DESIGN OF DATA-ORIENTED CONTROLLERS

（データ指向型制御系の設計）

by

Shin WAKITANI

A thesis submitted to the

Faculty of the Graduate School of Engineering

Hiroshima University in partial fulfillment

of the requirements for the

Dr. Eng.

Department of System Cybernetics

November 2013

Supervisor: Professor Toru Yamamoto

Acknowledgements

First, I am grateful to Professor Toru Yamoto of Hiroshima University for his valuable suggestions, guidance, support, patience, and continuous encouragement, without which it would not have been possible to complete my research. It is a great privilege to work under him. I would also like to acknowledge Professor Naoto Yorino and Professor Toshio Tsuji of Hiroshima University, the members of my dissertation committee. Their stimulating suggestions have helped me in writing this thesis.

I express my thanks to Assistant Professor Shinichi Imai of Hiroshima National College of Maritime Technology who supported me in designing the magnetic levitation control equipment. Associate Professor Takao Sato and Assistant Professor Nozomu Araki assisted me in the weigh feeder experiment. I also thank Associate Professor Yoshihiro Ohnishi of Ehime University for his useful advice on the CMAC-FRIT method.

I appreciate the support of Associate Professor Shuichi Ohno and Assistant Professor Masayoshi Nakamoto of Hiroshima University. I express my sincere gratitude to all of the past and present members of the control systems engineering laboratory. I would especially like to thank Dr. Manase, Emmanuel Chifuel, Mr. Jin Yong, and Ms. Kayoko Hayashi, who are members of the Ph.D. course.

Finally, I am grateful to all of the people who have supported me during my work on this thesis. Special thanks go to my family; their continuous support and encouragement made my studies possible.

Contents

Chapter

Chapter 1

Introduction

1.1 Background

Because of the continuing escalation of global competition in the industrial world, more efforts are required for improving productivity, energy saving, and power saving. In order to address these demands, highly-functional control systems are needed.

Controller design schemes are broadly divided into two categories: (i) Model-based and (ii) data-oriented controller designs. In model-based controller design, a mathematical model is required to design a controller; moreover, control performance depends on the accuracy of the system model. However, it can be difficult to obtain an accurate system model even if system identification schemes are introduced. It can also require considerable cost and on-site work. In addition, there are many undesirable situations from the viewpoint of safety of input signals with wide-range frequency in an operating plant, even if a better system model can be obtained using the input signals from the viewpoint of a persistently exciting (PE) condition [1, 2]. Based on these factors, a robust control theory [3] that defined a gap between a real system and a perturbed system, and a controller that considered this gap, was proposed in the 1980s. The theory has been applied in many industrial systems. However, this robust controller design is a worst-case design that guaranties a control performance (especially a robust stability), even if the most inconvenient system model is applied. Therefore, it becomes a conser-

vative design.

On the other hand, data-oriented controller design focuses on the I/O data of control systems (closed-loop data), and a controller is designed using this data to ensure that the control system has the desired control performance. Many data-oriented controller design schemes that are typified by iterative feedback tuning (IFT) [4, 5, 6, 7, 8, 9], virtual reference feedback tuning (VRFT) [10, 11, 12, 13, 14] and fictitious reference iterative tuning (FRIT) [15, 16, 17] have been proposed. In particular, VRFT and FRIT methods are expected to reduce cost and effort of controller design because they are able to tune a controller using only a set of closed-loop data. Moreover, in data-oriented controller design schemes, many approaches that are not only controller design but also closed-loop identification, soft sensing, and control-performance assessment have been actively studied. In recent years, in the manufacturing industry, high-mix/low-volume production predominates in order to correspond to market needs. In addition, industrial machines used in product manufacturing have become increasingly complex to address the wide range of applications. System modeling has also become complex. It requires large amount of time and economical costs to rebuild a system model that reflects a production change. Based on these factors, it is expected that the opportunities as well as the need for data-oriented controller design will increase.

1.2 Data-oriented Controller Design

As mentioned previously, it is expected that the requirement of data-oriented controllers in the industrial world will increase. However, there are many systems such as linear, nonlinear, and time varying systems in real systems. Thus, it is difficult to address all systems with the same method. Therefore, in this study, data-oriented controller design is divided into three types: A "direct controller design," wherein a controlled object can be de-

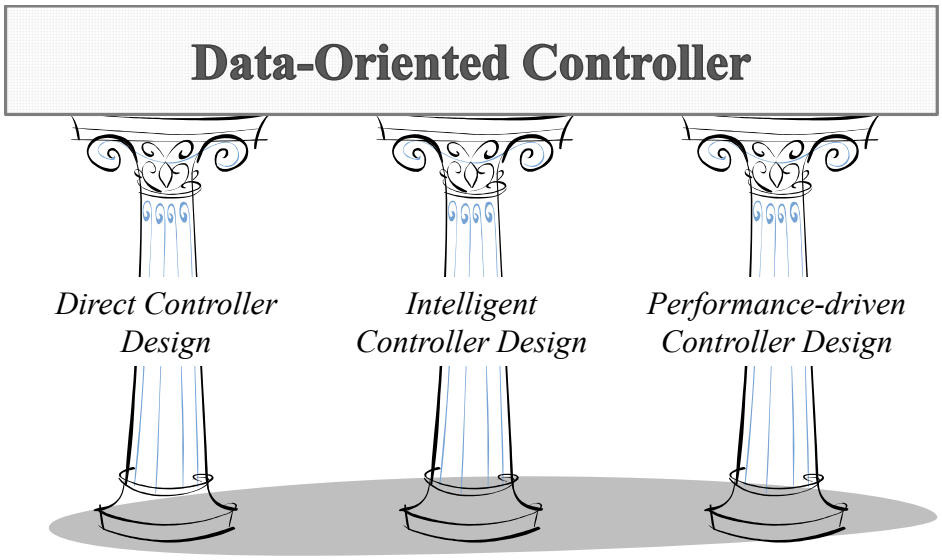

Figure 1.1: Three pillars of data-oriented controller.

scribed as a linear system; an "intelligent controller design," wherein a controlled object has nonlinearity; and a "performance-driven controller design," wherein system properties vary every hour. These design methods can be applied to the real systems being considered. A concept of this study is shown in Figure 1.1. In the next sections, conventional methods to different systems are summarized.

1.2.1 Direct controller design

In process systems, if a signal is inputted to a system, its effects can affect other control loops, and in the worst-case scenario, the entire system could become unstable. Therefore, Hjalmarsson *et al.* proposed the IFT method. This is a data-oriented controller design method that tunes a controller using closed-loop data. According to the IFT method, for a controller with a fixed structure, such as a proportional-integral-derivative (PID) controller [18, 19, 20, 21], its control parameters that minimize a given criterion can be tuned by iterating tests such as by changing the reference signals. Moreover, the method can retune the control parameters without performing system identification; thus, effects from other loops are suppressed because

it does not require an identification signals such as pseudo random binary sequence (PRBS) or M-sequence signals [22]. However, the IFT method requires many iterative tests in order to obtain optimal control parameters. To address this issue, Campi *et al.* proposed the VRFT method as one of the data-oriented controller design schemes that can directly design a controller using only a set of closed-loop data. However, the VRFT method requires power spectral density of the control input in the calculation process. The VRFT method also requires an open-loop test in order to acquire the information in advance [15]. Kaneko *et al.* proposed the FRIT method as a new control parameter tuning approach by using a set of closed-loop data. The FRIT method can perform offline control parameter tuning using a set of closed-loop data by introducing a fictitious reference signal in the unfalsified control [23]. This is an effective method from the viewpoint of reducing the work effort, required time, and costs; as a result, it is now being actively studied [24, 25, 26, 27, 28, 29, 30, 31, 32].

1.2.2 Intelligent controller design

The intelligent controller was first proposed by Fu [33]. The intelligent controller is especially effective for systems with unknown structure and strong nonlinearity that cannot be addressed by adaptive controllers [34]. An example of an intelligent controller is the cerebellar model articulation controller (CMAC) [35], proposed by Albus. In addition, hierarchical neural network (NN) controllers that mimic the behavior of NNs have been researched since the 1980s [36]. In these methods, learning methods are employed to achieve the desired controller performance by updating weight tables or weight coefficients of neurons based on the I/O data obtained by the controller. Iterating the procedure, the desired control performance can be obtained even if a system property is unknown. Moreover, in practical application, methods that use the CMAC and NN outputs for the tuning of PID controllers are pro-

posed [37, 38, 39]. In addition, many other intelligent controllers have been proposed, such as fuzzy controllers [40], and controllers that use a genetic algorithm (GA) [41], immunity algorithm (IA) [42] and particle swarm optimization (PSO)[43].

1.2.3 Performance-driven controller design

A large amount of research on control performance assessment has originated from the work proposed by Harris [44]. This uses minimum variance (MV) index. In the scheme, the control performance is evaluated using variances in the system output only. However, in most practical situations, MV index is not a desired evaluation algorithm because of its demand for excessive control action and/or poor robustness [45, 46]. To solve this problem, Huang and Shah proposed a method of control performance assessment based on the linear quadratic Gaussian (LQG) benchmark [45]. Using this method, the variance of the system output (or the system error) and the variance of the control input can be evaluated together. Moreover, the scheme takes the relationship between the system output and the control input together with a variable control parameter as a trade-off curve, and therefore, allows users to determine the control parameter beforehand based on this trade-off curve. However, in this method, the control parameter is chosen offline. If the system is varied by other unknown factors, the desired control performance cannot be maintained. Because of this, Yamamoto and Shah proposed a performance-driven (performance-adaptive) controller [47] that evaluates the control performance by an online manner. The controller is retuned and redesigned based on the result. The proposed performance-driven controller ensures that a controller is retuned and redesigned only when a control performance index falls below a baseline. Thus, a control system can be updated based on control-performance assessment. This method has been actively researched [48, 49, 50, 51, 52, 53].

1.3 Dissertation Outline

In this thesis, in order to develop a data-oriented controller design scheme, three design methods using closed-loop data for "direct controller design", "intelligent controller design" and "performance-driven controller design" are proposed.

In Chapter 2, a direct PID controller design method based on generalized minimum variance control (GMVC) [54, 55] is presented as a GMV-PID controller design. This design employs an adjustable parameter. An operator can adjust a controller based on its control performances (stability and tracking property) by tuning the adjustable parameter. The effectiveness of the proposed method is evaluated with a simulation and an experiment using a temperature-control system. In addition, it is also suggested that the proposed GMV-PID controller can be easily extended to MIMO systems. The availability of the proposed multivariable PID controller is verified by a simulation and an experiment using a level and temperature-control system.

In Chapter 3, an offline learning method based on closed-loop data unifying CMAC and the FRIT method is proposed and its implementation is considered. Previously, a CMAC-PID controller was proposed as an effective method for nonlinear systems and its usefulness is validated. However, it has been pointed out that CMAC requires many time-consuming experiments, and requires large amounts of memory to implement the algorithm on a computer. These factors became obstacles for the implementation of a CMAC-PID controller. In this chapter, it is shown that CMAC can be learned in an offline manner using closed-loop data by introducing a fictitious reference signal into the FRIT method. Moreover, a memory reduction method is proposed that converts the learned CMAC to a nonlinear function using a group method of data handling (GMDH) network [56]. These effectivenesses are validated by a simulation and an experiment for applying the method to

a magnetic levitation device.

In Chapter 4, a performance-driven controller design method that evaluates the control performance using closed-loop data and performs controller redesign is proposed. According to this method, control performance can be assessed without the use of any system models by focusing on a ratio of variances of control errors and differential control input. Moreover, by reflecting the result of the assessment to only one user-specified parameter included in a controller, it can perform "one-parameter tuning" which unifies the control performance assessment and controller redesign. In this chapter, the effectiveness of the proposed method is verified by a simulation and an experiment using a weigh feeder.

In Chapter 5, the conclusions and outstanding issues are presented. Finally, future development for a data-oriented controller is discussed.

Chapter 2

Design of Direct GMV-PID Controller

2.1 Introduction

In this chapter, a direct controller design method using closed-loop data is presented. The proportional-integral-derivative (PID) controller [18, 19, 20, 21] has been applied in many process systems, including chemical and petroleum processes because its control structure is simple and the meaning of the PID parameters is clear. It is very important to tune the PID parameters because the control performance of a closed-loop system is strongly affected by these adjusted parameters. In most model-based developments of PID controllers, system identification is first required to develop a system model. However, in order to accurately identify the system model, it is preferred to input an M-sequence signal from the viewpoint of a persistently exciting (PE) condition [57]. In large processes such as petroleum processes, it is difficult to input the signal because system output of a control loop affects the other loop outputs. On the other hand, a data-oriented controller design scheme such as FRIT [15, 16, 17] can use the closed-loop data to determine the control parameters. This effect is reduced by using closed-loop data obtained with a stable PID controller. Therefore, it is considered that a data-oriented controller design scheme is effective, especially for process systems.

In this chapter, a direct PID controller design based on the generalized minimum variance control (GMVC) [57, 54, 55] is presented as a data-oriented PID design scheme using closed-loop data. In the GMVC, the criterion of the

generalized output composed of a control error and differencing control input is first defined. The control law is derived based on the minimization of the criterion. Moreover, control parameters of the GMVC can be directly determined from the closed-loop data by applying the least squares method (LSM). By replacing these control parameters by PID gains, the GMV-PID controller can be derived. Furthermore, in the GMV-PID controller, the tracking property, or the stability of the controller, can be arbitrarily adjusted by tuning λ, which is a weight coefficient for the deferential input. In this chapter, the design scheme of the GMVC is first explained. Then, the method for converting the control parameters of the GMVC into PID gains is discussed. The effectiveness of the proposed method is evaluated by a numerical example and an experiment with a temperature-control system. Moreover, it is shown that the GMV-PID controller design scheme can be easily extended to MIMO systems. The availability of the method is verified by a simulation and an experiment for a level and temperature-control system.

2.2 PID Controller Design based on GMVC

2.2.1 System description

A control object is assumed to be described as the following system:

$$A(z^{-1})y(t) = z^{-(k+1)}B(z^{-1})u(t) + \frac{\xi(t)}{\Delta}, \tag{2.1}$$

$$\left.\begin{aligned} A(z^{-1}) &= 1 + a_1 z^{-1} + a_2 z^{-2} \\ B(z^{-1}) &= b_0 + b_1 z^{-1} + \cdots + b_m z^{-m} \end{aligned}\right\}. \tag{2.2}$$

In (2.1), $u(t)$ and $y(t)$ represent the control input and the system output, respectively; k is the minimum time-lag estimated by an operator (k is set as 0 when the time-lag is unknown); and $\xi(t)$ is the white Gaussian noise with zero mean and variance σ^2. Here, z^{-1} denotes the backward operator: $z^{-1}y(t) = y(t-1)$; Δ is the differencing operator: $\Delta := 1 - z^{-1}$. The model described in (2.1) is the so-called CARIMA (controlled auto-regressive

and integrated moving average) model [58, 59, 60] that has been frequently utilized in the process industries.

2.2.2 Direct controller design based on GMVC

The GMVC law for (2.1) is derived to minimize the following criterion:

$$J = E[\phi^2(t+k+1)], \tag{2.3}$$

where $\phi(t+k+1)$ is the generalized output given by following equation:

$$\phi(t+k+1) := P(z^{-1})y(t+k+1) + \lambda\Delta u(t) - P(1)r(t). \tag{2.4}$$

In (2.4), $r(t)$ denotes the step reference signal and λ is the weight coefficient for the differential control input $\Delta u(t)$. It is set by a user arbitrarily. Next, the following Diophantine equation is introduced:

$$P(z^{-1}) = \Delta E(z^{-1})A(z^{-1}) + z^{-(k+1)}F(z^{-1}), \tag{2.5}$$

$$\left.\begin{aligned} E(z^{-1}) &= 1 + e_1 z^{-1} \cdots + e_k z^{-k} \\ F(z^{-1}) &= f_0 + f_1 z^{-1} + f_2 z^{-2} \end{aligned}\right\}, \tag{2.6}$$

where the orders of $E(z^{-1})$ and $F(z^{-1})$ are determined to calculate these coefficients uniquely from $\Delta A(z^{-1})$ and $P(z^{-1})$. $P(z^{-1})$ is the design polynomial and is based on the following equation:

$$P(z^{-1}) = 1 + p_1 z^{-1} + p_2 z^{-2}, \tag{2.7}$$

$$\left.\begin{aligned} p_1 &= -2\exp(-\tfrac{\rho}{2\mu})\cos(\tfrac{\sqrt{4\mu-1}}{2\mu}\rho) \\ p_2 &= \exp(-\tfrac{\rho}{\mu}) \\ \rho &:= T_s/\sigma \\ \mu &:= 0.25(1-\delta) + 0.51\delta \end{aligned}\right\}. \tag{2.8}$$

In (2.8), T_s denotes the sampling time. σ and δ are the parameters that express the rise time and damping property that are set by an operator, respectively. The rise time σ is the time that the system output attains about 60% of the value of the 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 $\delta = 1.0$. From (2.1), (2.4), and (2.5), the predictive value of the generalized output after $k+1$ can be obtained as per the following equation:

$$\begin{aligned}\phi(t+k+1) = F(z^{-1})y(t) + \{E(z^{-1})B(z^{-1}) + \lambda\}\Delta u(t) \\ -P(1)r(t) + E(z^{-1})\xi(t+k+1).\end{aligned} \tag{2.9}$$

Here, the optimal predictive value at t is defined by:

$$\hat{\phi}(t+k+1|t) := F(z^{-1})y(t) + \{G(z^{-1}) + \lambda\}\Delta u(t) - P(1)r(t), \tag{2.10}$$

where,

$$\begin{aligned}G(z^{-1}) &:= E(z^{-1})B(z^{-1}) \\ &= g_0 + g_1 z^{-1} + \cdots + g_{n_g} z^{-n_g},\end{aligned} \tag{2.11}$$

$$n_g := m + k. \tag{2.12}$$

The following equation is obtained from (2.9) and (2.10):

$$\phi(t+k+1) = \hat{\phi}(t+k+1|t) + E(z^{-1})\xi(t+k+1). \tag{2.13}$$

From (2.3) and (2.13), the GMVC law to minimize the criterion J can be derived by $\hat{\phi}(t+k+1|t) = 0$ [57]. The control law is described by the following equation:

$$\Delta u(t) = \frac{P(1)}{G(z^{-1}) + \lambda} r(t) - \frac{F(z^{-1})}{G(z^{-1}) + \lambda} y(t). \tag{2.14}$$

To obtain an optimal predictive value $\hat{\phi}(t+k+1|t)$ in (2.13), it is necessary to compute $E(z^{-1})$ and $F(z^{-1})$ using (2.5). However, they require the system parameters $A(z^{-1})$ and $B(z^{-1})$ for the calculation. Therefore, a system identification is required. However it is difficult to accurately identify system parameters when a system has a high-order lag factor and/or uncertainty. Thus, in this study, control parameters are computed using only

closed-loop data based on the implicit GMVC method without system identification. More specifically, the predictive error constructed by the generalized output and the optimal predictive value in the closed-loop data are defined as follows:

$$\begin{aligned}\varepsilon(t+k+1) &:= \phi_0(t+k+1) - \hat{\phi}_0(t+k+1|t) \\ &= P(z^{-1})y_0(t+k+1) \\ &\quad -\{F(z^{-1})y_0(t) + G(z^{-1})\Delta u_0(t)\},\end{aligned} \tag{2.15}$$

where, $\phi_0(t+k+1)$ and $\hat{\phi}_0(t+k+1|t)$ are the generalized output and predicted generalized output calculated using the obtained closed-loop data, respectively; and $y_0(t)$ and $\Delta u_0(t)$ are the system output and deferential input of the closed-loop data, respectively. LSM is applied to minimize the square sum of these residual errors and calculate the control parameters of $F(z^{-1})$ and $G(z^{-1})$. That is, the parameters of $F(z^{-1})$ and $G(z^{-1})$ are identified without system parameters. By applying the control parameters to (2.14), the GMVC controller can be designed directly.

