura, and K.Nakano, "Fictitious reference iterative tuning for state feedback control of inverted pendulum with inertia rotor," *Proceedings of SICE Annual Conference (SICE2011)*, pp. 1087–1092, 2011.

[11] O.Kaneko, "Data-driven cotnroller tuning: Frit approach," *Proc. of 11th IFAC International Workshop on Adaptation and Learning in Control and Signal Processing*, pp. 326–336, 2013.