2.2.3 Design of GMV-PID controller

Next, the control parameters computed using the GMVC are converted into PID gains. The velocity type of the PID control law is given by the following equation:

$$\Delta u(t) = K_I e(t) - K_P \Delta y(t) - K_D \Delta^2 y(t), \tag{2.16}$$

where $e(t)$ expresses the control error:

$$e(t) := r(t) - y(t). \tag{2.17}$$

K_P, K_I, and K_D denote the proportional gain, integral gain, and derivative gain. By replacing the polynomial $G(z^{-1})$ in (2.14) by the steady-state term $G(1)$, the following equation can be obtained:

$$\Delta u(t) = \frac{P(1)}{G(1)+\lambda} r(t) - \frac{F(z^{-1})}{G(1)+\lambda} y(t). \tag{2.18}$$

By comparing the coefficients of (2.16) and (2.18), the following relationship between the control parameters and PID gains can be obtained:

$$K_P = -\frac{f_1 + 2f_2}{G(1) + \lambda},\ K_I = \frac{f_0 + f_1 + f_2}{G(1) + \lambda},\ K_D = \frac{f_2}{G(1) + \lambda}. \tag{2.19}$$

A detailed derivation process of (2.19) is presented in the appendix. Using this permutation, the GMV controller is replaced by the PID controller.

2.3 Numerical Example

In order to evaluate the effectiveness of the proposed method, the method is applied to the following fourth-order system [20]:

$$G(s) = \frac{12s + 8}{20s^4 + 113s^3 + 147s^2 + 62s + 8}, \tag{2.20}$$

The discrete time model with modeling error related to (2.20) is obtained as follows to discretize the above equation by $T_s = 1$[s]:

$$\begin{aligned} y(t) = {} & 1.835y(t-1) - 1.088y(t-2) + 0.2114y(t-3) \\ & -0.0035y(t-4) + 0.0358u(t-1) + 0.0356u(t-2) \\ & -0.0245u(t-3) - 0.0017u(t-4) + \xi(t)/\Delta. \end{aligned} \tag{2.21}$$

Here, $\xi(t)$ is the white Gaussian noise with zero mean and variance 1.0×10^{-4}. The reference value is set as $r(t) = 10$ in this simulation.

First, a control result using a conventional fixed PID controller whose PID gains are calculated using the Chien, Hrones, and Reswick (CHR) method [19], with special attention to stability, is run. The PID gains are set as:

$$K_P = 2.41,\ K_I = 0.442,\ K_D = 1.64. \tag{2.22}$$

Figure 2.1, shows the stable control result obtained using the CHR method. Next, the PID gains are computed using the proposed method, where the method uses the result shown in Figure 2.1 as the closed-loop data. The order of $B(z^{-1})$ is set as $m = 5$ and the estimated minimum time-lag is set

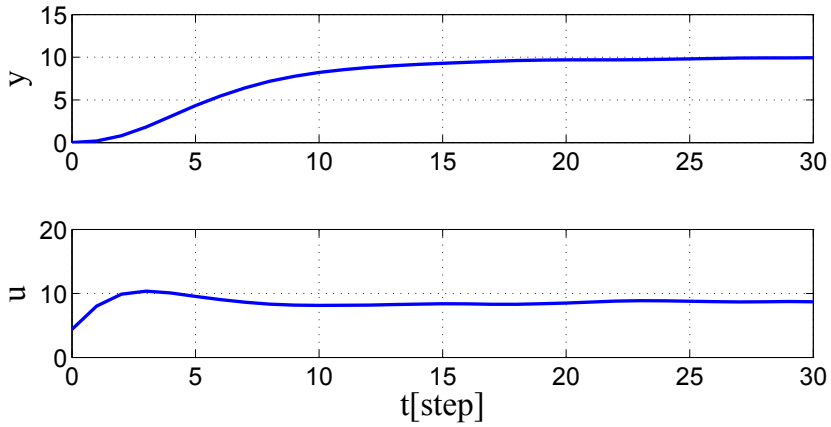

Figure 2.1: Simulation result obtained using Chien, Hrones, and Reswick method.

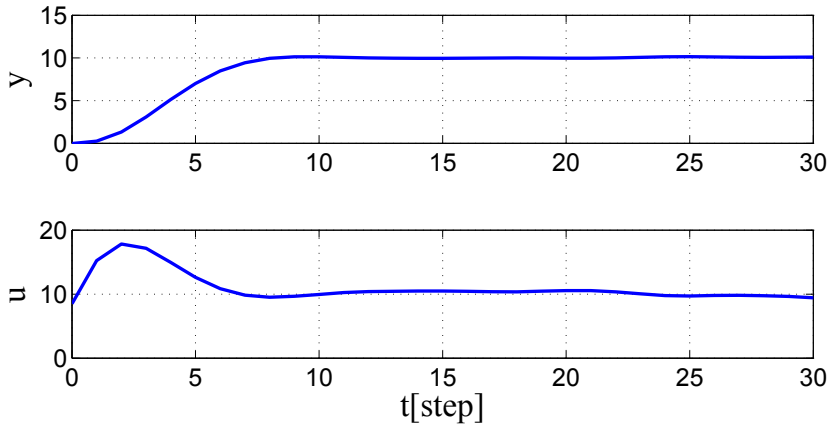

Figure 2.2: Simulation result obtained using proposed method.

as $k = 1$. In addition, the design polynomial is designed as follows when the rise time σ is set as 3[s] and the damping property $\delta = 0$:

$$P(z^{-1}) = 1 - 1.03z^{-1} + 0.264z^{-2}. \tag{2.23}$$

The PID gains when λ is set as 0.1 are calculated as follows:

$$K_P = 2.76,\ K_I = 0.84,\ K_D = 2.30. \tag{2.24}$$

The control result obtained these PID gains is shown in Figure 2.2. The figure shows that the proposed PID controller improved the tracking property compared to the result obtained using the CHR-PID controller.

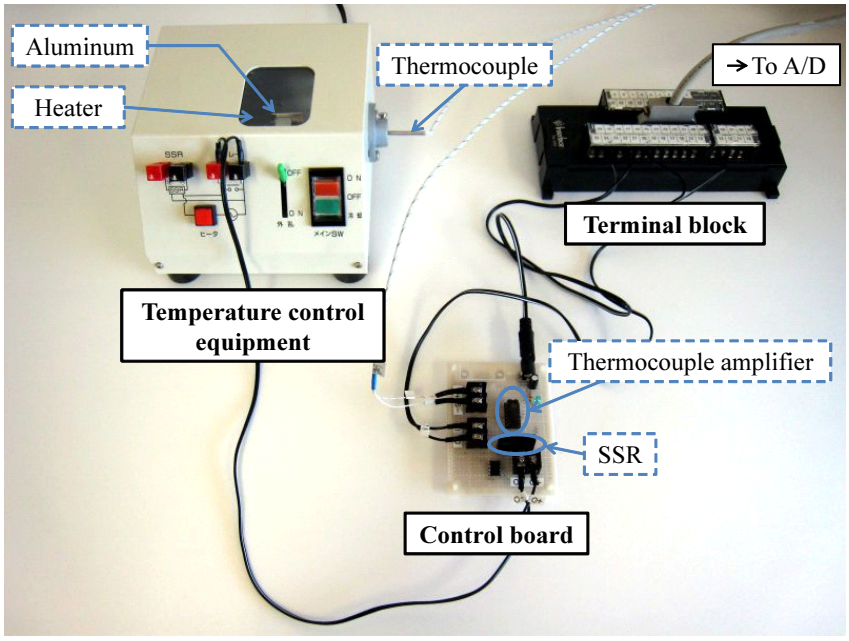

Figure 2.3: Configuration of temperature-control system.

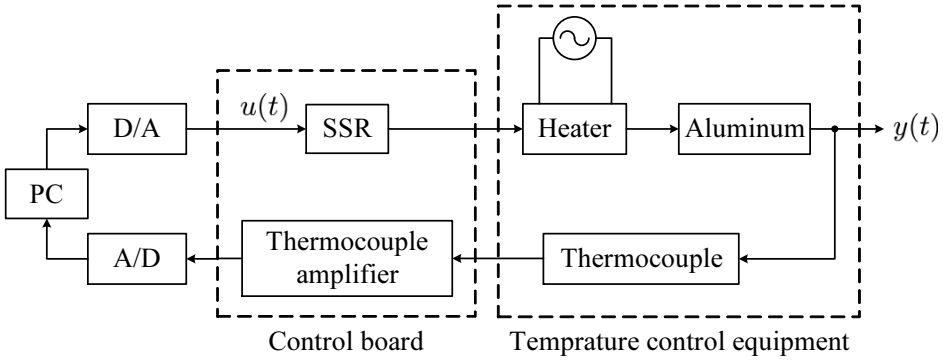

Figure 2.4: Schematic of temperature-control system.

2.4 Application to Temperature-control System

In this section, the proposed method is applied to a temperature-control system shown in Figure 2.3. In addition, the schematic of the equipment is shown in Figure 2.4. In this system, the system output $y(t)$ is the temperature of the aluminum and the control input $u(t)$ is the duty ratio of the heater limited from (0 – 100%). The reference signal $r(t)$ is set as 150[°C] and the sampling time T_s is set as 5.0[s].

First, a conventional fixed PID controller whose parameters are adjusted

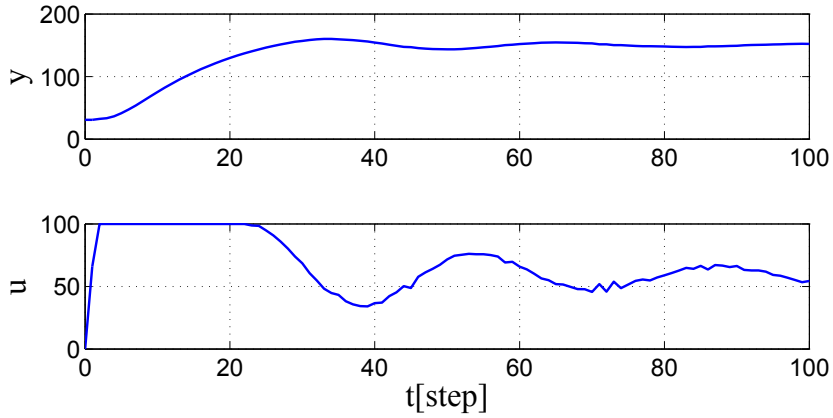

Figure 2.5: Experimental result obtained Chien, Hrones, and Reswick method.

by the CHR method is applied. The PID gains are calculated as follows:

$$K_P = 2.02,\ K_I = 0.587,\ K_D = 3.47. \tag{2.25}$$

The control result is shown in Figure 2.5. This shows that the system output is not stabilized around 100[step].

Next, the PID gains are tuned using the proposed method based on the result shown in Figure 2.5. The order of $B(z^{-1})$ is set as $m = 1$ and the estimated time-lag is set as $k = 4$. Moreover, $P(z^{-1})$ is designed as (2.26) where $\sigma = 30$[s] and $\delta = 0$:

$$P(z^{-1}) = 1 - 1.43z^{-1} + 0.513z^{-2}. \tag{2.26}$$

The weight coefficient λ is set as 0.01. The control result is shown in Figure 2.6. Figure 2.6 shows that the system output is stabilized around 30[step]; as a result, good control performance is obtained.

2.5 Extending to MIMO Systems

The proposed GMV-PID controller can be extended to MIMO systems. In this section, the design method of a multivariable GMV-PID controller is considered.

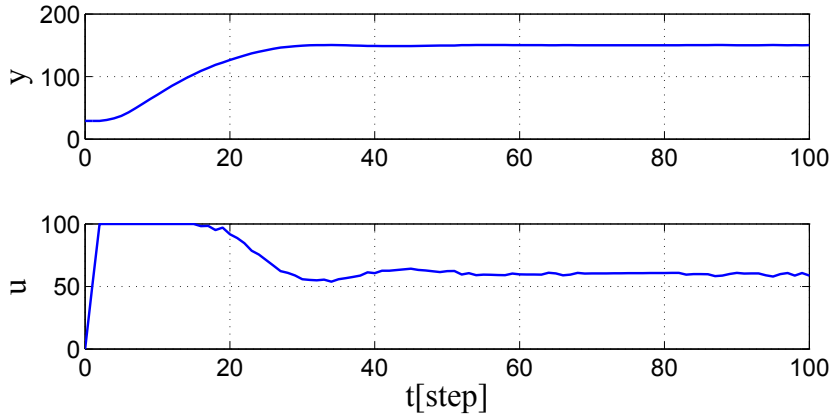

Figure 2.6: Experimental result obtained using proposed method, where $\lambda = 0.01$.

2.5.1 System description

A control object is assumed to be described as the following $p \times p$ system:

$$\boldsymbol{A}(z^{-1})\boldsymbol{y}(t) = z^{-(k+1)}\boldsymbol{B}(z^{-1})\boldsymbol{u}(t) + \frac{\boldsymbol{\xi}(t)}{\Delta}. \tag{2.27}$$

In (2.27), $\boldsymbol{y}(t)$ and $\boldsymbol{u}(t)$ indicate the system output vector and the control input vector, and $\boldsymbol{\xi}(t)$ is the white Gaussian noise vector whose elements have zero mean and a variance $\sigma_i^2 \quad (i = 1, \ldots, p)$. These vectors are defined by the following equations:

$$\boldsymbol{y}(t) := [y_1(t), y_2(t), \ldots, y_p(t)]^T, \tag{2.28}$$

$$\boldsymbol{u}(t) := [u_1(t), u_2(t), \ldots, u_p(t)]^T, \tag{2.29}$$

$$\boldsymbol{\xi}(t) := [\xi_1(t), \xi_2(t), \ldots, \xi_p(t)]^T. \tag{2.30}$$

Additionally, $\boldsymbol{A}(z^{-1})$ and $\boldsymbol{B}(z^{-1})$ are polynomial matrices shown as follows:

$$\boldsymbol{A}(z^{-1}) := \mathrm{diag}\left[A_1(z^{-1}), \ldots, A_p(z^{-1})\right], \tag{2.31}$$

$$\boldsymbol{B}(z^{-1}) := \begin{bmatrix} B_{11}(z^{-1}) & \cdots & B_{1p}(z^{-1}) \\ \vdots & \ddots & \vdots \\ B_{p1}(z^{-1}) & \cdots & B_{pp}(z^{-1}) \end{bmatrix}, \tag{2.32}$$

where

$$A_i(z^{-1}) := 1 + a_{i1}z^{-1} + a_{i2}z^{-2} \tag{2.33}$$

$$B_{ij}(z^{-1}) := b_{ij0} + b_{ij1}z^{-1} + \cdots + b_{ijm}z^{-m}. \tag{2.34}$$

In (2.33), $A_i(z^{-1})$ is at most a second-order polynomial corresponding to the i-th output. In (2.34), $B_{ij}(z^{-1})$ is a polynomial corresponding to the i-th input coupled by the j-th input; m expresses the order of $B_{ij}(z^{-1})$ and is determined by an operator considering the high-order lag factor of a system, and so on.

2.5.2 Design of multivariable GMV-PID controller

The multivariable GMVC law for the above system is derived based on minimizing the following criterion:

$$J = E[\boldsymbol{\Phi}^T(t+k+1)\boldsymbol{\Phi}(t+k+1)], \tag{2.35}$$

where $\boldsymbol{\Phi}(t+k+1)$ is the generalized output vector which is given by the following equation:

$$\boldsymbol{\Phi}(t+k+1) := \boldsymbol{P}(z^{-1})\boldsymbol{y}(t+k+1) + \boldsymbol{\Lambda}\Delta\boldsymbol{u}(t) - \boldsymbol{P}(1)\boldsymbol{r}(t), \tag{2.36}$$

where, $\boldsymbol{\Lambda}$ and $\boldsymbol{r}(t)$ are defined as the following equations:

$$\boldsymbol{\Lambda} := \mathrm{diag}[\lambda_1, \lambda_2, \ldots, \lambda_p], \tag{2.37}$$

$$\boldsymbol{r}(t) := [r_1(t), r_2(t), \ldots, r_p(t)]^T. \tag{2.38}$$

In (2.37) and (2.38), $\boldsymbol{\lambda}$ and $\boldsymbol{r}(t)$ indicate the weight coefficient vector for $\Delta\boldsymbol{u}(t)$ and the step reference signal vector. $\boldsymbol{P}(z^{-1})$ is a design polynomial matrix designed based on the following equations:

$$\boldsymbol{P}(z^{-1}) = \boldsymbol{I} + \boldsymbol{P}_1 z^{-1} + \boldsymbol{P}_2 z^{-2}, \tag{2.39}$$

$$\left.\begin{aligned} \boldsymbol{P}_1 &= \mathrm{diag}[p_{11}, p_{21}, \ldots, p_{p1}] \\ \boldsymbol{P}_2 &= \mathrm{diag}[p_{12}, p_{22}, \ldots, p_{p2}] \end{aligned}\right\}, \tag{2.40}$$

where, $\boldsymbol{I}$ expresses a unit matrix. In addition, p_{i1} and p_{i2} in (2.40) are given by the following equations:

$$\left.\begin{aligned} p_{i1} &= -2e^{-\frac{\rho_i}{2\mu_i}}\cos(\tfrac{\sqrt{4\mu_i-1}}{2\mu_i}\rho_i) \\ p_{i2} &= e^{-\frac{\rho_i}{\mu_i}} \\ \rho_i &:= T_s/\sigma_i \\ \mu_i &:= 0.25(1-\delta_i)+0.51\delta_i \end{aligned}\right\}. \tag{2.41}$$

In (2.41), σ_i and δ_i express the rise time and damping property of the i-th system, respectively. Next, the Diophantine equation (2.42) is introduced:

$$\boldsymbol{P}(z^{-1}) = \Delta\boldsymbol{E}(z^{-1})\boldsymbol{A}(z^{-1}) + z^{-(k+1)}\boldsymbol{F}(z^{-1}), \tag{2.42}$$

$$\left.\begin{aligned} \boldsymbol{E}(z^{-1}) &:= \boldsymbol{I} + \boldsymbol{E}_1 z^{-1} + \boldsymbol{E}_2 z^{-2}\cdots + \boldsymbol{E}_k z^{-k} \\ \boldsymbol{F}(z^{-1}) &:= \boldsymbol{F}_0 + \boldsymbol{F}_1 z^{-1} + \boldsymbol{F}_2 z^{-2} \end{aligned}\right\}, \tag{2.43}$$

where, $\boldsymbol{E}(z^{-1})$ and $\boldsymbol{F}(z^{-1})$ are matrix polynomials whose orders are respectively determined by the orders of $\boldsymbol{A}(z^{-1})$ and $\boldsymbol{P}(z^{-1})$.

From (2.27), (2.36) and (2.42), the $k+1$ step ahead generalized output vector $\boldsymbol{\Phi}(t+k+1)$ can be obtained using the following equation:

$$\begin{aligned} \boldsymbol{\Phi}(t+k+1) = \boldsymbol{F}(z^{-1})\boldsymbol{y}(t) + \{\boldsymbol{E}(z^{-1})\boldsymbol{B}(z^{-1}) + \boldsymbol{\Lambda}\}\Delta\boldsymbol{u}(t) \\ -\boldsymbol{P}(1)\boldsymbol{r}(t) + \boldsymbol{E}(z^{-1})\boldsymbol{\xi}(t+k+1). \end{aligned} \tag{2.44}$$

Here, the optimal predictive value vector at t is defined by:

$$\hat{\boldsymbol{\Phi}}(t+k+1|t) := \boldsymbol{F}(z^{-1})\boldsymbol{y}(t) + \{\boldsymbol{G}(z^{-1}) + \boldsymbol{\Lambda}\}\Delta\boldsymbol{u}(t) - \boldsymbol{P}(1)\boldsymbol{r}(t), \tag{2.45}$$

where $\boldsymbol{G}(z^{-1})$ is defined as follows:

$$\boldsymbol{G}(z^{-1}) := \boldsymbol{E}(z^{-1})\boldsymbol{B}(z^{-1}) \tag{2.46}$$

$$= \boldsymbol{G}_0 + \boldsymbol{G}_1 z^{-1} + \cdots + \boldsymbol{G}_{n_g} z^{-n_g} \tag{2.47}$$

where $n_g = m + k$. From the relationship between (2.44) and (2.45), the following equation is obtained:

$$\boldsymbol{\Phi}(t+k+1) = \hat{\boldsymbol{\Phi}}(t+k+1|t) + \boldsymbol{E}(z^{-1})\boldsymbol{\xi}(t+k+1). \tag{2.48}$$

From (2.35) and (2.48), the following GMVC law minimizing the criterion (2.35) is derived when $\hat{\boldsymbol{\Phi}}(t+k+1|t)=0$:

$$\Delta \boldsymbol{u}(t) = \{\boldsymbol{G}(z^{-1}) + \boldsymbol{\Lambda}\}^{-1}\{\boldsymbol{P}(1)\boldsymbol{r}(t) - \boldsymbol{F}(z^{-1})\boldsymbol{y}(t)\}. \tag{2.49}$$

In this case, the system parameters of $\boldsymbol{A}(z^{-1})$ and $\boldsymbol{B}(z^{-1})$ are required as in Section 2.2.2 in order to calculate the coefficients of $\boldsymbol{E}(z^{-1})$ and $\boldsymbol{F}(z^{-1})$. Therefore, it is proposed to calculate the parameters $\boldsymbol{E}(z^{-1})$ and $\boldsymbol{F}(z^{-1})$ by using the closed-loop data based on the implicit GMVC method. More specifically, the predictive error between the generalized output and the optimal predictive value is defined as follows:

$$\begin{aligned}\boldsymbol{\varepsilon}(t+k+1) &:= \boldsymbol{\Phi}_0(t+k+1) - \hat{\boldsymbol{\Phi}}_0(t+k+1|t) \\ &= \boldsymbol{P}(z^{-1})\boldsymbol{y}_0(t+k+1) \\ &\quad -\{\boldsymbol{F}(z^{-1})\boldsymbol{y}_0(t) + \boldsymbol{G}(z^{-1})\Delta\boldsymbol{u}_0(t)\}\end{aligned} \tag{2.50}$$

The LSM method is applied to minimize the square sum of these residual errors using the closed-loop data; then, the parameters of $\boldsymbol{F}(z^{-1})$ and $\boldsymbol{G}(z^{-1})$ are obtained. By applying these parameters to the GMVC law in (2.49), the multivariable GMV controller can be designed directly. Next, a conversion method from the multivariable GMVC to a multivariable GMV-PID controller is presented. The velocity type of the PID control law for MIMO systems is given by the following equation:

$$\Delta \boldsymbol{u}(t) = \boldsymbol{K}_I\boldsymbol{e}(t) - \boldsymbol{K}_P\Delta\boldsymbol{y}(t) - \boldsymbol{K}_D\Delta^2\boldsymbol{y}(t), \tag{2.51}$$

where

$$\boldsymbol{e}(t) := \boldsymbol{r}(t) - \boldsymbol{y}(t). \tag{2.52}$$

$\boldsymbol{K}_P$, $\boldsymbol{K}_I$, and $\boldsymbol{K}_D$ are the proportional gain matrix, integral gain matrix, and derivative gain matrix. By replacing the polynomial $\boldsymbol{G}(z^{-1})$ in (2.49) with the steady-state term $\boldsymbol{G}(1)$, the following equation can be obtained:

$$\Delta \boldsymbol{u}(t) = \{\boldsymbol{G}(1) + \boldsymbol{\Lambda}\}^{-1}\{\boldsymbol{P}(1)\boldsymbol{r}(t) - \boldsymbol{F}(z^{-1})\boldsymbol{y}(t)\}. \tag{2.53}$$

To compare the coefficients in (2.51) and (2.53), the following relationship between the GMV control parameter matrices and the PID gains matrices can be obtained:

$$\boldsymbol{K_P} = -\boldsymbol{\nu}(\boldsymbol{F}_1 + 2\boldsymbol{F}_2), \tag{2.54}$$

$$\boldsymbol{K_I} = \boldsymbol{\nu}(\boldsymbol{F}_0 + \boldsymbol{F}_1 + \boldsymbol{F}_2), \tag{2.55}$$

$$\boldsymbol{K_D} = \boldsymbol{\nu}\boldsymbol{F}_2, \tag{2.56}$$

$$\boldsymbol{\nu} := \{\boldsymbol{G}(1) + \boldsymbol{\Lambda}\}^{-1}. \tag{2.57}$$

With this replacement, the multivariable GMV-PID controller can be constructed.

2.5.3 Numerical example

To evaluate the effectiveness of the proposed method, it is applied to the following 2×2 system:

$$\begin{aligned}\boldsymbol{y}(t) = &\begin{bmatrix} 0.9 & 0 \\ 0 & 0.8 \end{bmatrix} \boldsymbol{y}(t-1) - \begin{bmatrix} 0.2 & 0 \\ 0 & 0.1 \end{bmatrix} \boldsymbol{y}(t-2) \\ &+ \begin{bmatrix} 0.6 & 0.5 \\ 0.4 & 0.5 \end{bmatrix} \boldsymbol{u}(t-1) + \begin{bmatrix} 0.3 & 0.4 \\ 0.3 & 0.3 \end{bmatrix} \boldsymbol{u}(t-2) + \frac{\boldsymbol{\xi}(t)}{\Delta},\end{aligned} \tag{2.58}$$

where the sampling time T_s is set as 1.0[s]. The elements in $\boldsymbol{\xi}(t)$ are the white Gaussian noise with zero means and variance 1.0×10^{-4}. Set values of both systems are switched over to 1.0 or 2.0 each instant. First, in order to obtain a set of operating data, the system is controlled using a multiloop PID controller. More specifically, PID gains in each loop are determined by the CHR method without considering the interference. According to this strategy, the PID gains are determined as follows:

$$\boldsymbol{K}_P = \begin{bmatrix} 0.33 & 0 \\ 0 & 0.49 \end{bmatrix}, \ \boldsymbol{K}_I = \begin{bmatrix} 0.18 & 0 \\ 0 & 0.21 \end{bmatrix}, \ \boldsymbol{K}_D = \begin{bmatrix} 0.19 & 0 \\ 0 & 0.26 \end{bmatrix}. \tag{2.59}$$

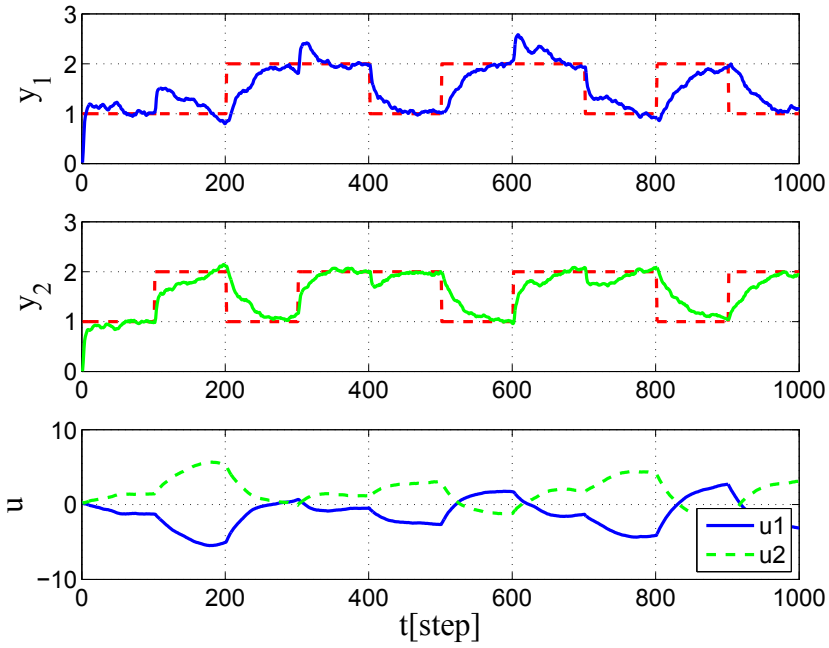

Figure 2.7: Simulation result obtained using multiloop proportional-integral-derivative (PID) controller.

The control result obtained using the multiloop PID controller is shown in Figure 2.7. The control result shows that the control inputs interfere with the system outputs, In particular, a system output varies considerably large when another loop's set value is changed.

Next, the proposed PID gains that consider each interference are calculated using the control result shown in Figure 2.7. For calculating the PID gains, the user specified parameters are set as follows:

$$\boldsymbol{\sigma} = \mathrm{diag}[1.0, 1.0], \quad \boldsymbol{\delta} = \mathrm{diag}[0, 0], \quad \boldsymbol{\Lambda} = \mathrm{diag}[1, 1], \tag{2.60}$$

and the design polynomial matrix is obtained as follows:

$$\boldsymbol{P}(z^{-1}) = \boldsymbol{I} + \begin{bmatrix} -0.27 & 0 \\ 0 & -0.27 \end{bmatrix} z^{-1} + \begin{bmatrix} 0.018 & 0 \\ 0 & 0.018 \end{bmatrix} z^{-2}. \tag{2.61}$$

It is assumed that the order of $\boldsymbol{B}(z^{-1})$ and the minimum estimated time-lag k are well-known. After setting these parameters, PID gains are calculated

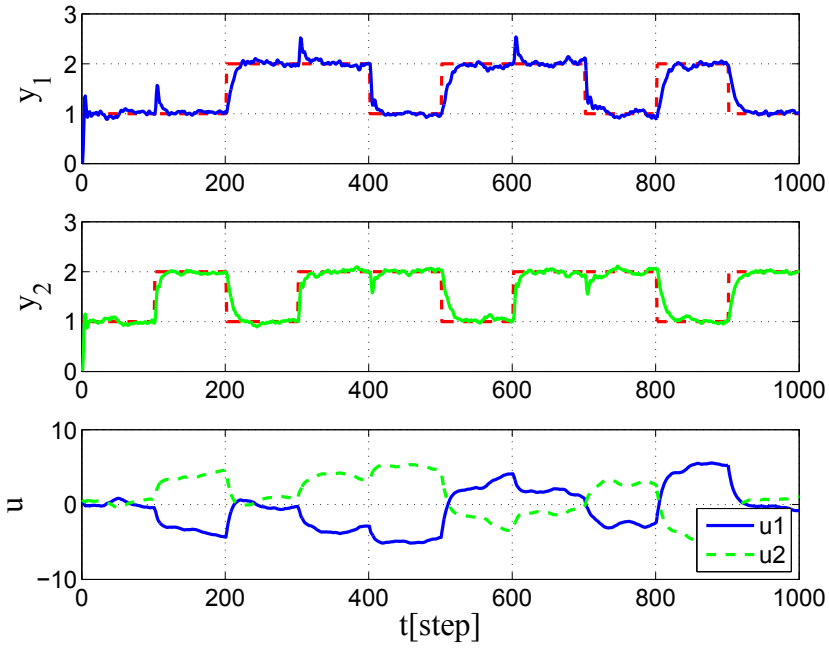

Figure 2.8: Simulation result obtained using proposed method.

as follows by the proposed method:

$$\boldsymbol{K}_P = \begin{bmatrix} 0.06 & 0.19 \\ 0.13 & 0.02 \end{bmatrix}, \ \boldsymbol{K}_I = \begin{bmatrix} 0.41 & -0.21 \\ -0.17 & 0.43 \end{bmatrix}, \ \boldsymbol{K}_D = \begin{bmatrix} 0.11 & 0.05 \\ -0.05 & 0.02 \end{bmatrix}. \quad (2.62)$$

The control result obtained using the proposed multivariable PID controller is shown in Figure 2.8. It is clear that the proposed multivariable PID controller works well in comparison to the multiloop PID controller.

2.5.4 Application to level and temperature-control system

In this section, the usefulness of the proposed method is evaluated by applying it to the level and temperature-control system shown in Figure 2.9. In this system, cold and hot water enter the tank, and the liquid discharge is adjusted by manipulating the exit valve position v. In this experiment, the control objectives are to regulate the water level in the tank y_1 and the temperature of the water discharged from the tank y_2 by manipulating the control valves u_1 and u_2, where, the exit valve position v is stabilized. In this example, the sampling time is set as 5[s], and the set values are set as

Figure 2.9: Level and temperature-control system.

$y_1 = 50$ or 80[mm] and $y_2 = 25$ or 30[°C] at all times. Furthermore, in this experiment, a proportional-integral (PI) controller is applied by considering the stability of the closed-loop system (i.e., the maximum order of $\boldsymbol{A}(z^{-1})$ is 1). The schematic of the system is shown in Figure 2.10.

First, the system is controlled using the multiloop PI controller in order to obtain a set of closed-loop data in common with the simulation. Multiloop PI gains are determined based on the CHR method and are obtained as follows:

$$\boldsymbol{K}_P = \begin{bmatrix} 18.0 & 0 \\ 0 & 19.0 \end{bmatrix}, \quad \boldsymbol{K}_I = \begin{bmatrix} 0.29 & 0 \\ 0 & 0.31 \end{bmatrix}. \tag{2.63}$$

The control result obtained using these PI gains is shown in Figure 2.11. The figure shows, that stabilized system outputs are obtained. However, each closed-loop's tracking property is not good and the cold-water control input vibrates when the system outputs are converged in a steady state. In order to solve the tracking properties, the control result when PI parameters are

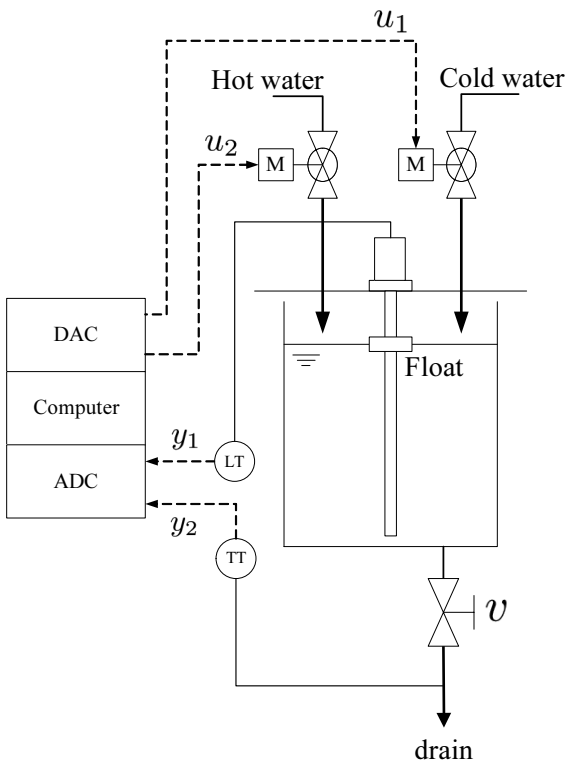

Figure 2.10: Schematic of equipment.

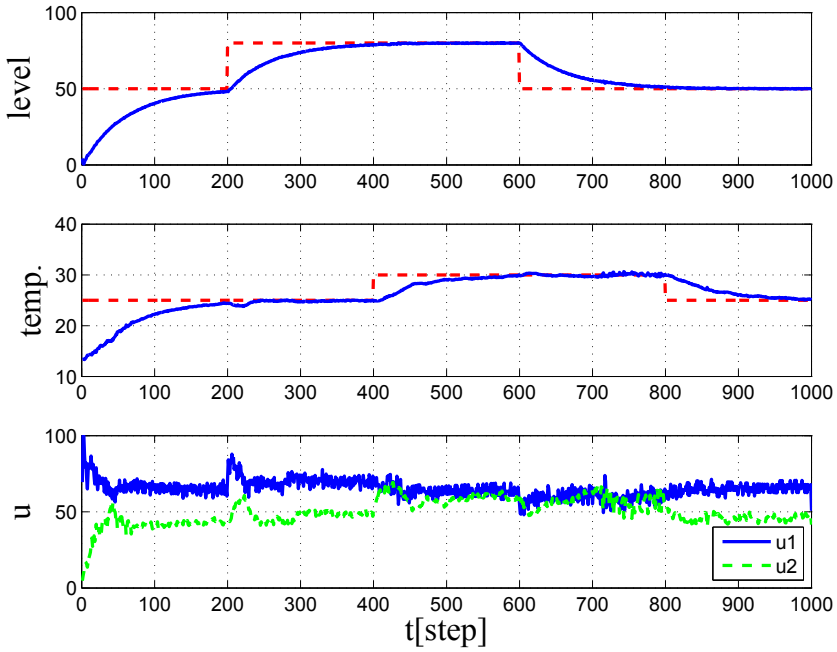

Figure 2.11: Experimental result obtained using multiloop proportional-integral (PI) controller.

doubled is shown in Figure 2.12, where, the doubled PI gains are as follows:

$$\boldsymbol{K}_P = \begin{bmatrix} 36.0 & 0 \\ 0 & 38.0 \end{bmatrix}, \quad \boldsymbol{K}_I = \begin{bmatrix} 0.58 & 0 \\ 0 & 0.62 \end{bmatrix}. \tag{2.64}$$

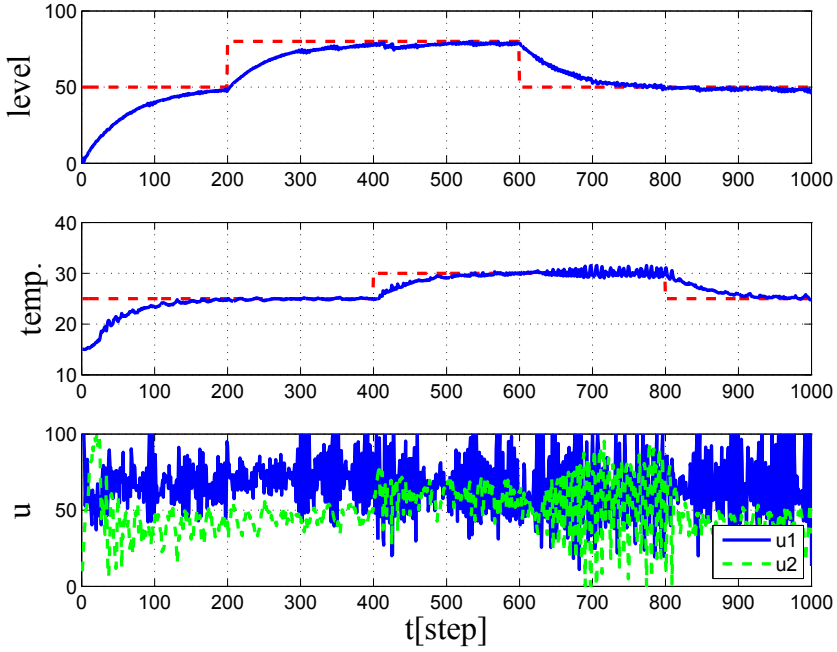

Figure 2.12: Experimental result obtained using multiloop proportional-integral (PI) controller when PI parameters obtained from Figure 2.11 are doubled.

The figure shows that the tracking properties are not improved and that the control input oscillations stronger because each of the input's respective interferences are increased. In order to obtain a stability and good tracking property in the closed-loop system, PI gains are computed by considering each interference via the proposed method. The control result shown in Figure 2.11 is used as the closed-loop data to calculate the PI gains. Moreover, the following user-specified parameters are set based on the prior information of the system:

$$\boldsymbol{\sigma} = \mathrm{diag}[165, 117], \quad \boldsymbol{\delta} = \mathrm{diag}[0, 0],$$
$$\boldsymbol{\Lambda} = \mathrm{diag}[0.001, 0.001], \tag{2.65}$$

and the design polynomial is computed as follows:

$$\boldsymbol{P}(z^{-1}) = \boldsymbol{I} + \begin{bmatrix} -1.9 & 0 \\ 0 & -1.8 \end{bmatrix} z^{-1} + \begin{bmatrix} 0.89 & 0 \\ 0 & 0.84 \end{bmatrix} z^{-2}. \tag{2.66}$$

After these settings, PI gains by the proposed method are calculated as fol-

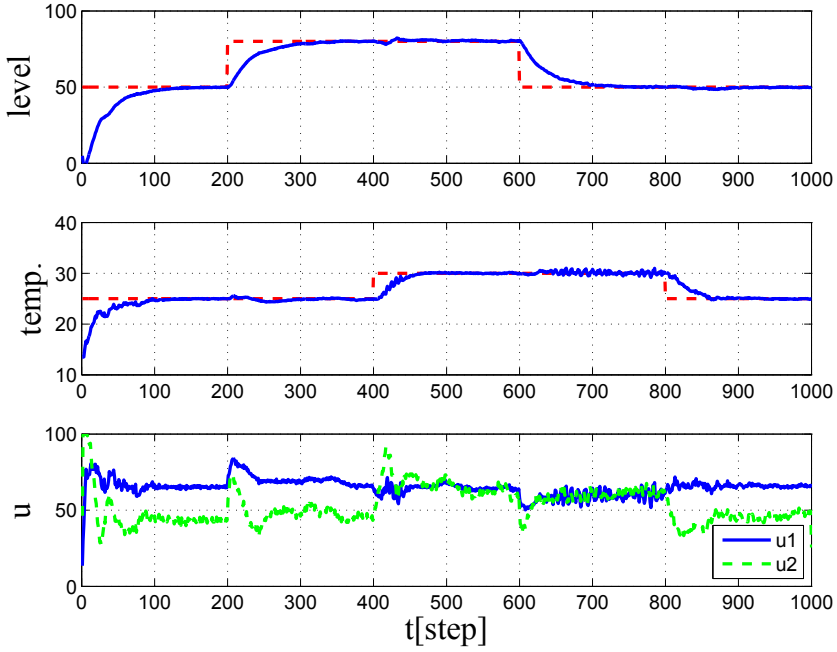

Figure 2.13: Experimental result obtained using proposed method.

lows:

$$\boldsymbol{K}_P = \begin{bmatrix} 3.8 & -8.4 \\ 10.0 & 0.23 \end{bmatrix}, \quad \boldsymbol{K}_I = \begin{bmatrix} 0.12 & -0.26 \\ 0.28 & 0.27 \end{bmatrix}. \tag{2.67}$$

The control result is shown Figure 2.13. From the figure, the tracking properties of the closed-loop systems are improved. Moreover, the oscillations of the control inputs are suppressed by comparing them to the multiloop PI controller. In order to validate the effectiveness of the off-diagonal elements in the PI gains, the PI controller whose off-diagonal elements in (2.67) are removed (2.68) is applied to the system (see Figure 2.14):

$$\boldsymbol{K}_P = \begin{bmatrix} 3.8 & 0 \\ 0 & 0.23 \end{bmatrix}, \quad \boldsymbol{K}_I = \begin{bmatrix} 0.12 & 0 \\ 0 & 0.27 \end{bmatrix}. \tag{2.68}$$

From Figure 2.14, the control results are drastically depleted because of the mutual interference of each of the inputs. Therefore, it is clear that the off-diagonal terms assume a key role to suppress mutual interference.

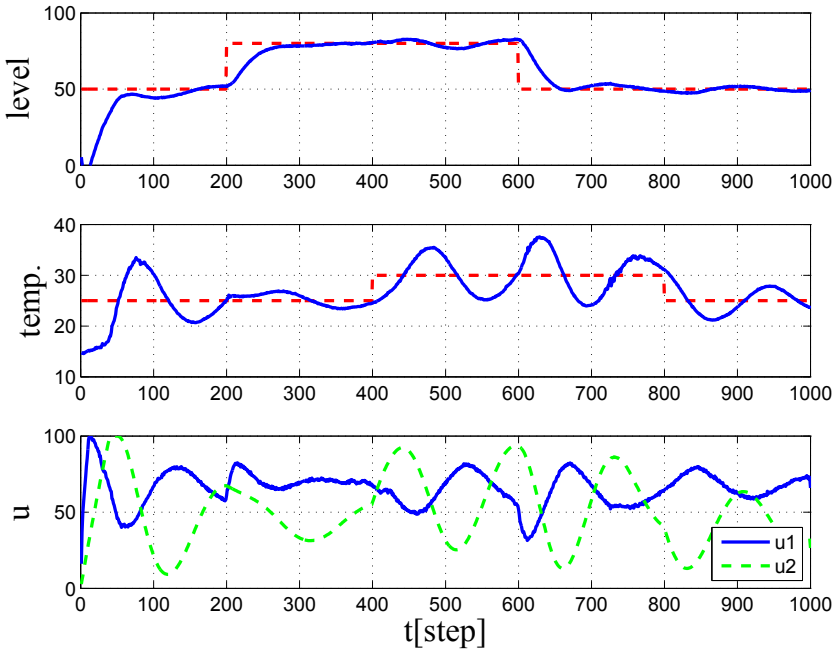

Figure 2.14: Experimental result obtained using proposed method without off-diagonal elements in proportional-integral (PI) gains.

2.6 Conclusions

In this chapter, a data-oriented PID controller design method based on the implicit GMVC without system identification was considered, and the effectiveness of the proposed method was evaluated using a simulation and an experiment in a temperature-control system. Moreover, it was indicated that the GMV-PID controller was easily extended to MIMO systems, and the availability of the multivariable GMV-PID controller was verified through a simulation and an experiment in a level and temperature-control system. The features of the proposed method are summarized as follows:

i) PID gains can be computed using a set of closed-loop data.

ii) The PID control system takes into consideration a relationship between the stability and the tracking property by designing only one adjustable parameter λ.

However, an online tuning method of λ is necessary to maintain a desired control performance at ii), because there is a trade-off relationship between the

tracking property and the stability by adjusting λ. Moreover, in a future work, a relationship among the proposed method, FRIT, and extended fictitious reference iterative tuning (E-FRIT) will be considered.

Appendix 2.A Convert from GMVC to PID Control

The following equation can be obtained by transforming (2.16):

$$C(z^{-1})y(t) + \Delta u(t) - K_I r(t) = 0, \tag{2.69}$$

where

$$C(z^{-1}) = c_0 + c_1 z^{-1} + c_2 z^{-1}, \tag{2.70}$$

$$\left.\begin{aligned} c_0 &= K_P + K_I + K_D \\ c_1 &= -(K_P + 2K_D) \\ c_2 &= K_D \end{aligned}\right\}. \tag{2.71}$$

Also, (2.18) is transformed to the following equation:

$$\frac{F(z^{-1})}{\nu} y(t) + \Delta u(t) - \frac{P(1)}{\nu} r(t) = 0, \tag{2.72}$$

where

$$\nu := G(1) + \lambda. \tag{2.73}$$

By comparing (2.69) and (2.72), the following relationship can be obtained:

$$\begin{cases} C(z^{-1}) = \dfrac{F(z^{-1})}{\nu} \\ K_I = \dfrac{P(1)}{\nu}. \end{cases} \tag{2.74}$$

From the relationship between (2.69) and the upper equation in (2.74), K_I is obtained as the following equation:

$$K_I = \frac{F(1)}{\nu}. \tag{2.75}$$

From the Diophantine equation of (2.5), the following relationship is realized:

$$P(1) = F(1). \tag{2.76}$$

Thus, the lower equation in (2.74) is always realized. Then, PID gains are calculated uniquely only comparing coefficients in the upper equation in (2.74), and the relationship of (2.19) can be obtained.

Chapter 3

Design of Data-oriented Intelligent Controller

3.1 Introduction

As mentioned in the Chapter 2, proportional-integral-derivative (PID) controllers have been applied in a wide range of industrial fields because of their simple structure. However, many actual control objects are nonlinear. Obtaining a desired control performance is difficult with a fixed PID controller when the equilibrium point of the system output changes according to a reference signal change. An effective control method for nonlinear systems is the cerebellar model articulation controller (CMAC) [35], which is a type of neural network (NN). The CMAC-PID controller [39] with a PID tuner constructed using a CMAC (CMAC-PID tuner) is proposed as an extended CMAC method.

CMAC has a mapping structure with layered weight tables. It can adjust its output adaptively for nonlinear systems by summing the weights that are determined from weight tables depending on the value of an input vector. CMAC must correct its weights through learning in order to create its optimal weight tables. Based on a previous CMAC-PID control method, the CMAC-PID tuner is learned in an online manner (called online learning), and the weights are corrected while controlling. However, it becomes impractical for the actual implementation of a CMAC-PID controller because many experiments are required for CMAC learning. Moreover, a large amount of memory

is needed to implement learned weight tables on a computer. Therefore, it is difficult to run the algorithm on general-purpose micro-controllers with a small memory capacity.

In this chapter, in order to solve these problems, an offline learning method using the fictitious reference iterative tuning (FRIT) method [61], which is a data-oriented controller design method, is proposed. In addition, a method that reduces the required memory by using group method of data handling (GMDH) [56, 62] is proposed. According to the proposed method, offline learning of a CMAC-PID tuner using closed-loop data can be performed by employing the fictitious reference signal defined in the FRIT method. The offline learning method using FRIT is called the CMAC-FRIT method [63, 64]. CMAC-FRIT can be treated as an extension of the FRIT method that is normally only applicable to linear systems. Furthermore, a GMDH-PID tuner with properties similar to the learned CMAC-PID tuner can be constructed by describing the CMAC-PID tuner as a nonlinear function using the GMDH network. The GMDH-PID tuner is designed by determining the coefficients, and thus, the required memory is dramatically reduced compared to the CMAC-PID tuner.

In this chapter, the composition of a CMAC-PID controller is first described, and the offline learning method for a CMAC-PID tuner using the FRIT method is explained. Next, a design method for a GMDH-PID controller, by converting from a composed CMAC-PID tuner, is considered. Moreover, the performance of these controllers is compared by applying the Hammerstein model, which is a nonlinear systems. It is shown that these controllers have similar characteristics. It is also shown that the required memory to compose the GMDH-PID controller is reduced compared to the CMAC-PID controller. Finally, the effectiveness of the proposed method is evaluated using an experimental result for a magnetic levitation control system.

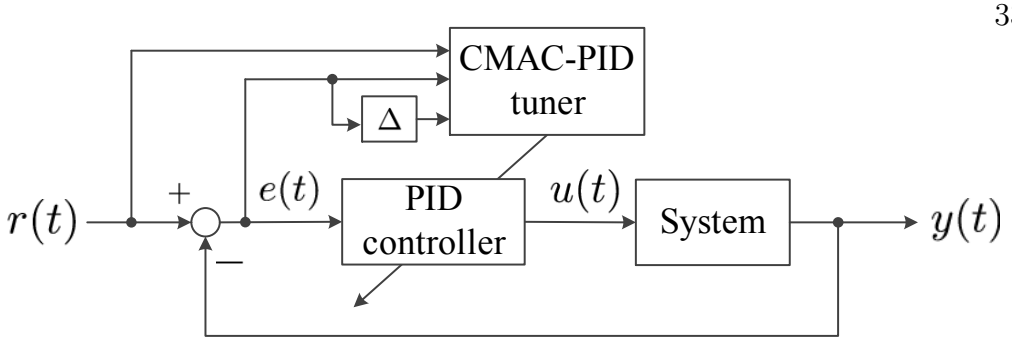

Figure 3.1: Block diagram of proportional-integral-derivative tuner constructed using cerebellar model articulation controller (CMAC-PID).

3.2 Design of CMAC-PID Control System

3.2.1 PID controller design

First, the velocity-type PID control law is considered as follows:

$$\Delta u(t) = K_P(t)\Delta e(t) + K_I(t)e(t) + K_D(t)\Delta^2 e(t). \tag{3.1}$$

Here, $u(t)$ is the control input and $e(t)$ is the control error defined as follows:

$$e(t) := r(t) - y(t). \tag{3.2}$$

In equation (3.2), $r(t)$ and $y(t)$ indicate the reference signal and the system output, respectively. Δ is the differencing operator defined as $\Delta := 1 - z^{-1}$. In addition z^{-1} represents the backward operator defined as $z^{-1}e(t) := e(t-1)$. The CMAC-PID control system configuration is shown in Figure 3.1. The figure shows that the PID gains in each step $K_P(t), K_I(t)$, and $K_D(t)$ are adjusted by the CMAC-PID tuner. The CMAC-PID tuner is constructed as shown in Figure 3.2. From Figure 3.2, the CMAC-PID tuner is constructed using three CMACs ($CMAC_P, CMAC_I$ and $CMAC_D$) with the corresponding inputs $r(t), e(t)$ and $\Delta e(t)$. The CMAC output PID gains depend on these input values.

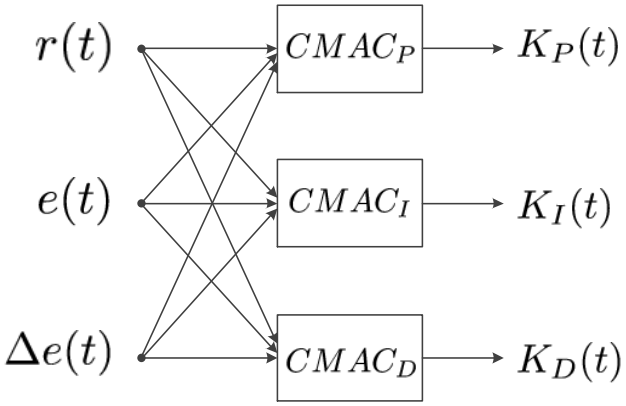

Figure 3.2: Structure of proportional-integral-derivative tuner constructed using cerebellar model articulation controller (CMAC-PID) tuner.

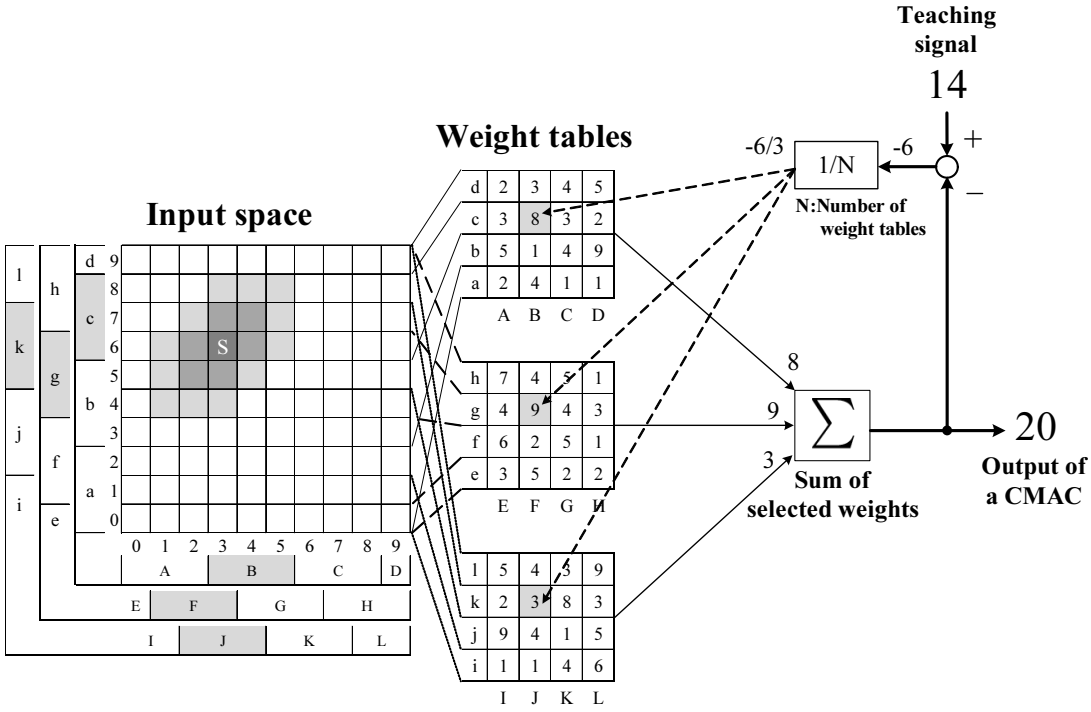

Figure 3.3: Structure of cerebellar model articulation controller (CMAC).

3.2.2 Cerebellar model articulation controller

CMAC is a mathematical model of an information-processing mechanism of the cerebellar cortex in humans. The schematic diagram of a CMAC is shown in Figure 3.3, and a sequence of actions of the CMAC is explained. First, an input vector of $(3, 6)$ is given to the input space, which is then con-

verted into the labeled set $\{B, c\}, \{F, g\}$, and $\{J, k\}$. Based on these labels, a weight table of $8, 9$, and 3 is referenced that outputs 20 as the total. For example, if 14 is the desired output, then the difference between the output and the teaching signal is divided by the number of weight tables, and the value is fed back into the weight tables. That is, learning is performed when $(14 - 20)/3$ is inserted back into the weight tables as a corrective term. After this round of learning, an input vector of $(4, 5)$ is assigned to the input space, and CMAC refers to the label sets $\{B, b\}, \{G, g\}$, and $\{J, k\}$ weight table, i.e. outputted. At this time, the referenced weight tables $\{J, k\}$ are modified from the previous weight. It is possible to consider that the effects of the previous learning sessions have spread to other outputs. This is referred to as the generalization ability of CMAC. Also, the computation time for learning shortens because the CMAC structure shows that it corrects only the weight of a single part in the learning session. Therefore, when compared, it is shown that CMAC is capable of learning faster than a hierarchical NN. The example described above deals with an input space of two dimensions for a simplified illustration; however, in the CMAC-PID controller, a CMAC with three dimensions is used, which is defined as $r(t)$, $e(t)$, and $\Delta e(t)$. This time, the total number of each label, n_1, n_2, and n_3 becomes $n_1 \times n_2 \times n_3$, and the weight table is discretized into N pieces.

CMAC requires learning to obtain its optimal weight tables. In the conventional CMAC-PID control method, CMACs learn these weight tables while controlling. In other words, the CMAC tuner requires online learning. However, the CMACs need many experiments in order to obtain optimal weight tables. This is undesirable for practical situations. Therefore, in order to solve the above problem, a CMAC offline learning method based on the FRIT method is proposed in Section 3.2.4.

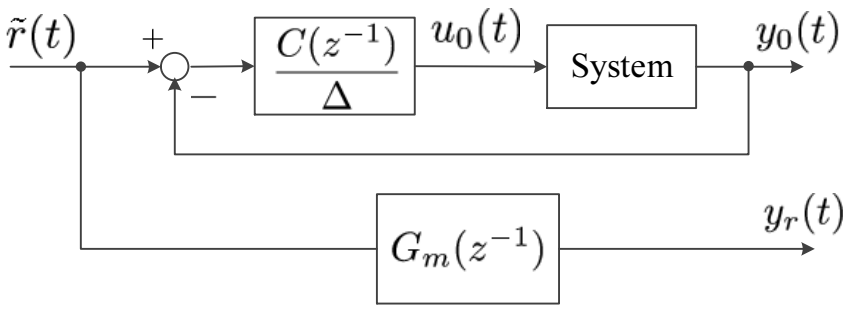

Figure 3.4: Block diagram of fictitious iterative tuning (FRIT) method.

3.2.3 Fictitious reference iterative feedback tuning

FRIT is a method to compute control parameters directly using a fictitious reference signal generated by the closed-loop data. The block diagram of the FRIT method is shown in Figure 3.4. Where, $u_0(t)$ and $y_0(t)$ express the control input and the control output in the closed-loop data, respectively. $C(z^{-1})/\Delta$ is a controller and $C(z^{-1})$ is given as the following equation when the controller is constructed as a PID controller:

$$\begin{aligned} C(z^{-1}) &= \{K_P(t) + K_I(t) + K_D(t)\} \\ &\quad -\{K_P(t) + 2K_D(t)\}z^{-1} + K_D(t)z^{-2}, \qquad (3.3) \\ &= c_0(t) + c_1(t)z^{-1} + c_2(t)z^{-2}. \qquad (3.4) \end{aligned}$$

From Figure 3.4, the fictitious reference signal is derived as follows:

$$\tilde{r}(t) = C^{-1}(z^{-1})\Delta u_0(t) + y_0(t). \qquad (3.5)$$

In addition, a reference model $G_m(z^{-1})$ is designed by a user in advance. The design of $G_m(z^{-1})$ will be explained in Section 3.2.4. In the FRIT method, the output from $G_m(z^{-1})$ corresponding to $\tilde{r}(t)$ is $y_r(t)$, and the control parameters are determined to minimize the squared error between $y_0(t)$ and $y_r(t)$ as follow:

$$J(t) = \frac{1}{2}\{y_0(t) - y_r(t)\}^2 \qquad (3.6)$$

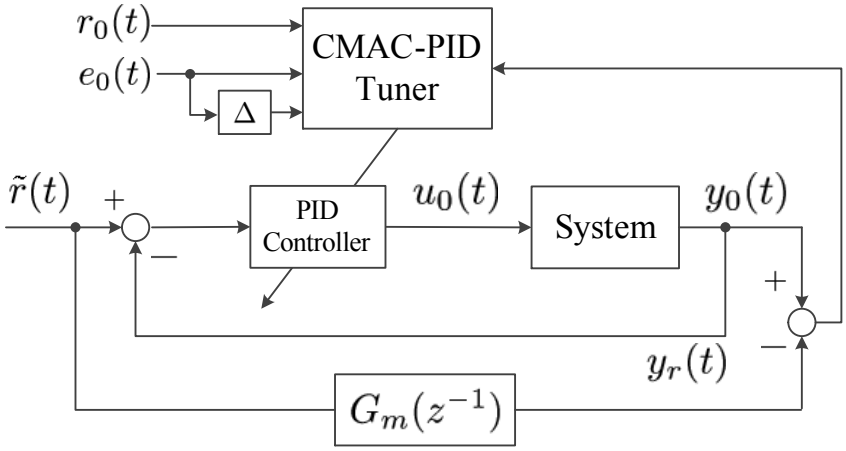

Figure 3.5: Block diagram of cerebellar model articulation controller-fictitious reference iterative tuning (CMAC-FRIT) method.

3.2.4 CMAC-FRIT

CMAC offline learning is implemented by introducing the FRIT method. A block diagram of the offline learning system of the CMAC-PID tuner is shown in Figure 3.5. Henceforth, the offline learning method by the FRIT method will be referred to as the CMAC-FRIT method. In Figure 3.5, $r_0(t)$, $e_0(t)$, $u_0(t)$ and $y_0(t)$ indicate signals in the previously obtained closed-loop data. First, the CMAC-PID tuner outputs the PID gains at t step depending on $r_0(t), e_0(t)$ and $\Delta e_0(t)$. In particular, the PID gains are calculated using the following sum of weights:

$$K_P(t) = \sum_{h=1}^{N} W_{P,h}^{old}, \quad K_I(t) = \sum_{h=1}^{N} W_{I,h}^{old}, \quad K_D(t) = \sum_{h=1}^{N} W_{D,h}^{old}, \tag{3.7}$$

where, $W_{\{P,I,D\},h}^{old}$ is a weight referenced from the h-th table of $CMAC_{\{P,I,D\}}$ in Figure 3.2. Moreover, the reference model $G_m(z^{-1})$ is defined beforehand as the following second-order system:

$$G_m(z^{-1}) = \frac{z^{-1}P(1)}{P(z^{-1})}, \tag{3.8}$$

$$P(z^{-1}) = 1 + p_1 z^{-1} + p_2 z^{-2}, \tag{3.9}$$

$$\left.\begin{aligned} p_1 &= -2\exp(-\tfrac{\rho}{2\mu})\cos(\tfrac{\sqrt{4\mu-1}}{2\mu}\rho) \\ p_2 &= \exp(-\tfrac{\rho}{\mu}) \\ \rho &:= T_s/\sigma \\ \mu &:= 0.25(1-\delta)+0.51\delta \end{aligned}\right\}. \tag{3.10}$$

According to the above reference model, a model that has a desired rise-time and a damping property can be designed by adjusting σ and δ as is the case in Chapter 2. In the CMAC-FRIT method, the weights that are referenced from (3.7) are corrected based on the following gradient method:

$$\left.\begin{aligned} W_{P,h}^{new} &= W_{P,h}^{old} - \eta_P \frac{\partial J(t+1)}{\partial K_P(t)}\frac{1}{N} \\ W_{I,h}^{new} &= W_{I,h}^{old} - \eta_I \frac{\partial J(t+1)}{\partial K_I(t)}\frac{1}{N} \\ W_{D,h}^{new} &= W_{D,h}^{old} - \eta_D \frac{\partial J(t+1)}{\partial K_D(t)}\frac{1}{N} \\ & (h=1,\ldots,N) \end{aligned}\right\}. \tag{3.11}$$

In (3.11), η_P, η_I, and η_D represent the learning coefficients. Moreover, the partial differential of the second term on the right side is expanded as follows:

$$\left.\begin{aligned} \frac{\partial J(t+1)}{\partial K_P(t)} &= \frac{\partial J(t+1)}{\partial y_r(t+1)}\frac{\partial y_r(t+1)}{\partial \tilde{r}(t)}\frac{\partial \tilde{r}(t)}{\partial K_P(t)} \\ \frac{\partial J(t+1)}{\partial K_I(t)} &= \frac{\partial J(t+1)}{\partial y_r(t+1)}\frac{\partial y_r(t+1)}{\partial \tilde{r}(t)}\frac{\partial \tilde{r}(t)}{\partial K_I(t)} \\ \frac{\partial J(t+1)}{\partial K_D(t)} &= \frac{\partial J(t+1)}{\partial y_r(t+1)}\frac{\partial y_r(t+1)}{\partial \tilde{r}(t)}\frac{\partial \tilde{r}(t)}{\partial K_D(t)} \end{aligned}\right\}. \tag{3.12}$$

A more specific expansion of the partial differential in (3.12) is described in the appendix. This offline learning process is iterated until the criterion (3.6) for each instant of steps becomes sufficiently small.

3.3 GMDH-PID Controller Design

In Section 3.2, it is explained that optimal weight tables can be obtained using a set of closed-loop data by the CMAC-FRIT method. However, the CMAC requires great deal of memory to implement the weight tables on a computer. In other words, the CMAC-PID controller cannot be implemented

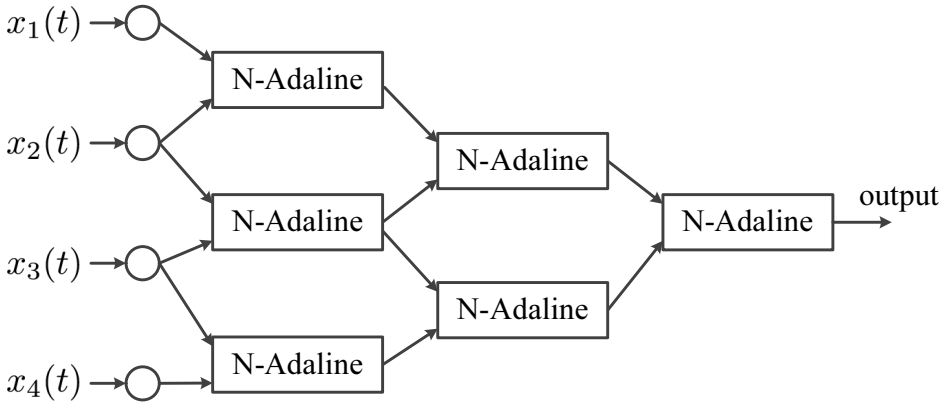

Figure 3.6: Block diagram of group method data handling (GMDH) network.

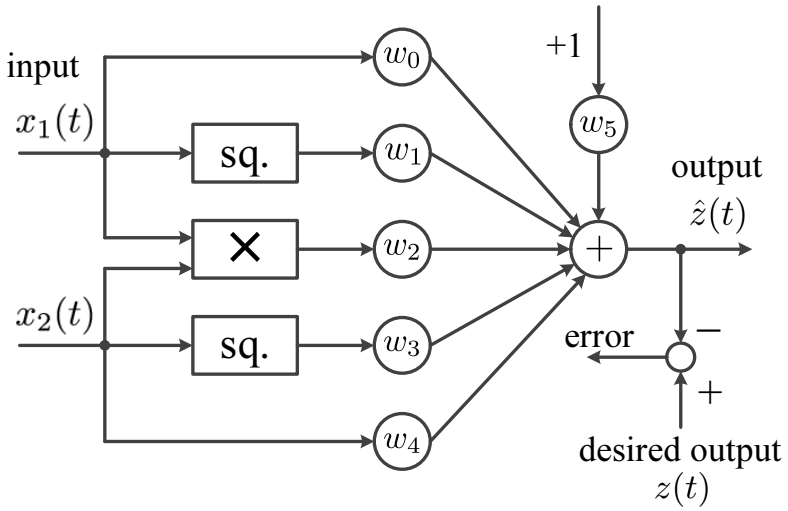

Figure 3.7: Structure of N-Adaline.

on a computer with small memory capacity, such as a micro-controller. In this section, a method in which the required memory is decreased using a GMDH network is considered.

3.3.1 Group method of data handling

A GMDH network is a type of NN (Figure 3.6). This network is implemented by an optimal l-layer combination of N-Adalines. Furthermore, each N-Adaline is configured as shown in Figure 3.7. Here "sq." and "$\times$" represent the square and multiplication of input signals, respectively. In addition, Figure 3.7 shows that the N-Adaline is configured as a unit of two inputs and

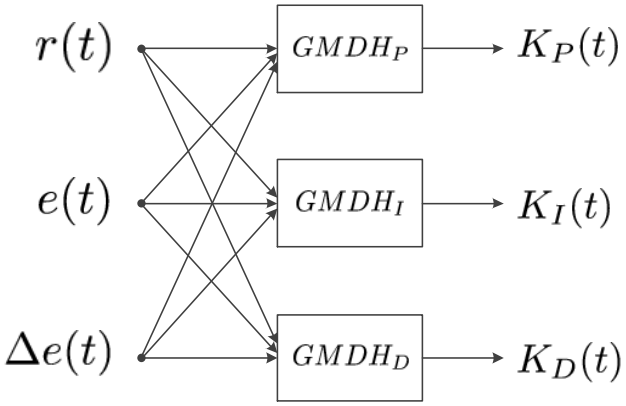

Figure 3.8: Block diagram of group method of data handling-proportional-integral-derivative (GMDH-PID) tuner.

one output. Here, $x_1(t)$ and $x_2(t)$ are the input signals for the N-Adaline, where $z(t)$ is expressed by the following equation:

$$z(t) = w_0 x_1(t) + w_1 x_1^2(t) + w_2 x_1(t) x_2(t) + w_3 x_2^2(t) + w_4 x_2(t) + w_5. \quad (3.13)$$

From these figures, it is considered that the relationship between inputs from $x_1(t)$ to $x_4(t)$ and the output can be expressed by a nonlinear function with a maximum order of 2^l. In the case of Figure 3.6, the I/O relationship is described by a maximum eighth-order nonlinear function.

3.3.2 Design of GMDH-PID tuner based on CMAC-PID tuner

In this section, a description will be given for the design methods of a GMDH-PID tuner that has the performance equivalence of the learned CMAC-PID tuner constructed in Section 3.2. The structure of the GMDH-PID tuner is shown in Figure 3.8. A block diagram of a tuner that outputs $K_P(t)$ is shown in Figure 3.9 as an example. Here, a GMDH-PID tuner based on a CMAC-PID tuner is constructed according to the following steps.

(1) Initial closed-loop data with fully stabilized PID controller are taken.

(2) A CMAC-PID controller is configured, and its weight tables are learned by the CMAC-FRIT.

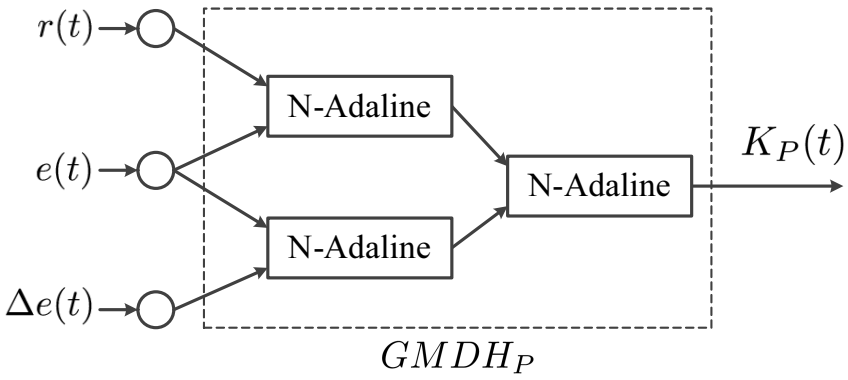

Figure 3.9: Block diagram of one of group method of data handling-proportional-integral-derivative (GMDH-PID) tuners.

(3) In order to configure the GMDH-PID tuner, weight coefficients in all combinations of N-Adalines are determined by an LSM to minimize the output error between the CMAC-PID tuner's output and the constructed GMDH-PID turner's output.

(4) A combination of N-Adalines maximizing a contribution rate of the GMDH network output is determined, and the combination is employed as the GMDH-tuner.

In order to minimize the error, a calculation is performed using an LSM on the error of the outputted PID gains obtained from the N-Adaline's output (PID gain) and CMAC-PID tuner. In addition, an indicator of each N-Adaline's contribution rate is shown below:

$$R^2 = \sum_{t=1}^{M}(\hat{z}(t) - \bar{\hat{z}}(t))^2 / \sum_{t=1}^{M}(z(t) - \bar{z}(t))^2. \tag{3.14}$$

where, M is the number of data in the operational data, and $\bar{\hat{z}}(t)$ and $\bar{z}(t)$ represent the average value of $\hat{z}(t)$ and $z(t)$, respectively. R^2 is sufficiently large; therefore, the input receives N-Adaline output characteristics that are close to the desired properties. Figure 3.10 shows a block diagram of the GMDH-PID control system obtained using the proposed method.

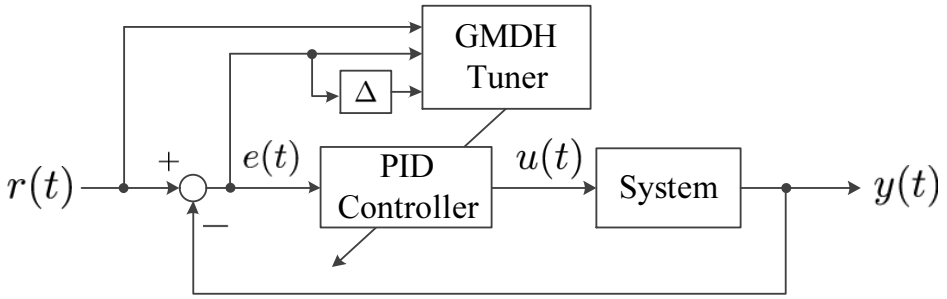

Figure 3.10: Brock diagram of group method of data handling-proportional-integral-derivative (GMDH-PID) control system.

3.4 Numerical Example

In order to verify the effectiveness of the proposed method, the method is applied to the following Hammerstein model:

$$\left.\begin{aligned} y(t) &= 0.6y(t-1) - 0.1y(t-2) + 1.2x(t-1) - 0.1x(t-2) + \xi(t) \\ x(t) &= 1.5u(t) - 1.5u^2(t) + 0.5u^3(t) \end{aligned}\right\}, \quad (3.15)$$

where $\xi(t)$ is white Gaussian noise with zero mean and variance 1.0×10^{-4}. In addition, the sampling time is set as 1[s] in the simulation. Figure 3.11 shows the static property of the system. At each step the reference signal is set by the following equation:

$$r(t) = \begin{cases} 1.0(0 \leq t < 100) \\ 3.0(100 \leq t < 200) \\ 0.5(200 \leq t < 300) \\ 1.5(300 \leq t < 400). \end{cases} \quad (3.16)$$

First, in order to obtain the initial closed-loop data, a simulation is performed with a fixed PID controller. The PID gain values at this time are determined by the CHR method so that the controlled system output around $y(t) = 3.0$ becomes stable. Because, Figure 3.11 shows that this is the highest gain point of the system in these reference value points. The values are shown below:

$$K_P = 0.059, \quad K_I = 0.058, \quad K_D = 0.0038. \quad (3.17)$$

The result obtained from fixed PID controller is shown in Figure 3.12. Figure

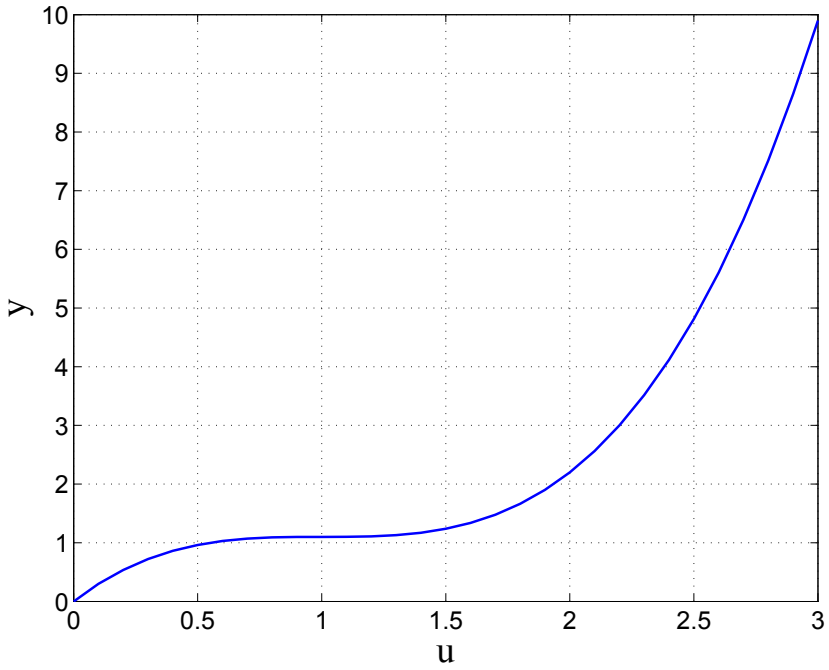

Figure 3.11: Static property of Hammerstein model.

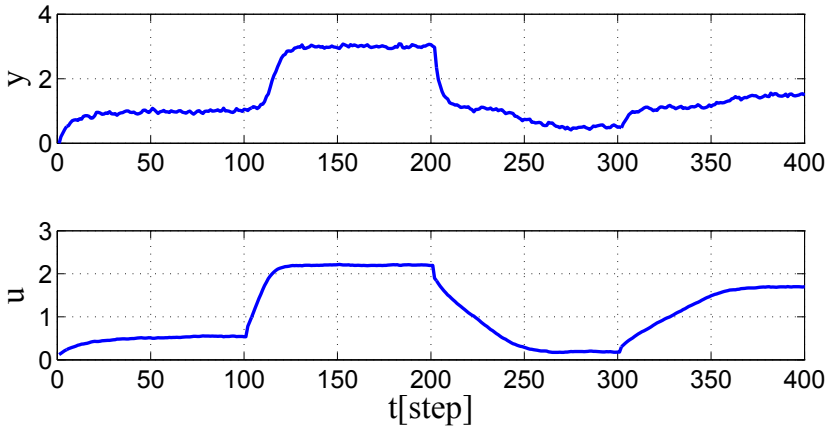

Figure 3.12: Closed-loop data using fixed proportional-integral-derivative (PID) controller.

3.12 shows that the tracking property degrades between 200[step] to 400[step]. It is considered that the PID gains are tuned so that the system output becomes stable when $y(t) = 3.0$; therefore, the selected PID gains become smaller for the equilibrium point at $y(t) = 1.0$ because its system gain are smaller than that at $y(t) = 3.0$. This means that the desired tracking property cannot be obtained at each equilibrium point by using a fixed PID controller

Table 3.1: Design parameters of cerebellar model articulation controller-fictitious reference iterative tuning (CMAC-FRIT).

Number of labels	$n1 = n2 = n3 = 7$
Number of weight tables	$N = 3$
Learning coefficients	$\eta_P = 1.0 \times 10^{-5}$
	$\eta_I = 1.0 \times 10^{-5}$
	$\eta_D = 1.0 \times 10^{-7}$

for a nonlinear system. In Section 3.4.1 and 3.4.2, control results for the CMAC-PID and the GMDH-PID controllers are shown, and their control performances are compared in Section 3.4.3.

3.4.1 Applying CMAC-PID controller

First, a CMAC-PID controller as shown in Figure 3.1 is constructed. The CMAC-PID tuner is learned by the CMAC-FRIT, where the control results shown in Figure 3.12 are used as the closed-loop data for offline learning. The specific design parameters to compose the CMAC-PID controller are shown in Table 3.1. Here, these parameters are adjusted by a trial-and-error process. In addition, the reference model $G_m(z^{-1})$ is designed as follows:

$$G_m(z^{-1}) = \frac{0.109z^{-1}}{1 - 1.34z^{-1} + 0.449z^{-2}}, \tag{3.18}$$

and its coefficients are computed when the rise time is set as $\sigma = 5.0$[s] and the damping property δ is equal to 0. In the CMAC-FRIT method, an CMAC-PID tuner requires an iterated learning in order to obtain optimal weight tables. However, it is necessary to determine when the CMAC-PID tuner has learned enough. Thus, in this simulation, the following integrated squared error (ISE) index is defined:

$$ISE := \sum_{t=1}^{M} \{y_0(t) - y_r(t)\}^2. \tag{3.19}$$

It is considered that the learning process has completed sufficiently when the index is converged on a minimum value. A transition of the ISE index of

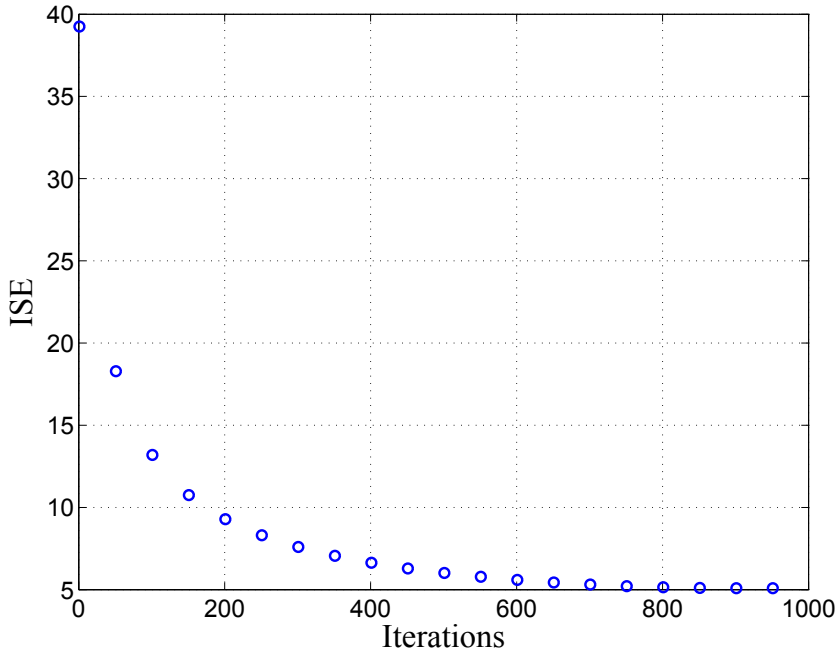

Figure 3.13: Trajectory of integrated squared error (ISE) index.

this simulation is shown in Figure 3.13. From Figure 3.13, it is shown that the ISE index has converged at a minimum value of around 1000 iterations. Therefore, the CMAC-PID controller that learns 1000 iterations is applied to this system. The control result is shown in Figure 3.14. The trajectories of the PID gains are shown in Figure 3.15. Here, the y_m in Figure 3.14 is the output from $G_m(z^{-1})$ that is the input used as the reference signal. From these figures, the desired tracking property can be almost obtained because the system output draws the same trajectory of y_m. Moreover, the PID gains are adaptively tuned at each equilibrium point. However, the output is not improved completely when $r(t) = 0.5$ and $r(t) = 1.5$. This is because the weights obtained by the CMAC-FRIT method fall with in the local minimums when the gradient method is used as a learning method. Therefore, it can be said that the CMAC-PID tuner obtained using the CMAC-FRIT method depends on the initial values of the PID gains in the closed-loop data.

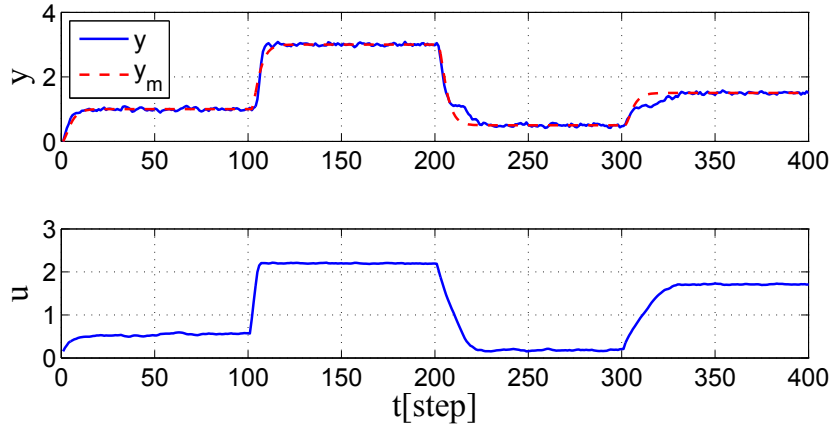

Figure 3.14: Control result obtained using proportional-integral-derivative tuner constructed using cerebellar model articulation controller (CMAC-PID).

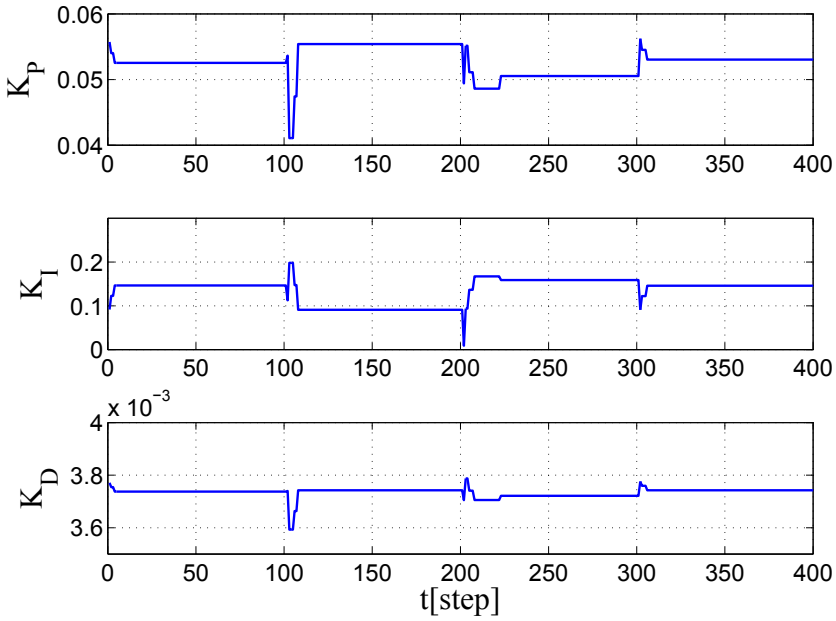

Figure 3.15: Trajectories of proportional-integral-derivative (PID) gains corresponding to Figure 3.14.

3.4.2 Applying GMDH-PID controller

Next, the CMAC-PID controller is converted to a GMDH-PID controller. At this time, the number of GMDH layers is set at $l = 2$. The constructed GMDH-PID tuner structure and the contribution rate are shown in Figure 3.16. The control result obtained using the constructed GMDH-PID

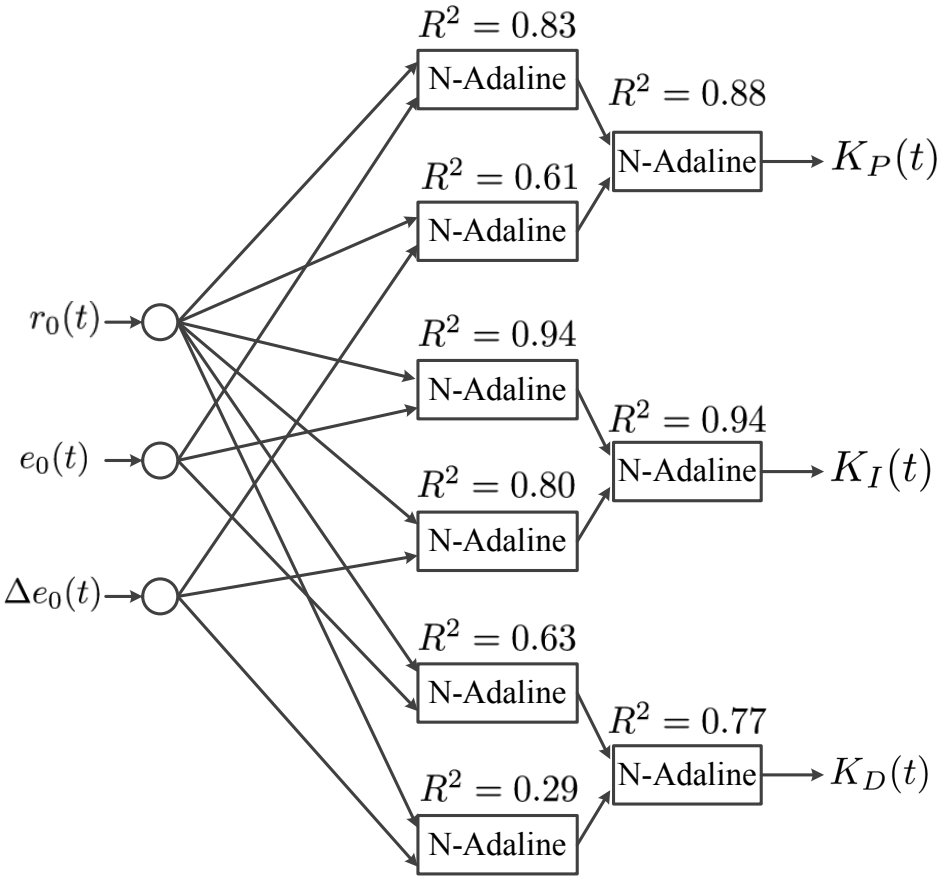

Figure 3.16: Constructed group method of data handling (GMDH) tuners using proposed method.

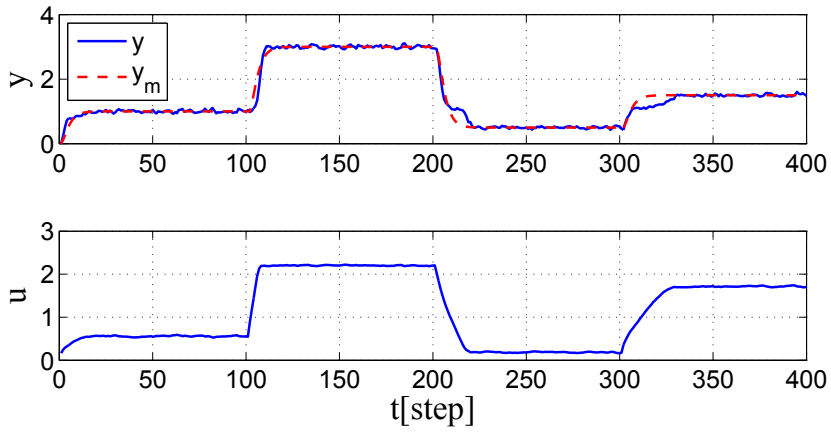

Figure 3.17: Control result obtained using group method of data handling-proportional-integral-derivative (GMDH-PID) tuner.

controller is shown in Figure 3.17. The trajectories of the PID gains at this time are presented in Figure 3.18. From these figures, it is shown that the

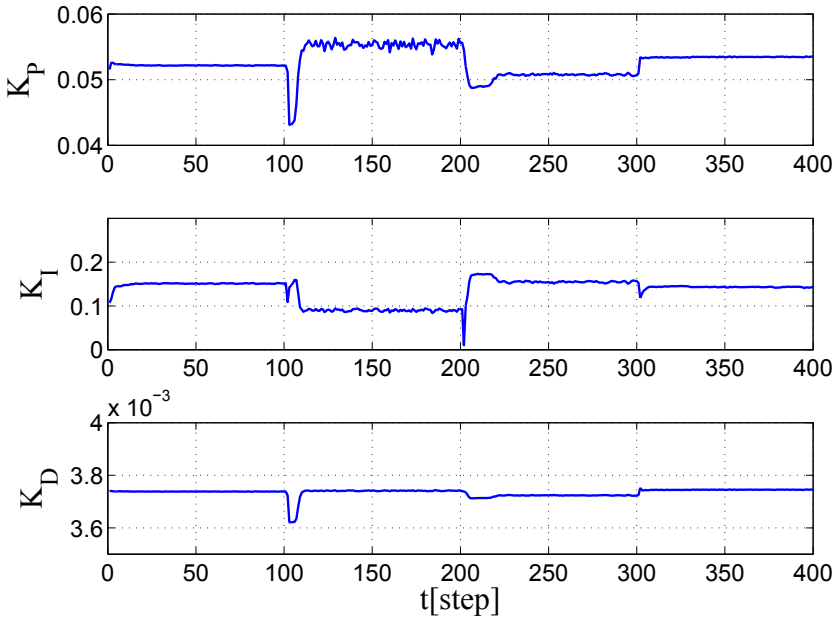

Figure 3.18: Trajectories of proportional-integral-derivative (PID) gains corresponding to Figure 3.17.

PID gains are adjusted appropriately at each equilibrium point; as a result, a good control result is obtained for each reference signal.

3.4.3 Comparison of control performances

To compare the performances of the CMAC-PID and GMDH-PID controllers, the control results are shown in Figure 3.19. Additionally, Figure 3.20 shows a comparison of the trajectories of the PID gains. The comparisons of Figure 3.19 and Figure 3.20 show that performance of the GMDH-PID controller is almost equivalent to that of the CMAC-PID controller. Here, the GMDH-PID controller calculated PID gains are shown to be oscillatory in the steady state. This is because the GMDH is expressed as a nonlinear function, which is believed to have affected the output noise component of the inputs $e(t)$ and $\Delta e(t)$. It is felt that a digital filter should be inserted into the tuner if the oscillation is intensified by noise.

The amount of memory capacity required for the implementation of the CMAC-PID and GMDH-PID controllers were calculated. When the

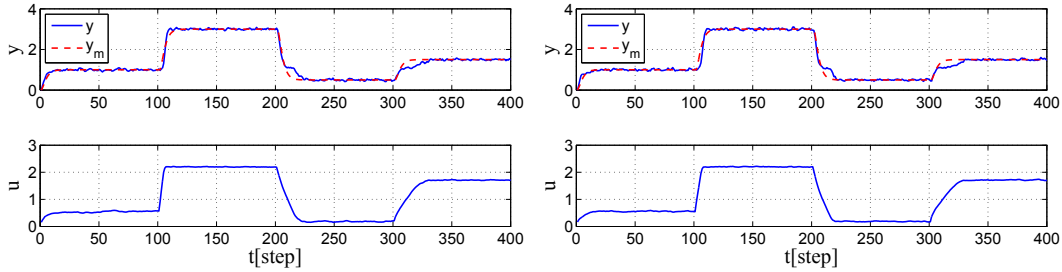

Figure 3.19: Comparison of control results between proportional-integral-derivative tuner constructed using cerebellar model articulation controller (CMAC-PID) (left-hand) and group method of data handling-proportional-integral-derivative (GMDH-PID) controller (right-hand).

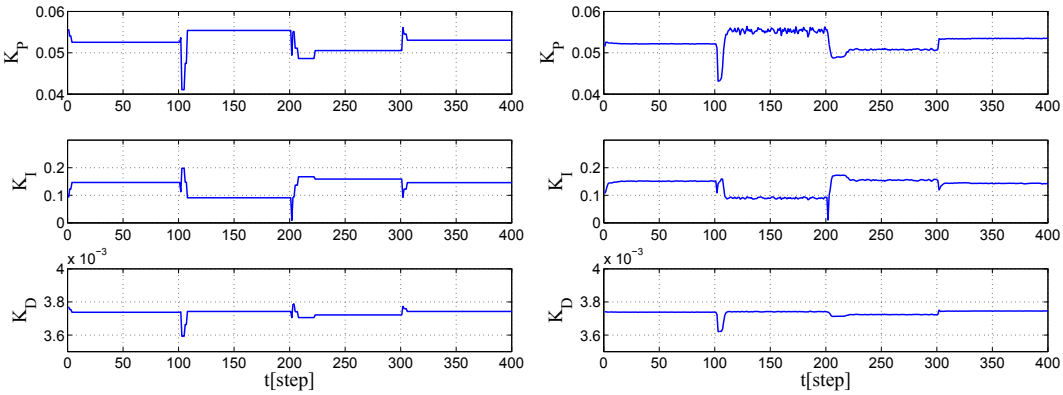

Figure 3.20: Comparison of trajectories of proportional-integral-derivative (PID) gains between PID tuner constructed using cerebellar model articulation controller (CMAC-PID) (left-hand) and group method of data handling-PID (GMDH-PID) tuner (right-hand).

weight coefficients are expressed in a single-precision floating-point type with IEEE754 (32bit), the amount of memory required to store the weights of a CMAC-PID tuner is calculated as:

$$n_1 \times n_2 \times n_3 \times N \times 32[\text{bit}] \times 3(\text{tuners}). \tag{3.20}$$

Therefore, the required memory in the case of this simulation is:

$$7^3 \times 3 \times 32 \times 3 = 98784[\text{bit}] = 12348[\text{B}]. \tag{3.21}$$

On the other hand, the amount of memory required for the GMDH-PID controller is:

$$6(\text{coefficients}) \times 3(\text{N-Adalines}) \times 32[\text{bit}] \times 3(\text{tuners}), \tag{3.22}$$

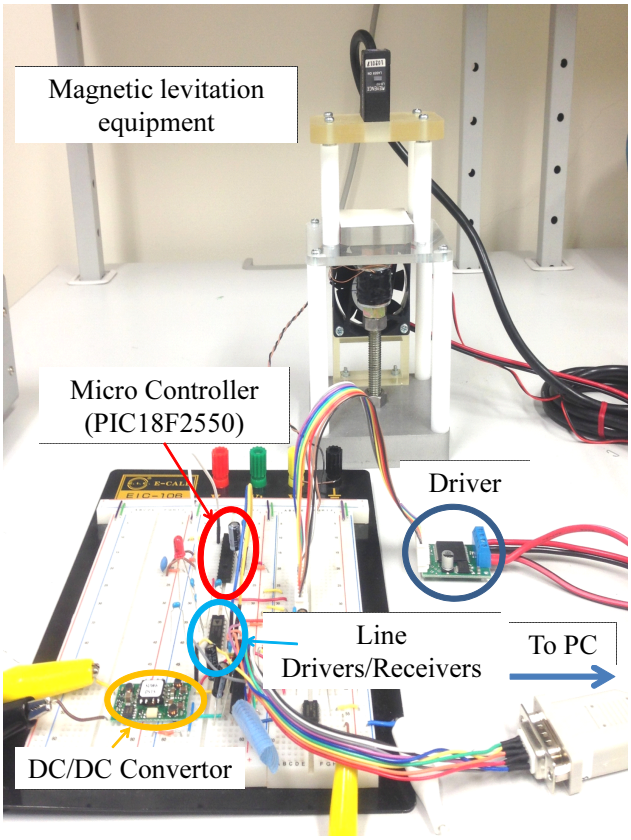

Figure 3.21: Configuration of magnetic levitation device.

then,

$$6 \times 3 \times 32 \times 3 = 1728[\text{bit}] = 216[\text{B}]. \tag{3.23}$$

Therefore in the case of this simulation, it is shown that the amount of memory required is decreased by 98.2% or more.

3.5 Application to Magnetic Levitation Device

In this section, the proposed method is applied to the magnetic levitation device in order to verify the method's usefulness (Figure 3.21). The schematic diagram of the magnetic levitation control system used in the experiment is shown in Figure 3.22. The control system is composed of a magnetic levitation device body, driver, and micro-controller. The magnetic levitation device works as follows. The magnetic force is generated from an electromagnet in response to a current supplied by the driver. Meanwhile, a magnet

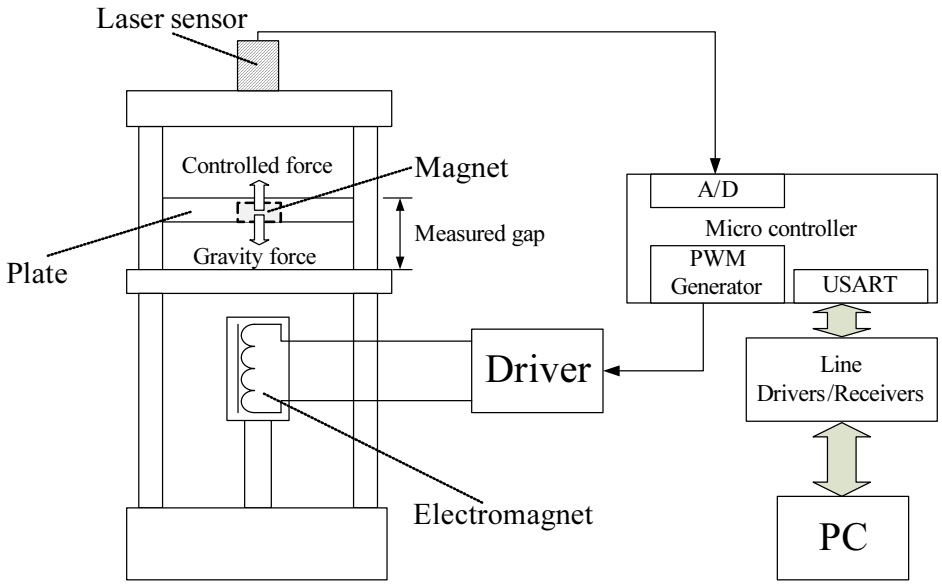

Figure 3.22: Schematic diagram of the magnetic levitation device.

is embedded on the plate above the electromagnet. The plate floats by the repulsive force between the magnets in the plate and electromagnet. The levitation height is measured using a laser sensor at the top of the unit. A voltage signal corresponding to the distance is sent as input to the A/D port on the micro-controller. An algorithm of the proposed method (a GMDH-PID controller converted from a CMAC-PID controller) is programmed on the micro-controller. The GMDH-PID controller computes the input $u(t)$ to the driver as a duty ratio depending on the margin of error between the reference distance $r(t)$ and the current plate level $y(t)$. The PWM signal corresponding to the input is outputted from the PWM generator. In addition, PID gains and the measured data is transferred to the PC from a USART port on the micro-controller. The micro-controller used in this experiment is a general-purpose PIC18F2550 from Microchip Technology Inc., operating at 48MHz from an external clock. The sampling time of the system is set to $T_s = 0.1$[s] taking into account the sampling time of the control system, A/D conversion, controller output calculation, and USART data send time. However, because of the experimental apparatus, the first input of $u(0) = 100$

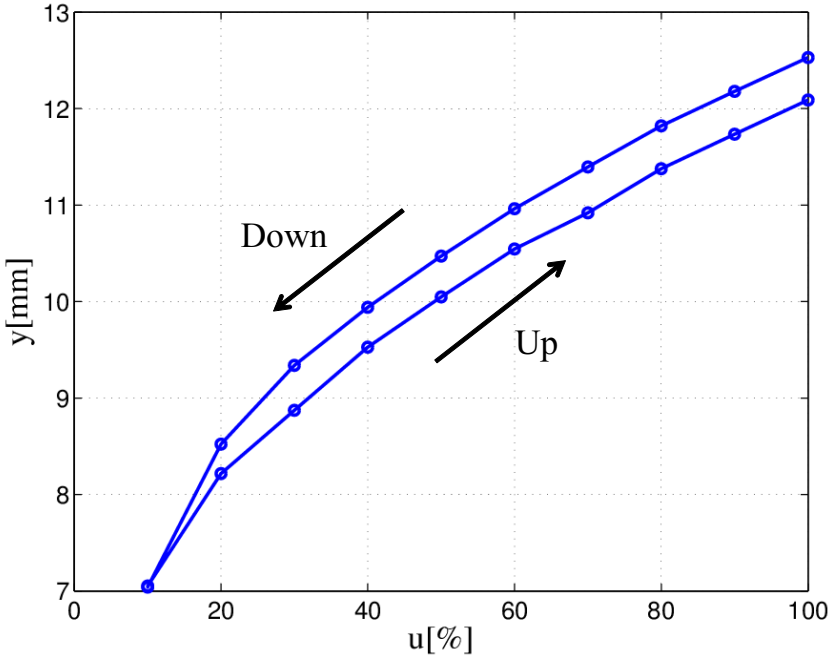

Figure 3.23: Static property of magnetic levitation device.

is obtained when the plate is floating. The static characteristics of the magnetic levitation device are shown in Figure 3.23. From this figure, the gain characteristics of the experimental apparatus are found to be nonlinear, and have hysteresis in the rise and fall of the plate. The reference signal at each time in this experiment is set as follows:

$$r(t) = \begin{cases} 10.0(0 \leq t < 50) \\ 8.5(50 \leq t < 100) \\ 11.0(100 \leq t < 150) \\ 7.5(150 \leq t < 200). \end{cases} \tag{3.24}$$

First, in order to obtain the initial closed-loop data, the fixed PID controller is set with the PID gains as (3.25):

$$K_P = 0.1, \quad K_I = 0.1, \quad K_D = 0.1. \tag{3.25}$$

Because of the nature of the experiment, it is difficult to select initial PID gains that will operate in a stable manner beforehand, therefore, these gains are set by experimental testing. The control results at this time is shown

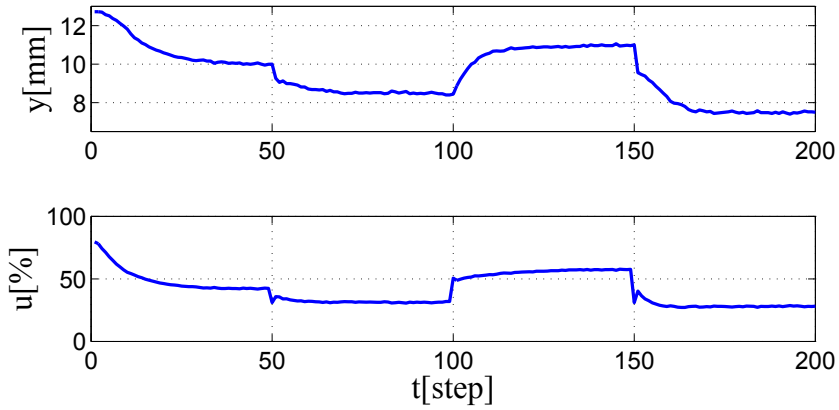

Figure 3.24: Control result for fixed proportional-integral-derivative (PID) controller.

Table 3.2: Design parameters for cerebellar model articulation controller-fictitious reference iterative tuning (CMAC-FRIT).

Number of labels	$n1 = n2 = n3 = 10$
Number of weight tables	$N = 5$
Learning coefficients	$\eta_P = 1.0 \times 10^{-2}$
	$\eta_I = 1.0 \times 10^{-2}$
	$\eta_D = 1.0 \times 10^{-3}$

in Figure 3.24. Then, the proposed method uses the data in Figure 3.24 to create a GMDH-PID controller implemented in the micro-controller. In order to compose the GMDH-PID controller, the CMAC-PID controller learned by using the CMAC-FRIT method is necessary. In this experiment, the design parameters in the CMAC-FRIT are set as shown in Table 3.2 using the trial-and-error. A reference model $G_m(z^{-1})$ is designed as follows:

$$G_m(z^{-1}) = \frac{0.109z^{-1}}{1 - 1.34z^{-1} + 0.449z^{-2}}, \tag{3.26}$$

where, the rise time σ is set as 0.5[s] and the damping property is equal to 0. In order to determine whether the CMAC-PID tuner is sufficiently learned. the ISE index in (3.19) is also introduced. The trajectory of the ISE index is shown in Figure 3.25. From Figure 3.25, the index is converged around 2000 iterations. Therefore, a CMAC-PID controller that learns 2000 iterations is employed for constructing the GMDH-PID controller. The constructed

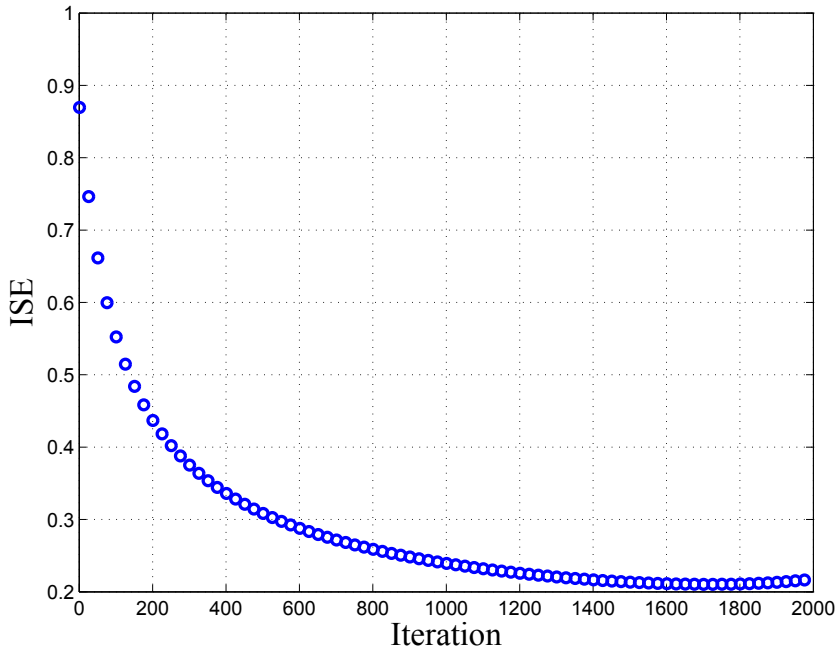

Figure 3.25: Trajectory of integrated squared error (ISE) index.

GMDH-PID tuner from the CMAC-PID tuner is shown in Figure 3.26. The control results obtained using the proposed method is shown in Figure 3.27. The transitory of the PID gains at this time is shown in Figure 3.28. From these figures, it can be seen that the GMDH-PID controller adjusts the PID gains appropriately corresponding to each reference signal and therefore the responsiveness is greatly improved when compared to the fixed PID controller. Here, the amount of the memory required for implementing the CMAC-PID and GMDH-PID controllers are calculated. When the weighting factor is expressed as a single-precision floating-point type with IEEE754 (32bit), the amount of memory required to store the weight of the CMAC-PID tuner is shown as:

$$10^3 \times 5 \times 32 \times 3 = 480000[\text{bit}] = 60000[\text{B}]. \tag{3.27}$$

On the other hand, the GMDH-PID controller is:

$$10 \times 3 \times 32 \times 3 = 1728[\text{bit}] = 216[\text{B}]. \tag{3.28}$$

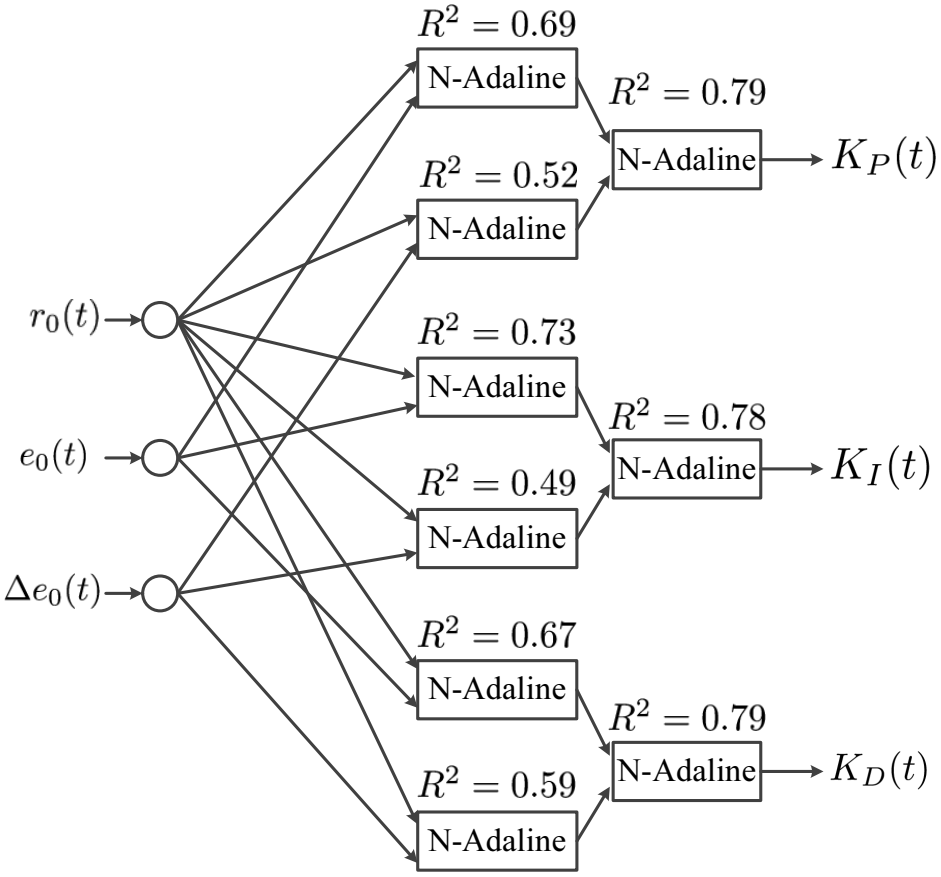

Figure 3.26: Constructed group method of data handling (GMDH) tuner using proposed method.

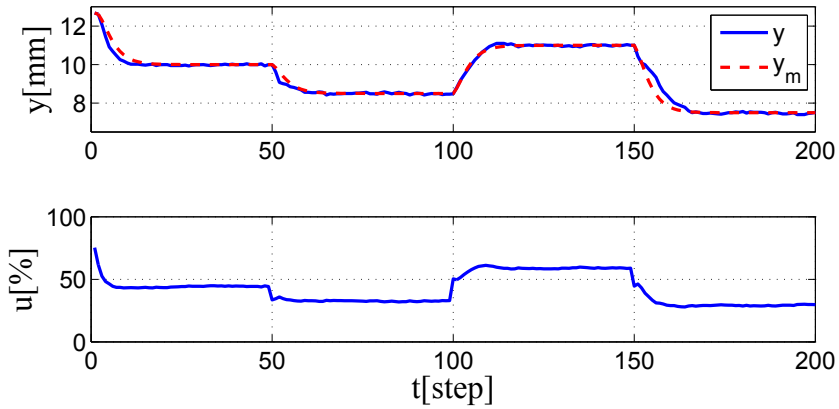

Figure 3.27: Control result obtained using proposed method.

Therefore, this simulation shows that the amount of required memory is decreased by 99.6% or more. Moreover, because the maximum amount of data memory in the micro-controller used in this experiment is 2048[B], imple-

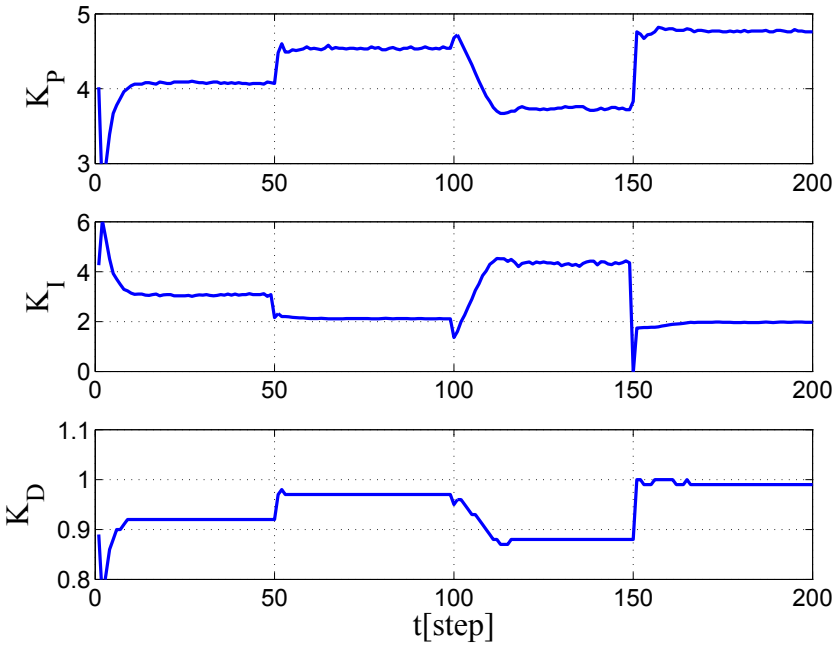

Figure 3.28: Trajectories of proportional-integral-derivative (PID) gains corresponding to Figure 3.27.

menting the CMAC-PID would have been physically impossible.

3.6 Conclusions

This section discussed a learning method for a CMAC-PID controller based on a set of closed-loop data, and its implementation using a GMDH-PID controller. In this method, a CMAC-PID tuner is first configured based on the FRIT method (the CMAC-FRIT method). Then, a GMDH-PID tuner, which is a combination of N-Adalines, is constructed from the CMAC-PID tuner with similar performance. According to the proposed method, a CMAC-PID tuner can learn in an offline manner. In addition, the amount of the required memory is decreased by converting a learned CMAC-PID tuner to a GMDH-tuner. The proposed GMDH-PID controller can be implemented on a computer with a small memory capacity. Furthermore, the effectiveness of the proposed method is verified through a simulation and an applied magnetic levitation device.

Appendix 3.A Expansion of Partial Differentials in (3.12)

The partial differentials in (3.12) are expanded as follows:

$$\begin{aligned}\frac{\partial J(t+1)}{\partial K_P(t)} &= -\Gamma(t+1)[\{K_I(t)-K_D(t)\}\tilde{e}(t-1)\\ &\quad +K_D(t)\tilde{e}(t-2)-\Delta u_0(t)], \qquad (3.29)\\ \frac{\partial J(t+1)}{\partial K_I(t)} &= \Gamma(t+1)[\{K_P(t)+2K_D(t)\}\tilde{e}(t-1)\\ &\quad -K_D(t)\tilde{e}(t-2)+\Delta u_0(t)], \qquad (3.30)\\ \frac{\partial J(t+1)}{\partial K_D(t)} &= -\Gamma(t+1)[\{K_P(t)+2K_I(t)\}\tilde{e}(t-1)\\ &\quad -\{K_P(t)+K_I(t)\}\tilde{e}(t-2)-\Delta u_0(t)], \qquad (3.31)\end{aligned}$$

where,

$$\tilde{e}(t) := \tilde{r}(t) - y_0(t) \qquad (3.32)$$

and $\Gamma(t+1)$ is given as follows:

$$\Gamma(t+1) := \frac{\varepsilon(t+1)P(1)}{\{K_P(t)+K_I(t)+K_D(t)\}^2}, \qquad (3.33)$$

$$\varepsilon(t) := y_0(t) - y_r(t). \qquad (3.34)$$

Chapter 4

Design of Performance-driven Controller

4.1 Introduction

In this section, a one-parameter tuning method for performance-driven control system, which unifies a control-performance assessment and a controller redesign, is presented. In the previous chapters, a discussion focusing on direct controller design using closed-loop data was considered. However, in a real-time system, characteristics vary every hour. Sometimes, a controller is in an unstable state when a system change not included in the closed-loop data takes place. Moreover, in most chemical processes, a high priority is put on control performance in a steady state from the view point of a stable supply of products and energy saving. Therefore, it is important to consider a design scheme for a performance-driven controller that tunes a controller based on constantly evaluated control performance.

According to a one-parameter tuning method, a controller based on pole-assignment strategy is first designed. The pole-assignment controller determines its control parameters in order to put the poles of the closed-loop system to arbitrary points. However, in this method, poles are put in indirectly by determining control parameters based on a Diophantine equation. According to the Diophantine equation, control parameters are determined by comparing coefficients between a design polynomial and a polynomial of the closed-loop system. Here, the design polynomial includes a user-specified

parameter δ related to the rise property. The one-parameter tuning method is a method to adjust δ in order to meet the desired control performance constantly. Although many control performance assessment methods described by minimum variance (MV) index [44] have been proposed, in this method, the control performance is evaluated based on the ratio between the variance of the system output and the variance of the differential input. The proposed one-parameter tuning method has the following features:

i) The desired control performance can be obtained by adjusting only one user-specified parameter δ.

ii) It is not necessary to identify the system parameters recursively. The user-specified parameter δ becomes a measure whether the system parameters need to be reidentified.

iii) The desired control performance is maintained by adjusting δ regardless of the accuracy of the system identification.

The proposed method reidentifies a system model and redesigns a controller when the system property is drastically changed for any reason and it is difficult to maintain the control performance by adjusting δ. However, according to the proposed method, the number of times that system identification is performed is dramatically reduced; thus, a more applicative performance-driven controller can be designed.

In this chapter, the design method of a pole-assignment controller with adjustable parameter δ is first explained. Next, the trade-off relationship according to δ change between the variance of the system output and the variance of the differential input is discussed. Then, the δ adjusting method is discussed based on the trade-off relationship and the proposed one-parameter tuning algorithm. The effectiveness of the proposed method is evaluated by a simulation for a linear time-variant system. Finally, the usefulness of the

proposed method is further verified by employing it in a weigh feeder, a food process system.

4.2 Design of Performance-driven Control System

4.2.1 System description

Most real-time systems such as chemical plants and petroleum processes, have a higher-order lag factor and large time-delay. As parameters in a system model increase, uncertain factors also increase, and results of a control system derived by the system model becomes unstable. To mitigate this problem, on factory floors where processing systems are operated, the control object often approximates system parameters using a first-order with time delay system. Thus, the system can be described as:

$$G(s) = \frac{K}{1+Ts} e^{-Ls}, \tag{4.1}$$

where K, T, and L denote the system gain, time constant, and time lag, respectively. A CARIMA model corresponding to (4.1) with a zero-order-holder and model error can be represented as follows:

$$A(z^{-1})y(t) = z^{-(k+1)}B(z^{-1})u(t) + \frac{\xi(t)}{\Delta}, \tag{4.2}$$

$$\left.\begin{aligned} A(z^{-1}) &= 1 + a_1 z^{-1} \\ B(z^{-1}) &= b_0 + b_1 z^{-1} \end{aligned}\right\}. \tag{4.3}$$

In (4.2), $u(t)$ and $y(t)$ represent the control input and the system output, k is given by the maximum integer less than L/T_s (where T_s denotes the sampling time), and $\xi(t)$ is the white Gaussian noise with zero mean and variance σ^2. Here, z^{-1} denotes the backward operator that implies $z^{-1}y(t) = y(t-1)$ and Δ is the differencing operator given by $\Delta := 1 - z^{-1}$. Note that a_1, b_0, and b_1 are system parameters of discrete-time models and are computed by the

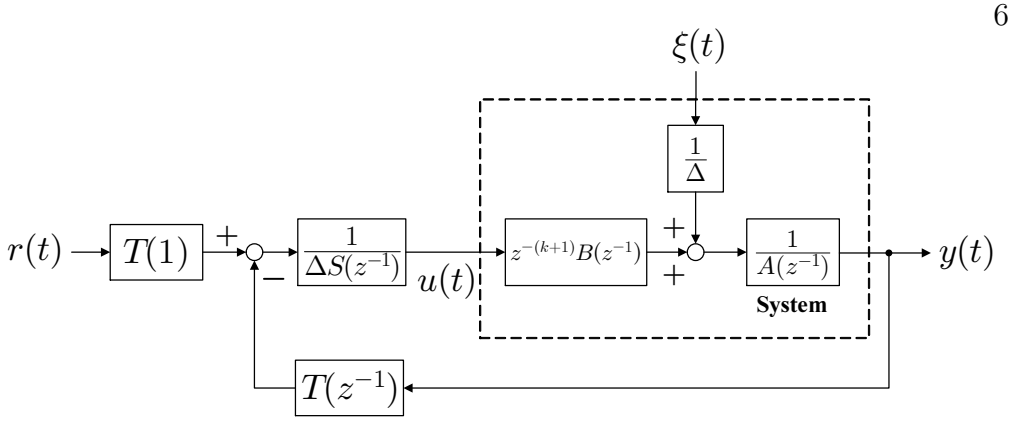

Figure 4.1: Block diagram of control system constructed by (4.2) and (4.8).

following equations that are given by the z-transform:

$$a_1 = -\exp\left(-\frac{Ts}{T}\right), \tag{4.4}$$

$$b_0 = K\left\{1-\exp\left(-\frac{Ts-L_0}{T}\right)\right\}, \tag{4.5}$$

$$b_1 = K\left\{\left(\exp(-\frac{Ts-L_0}{T}\right)-\exp\left(-\frac{Ts}{T}\right)\right\}, \tag{4.6}$$

$$L_0 := L \mod T_s. \tag{4.7}$$

4.2.2 Design of pole-assignment control system

In this section, a controller having only one adjustable parameters is designed for the system described in (4.2). The controller is designed based on the pole-assignment strategy. The following expression is introduced as the control law [57]:

$$T(z^{-1})y(t) + S(z^{-1})\Delta u(t) - T(1)r(t) = 0 \tag{4.8}$$

where $r(t)$ is the reference signal and $T(z^{-1})$ and $S(z^{-1})$ represent polynomials of the controller. The block diagram constructed by (4.2) and (4.8) is shown in Figure 4.1. From this figure, the I/O relationship of the closed-loop system

can be described as:

$$y(t) = \frac{z^{-(k+1)}B(z^{-1})T(1)}{\Delta A(z^{-1})S(z^{-1}) + z^{-(k+1)}B(z^{-1})T(z^{-1})}r(t) + \frac{S(z^{-1})}{\Delta A(z^{-1})S(z^{-1}) + z^{-(k+1)}B(z^{-1})T(z^{-1})}\xi(t). \quad (4.9)$$

From (4.9), $S(z^{-1})$ and $T(z^{-1})$ are obtained by solving the Diophantine equation given by:

$$P(z^{-1}) = \Delta A(z^{-1})S(z^{-1}) + z^{-(k+1)}B(z^{-1})T(z^{-1}), \quad (4.10)$$

where

$$\left.\begin{aligned} S(z^{-1}) &= 1 + s_1 z^{-1} + \cdots + s_{k+1} z^{-(k+1)} \\ T(z^{-1}) &= t_0 + t_1 z^{-1} \end{aligned}\right\}, \quad (4.11)$$

where $P(z^{-1})$ is the desired characteristic polynomial in the control system and is given by:

$$P(z^{-1}) = 1 + p_1 z^{-1} + p_2 z^{-2}, \quad (4.12)$$

where

$$\left.\begin{aligned} p_1 &= -2e^{-2\rho} \\ p_2 &= e^{-4\rho} \\ \rho &= T_s/\sigma \end{aligned}\right\} \quad (4.13)$$

and

$$\sigma = \frac{T + L}{\delta}. \quad (4.14)$$

In (4.14), δ is the user-specified adjustable parameter. It can regulate the sensitivity of the controller by changing δ.

The desired control performance can be maintained by adjusting δ based on the control-performance assessment. In Section 4.3, an approach that unifies the control-performance assessment and the controller design using only the one user-specified parameter δ is proposed, and is referred to as the one-parameter tuning method. Figure 4.2 shows a conceptual diagram of the

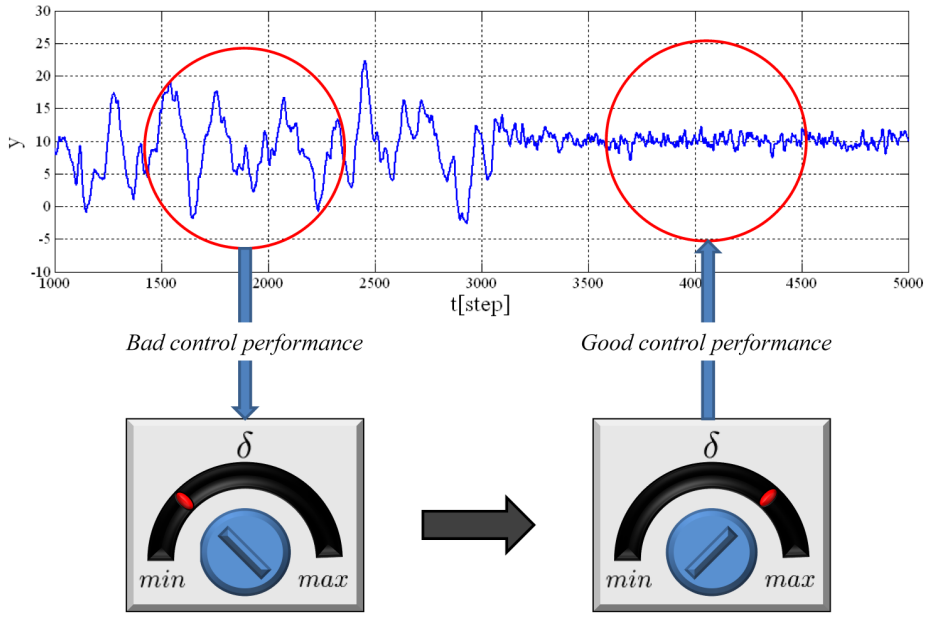

Figure 4.2: Conceptual figure of one-parameter tuning.

proposed method. In Figure 4.2, the control performance is evaluated while controlling. The control performance is defined with the aid of the variance of the control error and the variance of the differential control input. A more detailed definition is provided in Section 4.3. If the control performance is evaluated with an undesired control performance, the user-specified parameter δ is adjusted to improve the control performance. According to this method, a desired control performance is maintained by adjusting only δ. Moreover, a constant system identification is not required in the proposed control system.

4.3 One-parameter Tuning Method

4.3.1 Control-performance assessment

First, the control performance, which is defined with the aid of the variance of the control error and the variance of the differential control input, is considered. The closed-loop system in Figure 4.1 includes the controller in (4.8) and the trade-off curve between $E[e(t)^2]$ (where $e(t) := r(t) - y(t)$) and $E[\{\Delta u(t)\}^2]$ with δ changing as shown in Figure 4.3. From the figure, it is

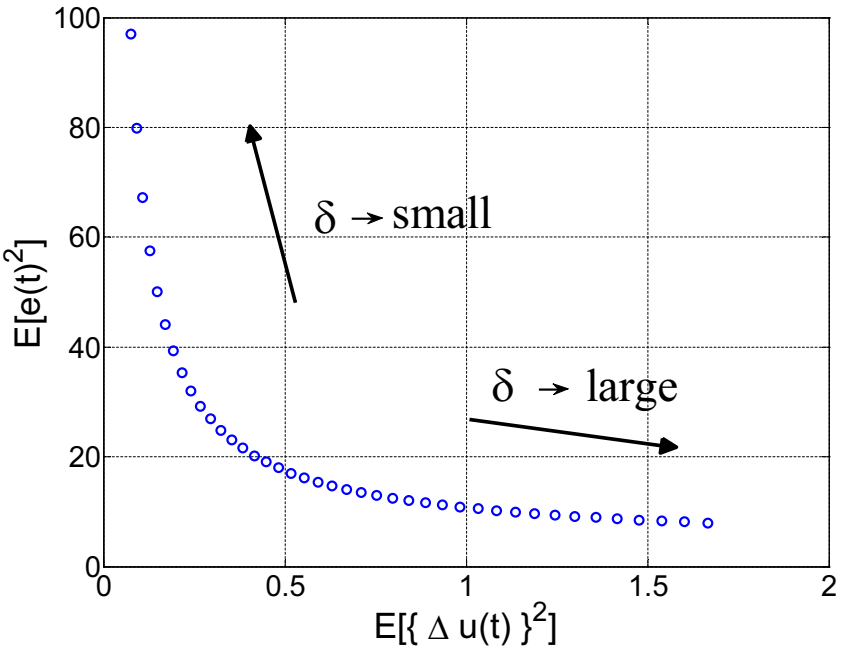

Figure 4.3: Control performance trade-off curve to indicate effect of δ.

seen that the variance of the control error and the variance of the differential control input have a trade-off relationship. Therefore, while the system is in a stable state, the control-performance evaluation is done using $E[e^2(t)]$ and $E[\{\Delta u(t)\}^2]$ as in (4.15):

$$J = \frac{E[e^2(t)]}{E[\{\Delta u(t)\}^2]}. \tag{4.15}$$

In this method, it is assumed that the system has approved ergodicity; thus, in practical operation, (4.15) is computed based on the following unbiased estimate of variances:

$$J(t) = \frac{\dfrac{1}{M-1}\displaystyle\sum_{\tau=t-M}^{t} e^2(\tau)}{\dfrac{1}{M-1}\displaystyle\sum_{\tau=t-M}^{t} \Delta u^2(\tau)}, \tag{4.16}$$

where M expresses a number of data and is set by an operator.

4.3.2 One-parameter tuning

From the above relationship, δ adjusts itself to a larger values if $J(t)$ is larger than γK^2. If $J(t)$ is smaller than K^2/γ, it adjusts itself to smaller value. The control-performance assessment and the controller redesign are integrated by adjusting δ. The δ updating rule is shown in the following equations.

i) $J(t) > \gamma K^2$

$$\delta \leftarrow \delta + \delta_{\Delta}, \tag{4.17}$$

ii) $J(t) < K^2/\gamma$

$$\delta \leftarrow \delta - \delta_{\Delta}, \tag{4.18}$$

iii) $K^2/\gamma \leq J(t) \leq \gamma K^2$

$$\delta \text{ is not changed,}$$

where, γ is a design parameter set as a dead zone and δ_{Δ} is the adjustment width of δ. The proposed method has the advantage of maintaining good control performance by only having to adjust δ for any variation of the system parameters. However, if the system drastically changes, a desired control performance cannot be maintained by changing δ. Therefore, in this method, the range of δ movement is specified and the system parameters are reidentified whenever δ falls out of the predetermined range. The upper and lower limits of δ are represented by δ_{upper} and δ_{lower}, respectively. These parameters are arbitrarily set by the operators. Thus, system identification is not needed while δ is within this specified range. A block diagram of the proposed method is shown in Figure 4.4. In Figure 4.4, the Estimator is identifying system parameters while controlling. The Estimator first calculates the discrete system parameters using the following weighted recursive least squares

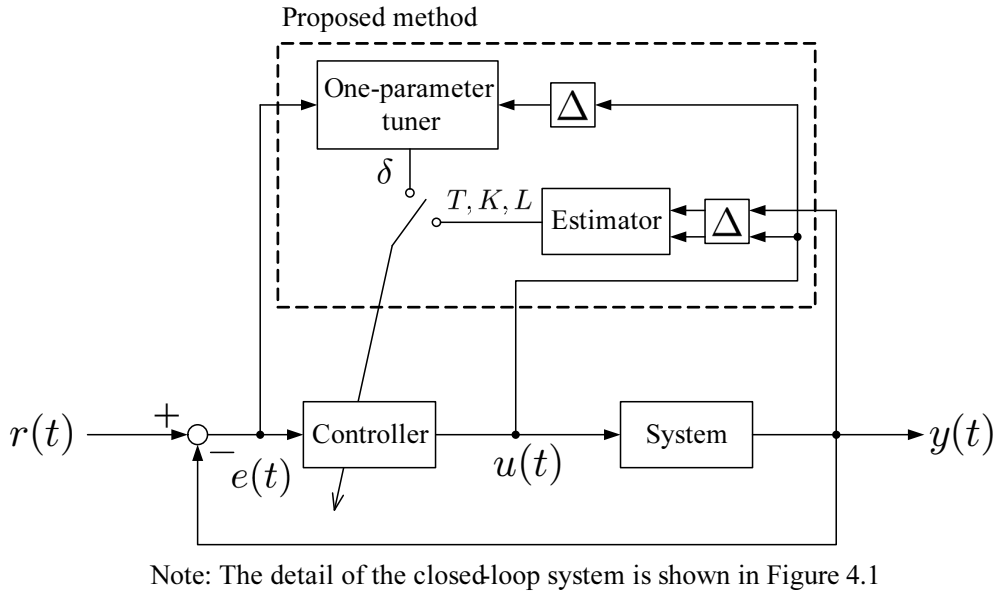

Figure 4.4: Block diagram of proposed method.

(WRLS) method [65, 66]:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \frac{\Gamma(t-1)\psi(t-1)}{\alpha + \psi^T(t-1)\Gamma(t-1)\psi(t-1)}\varepsilon(t) \tag{4.19}$$

and

$$\Gamma(t) = \frac{1}{\alpha}\left[\Gamma(t-1) - \frac{\Gamma(t-1)\psi(t-1)\psi^T(t-1)\Gamma(t-1)}{\alpha + \psi^T(t-1)\Gamma(t-1)\psi(t-1)}\right], \tag{4.20}$$

where

$$\varepsilon(t) = \Delta y(t) - \hat{\theta}^T(t-1)\psi(t-1) \tag{4.21}$$

and

$$\psi(t) := [-\Delta y(t), \Delta u(t-k), \Delta u(t-k-1)]^T. \tag{4.22}$$

Here, α is a forgetting factor and is commonly set between 0.95 and 1.0 [57], and k is the estimated time delay in the discrete time model. If a fluctuating band of k can be estimated in advance, then k can be set as the minimum value of a fluctuating band, otherwise k is set as zero. The computation of (4.19) affords:

$$\hat{\theta}(t) := [\hat{a}_1(t), \hat{b}_0(t), \hat{b}_1(t)]^T \tag{4.23}$$

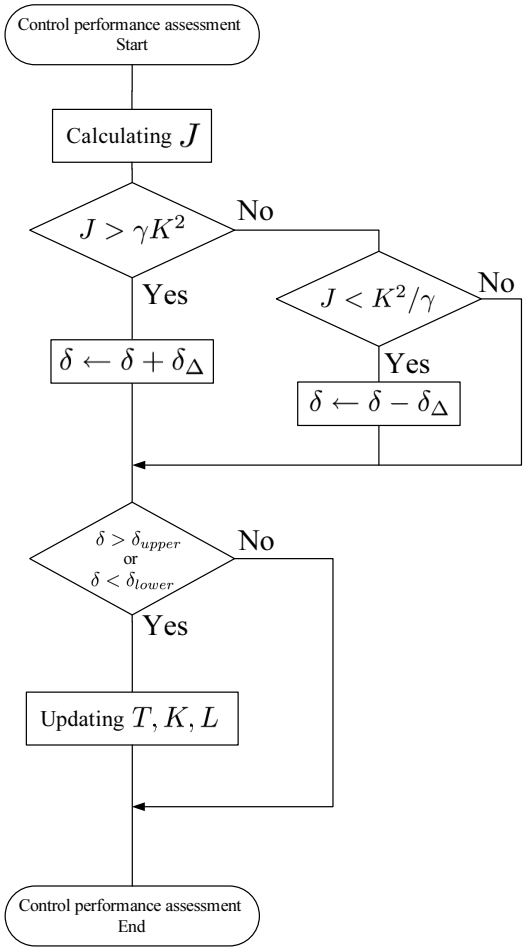

Figure 4.5: Flow chart of proposed method algorithm.

If δ falls out of the specified range δ_{upper} and δ_{lower}, then the system parameters in (4.1) are updated by the following equations [67]:

$$K \leftarrow \frac{\hat{b}_0(t) + \hat{b}_1(t)}{1 + \hat{a}_1(t)}, \tag{4.24}$$

$$T \leftarrow \frac{T_s}{-\log(-\hat{a}_1(t))}, \tag{4.25}$$

$$L \leftarrow \left\{ \frac{\hat{b}_1(t)}{\hat{b}_0(t) + \hat{b}_1(t)} + k \right\} T_s. \tag{4.26}$$

The controller is then re-constructed using the updated parameters given in (4.24)-(4.26). The flow chart of the proposed method is shown in Figure 4.5.

4.4 Numerical Example

To validate the effectiveness of the proposed method, an existing numerical model is used to perform the simulation. The following three methods are employed by the system model for comparison:

- Conventional pole-assignment controller constructed by (4.1)-(4.14).
- Conventional self-tuning control.
- The proposed method.

In the simulations, the initial system parameters are well-known. Moreover, the self-tuning and the proposed controller are applied when the system is in a stable state, i.e., from $t = 1000$ steps, and it is observed for a considerable amount of time to ensure a stationary state is maintained.

The control system is given by:

$$G(s) = \frac{K}{1 + 150s} e^{-15s}, \tag{4.27}$$

where

$$K = \begin{cases} 1 & (t < 3000) \\ 1 + \dfrac{5(t - 3000)}{1000} & (3000 \leq t \leq 4000) \\ 6 & (t > 4000) \end{cases} \cdot \tag{4.28}$$

The system in (4.27) is discretized at sampling time $T_s = 10$, and the white Gaussian noise with zero mean and variance 0.1^2 is set as the model error. In addition, it is assumed that the system gain changes with time as shown by (4.28). In addition, the reference signal $r(t)$ is set as 10.

First, the control result employed in the conventional pole-assignment controller is shown in Figure 4.6. The user-specified parameter δ in (4.14) is set at 2.0. Before system variance, good control performance is maintained and a certain amount of the variance of $\Delta u(t)$ is also maintained. However,

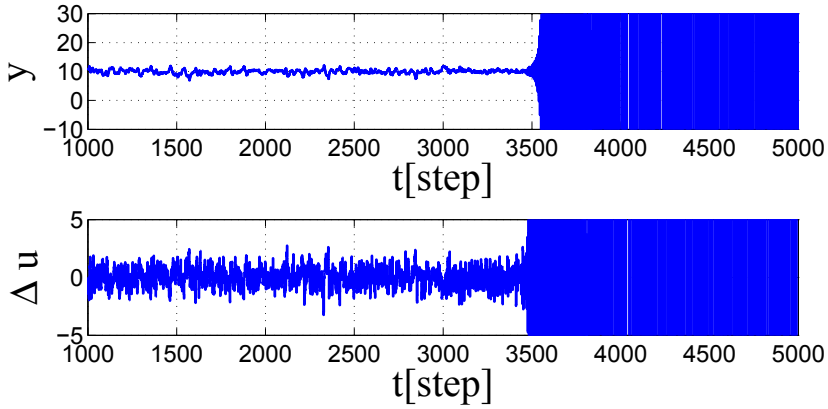

Figure 4.6: Control result obtained conventional pole-assignment control with $\delta = 2.0$.

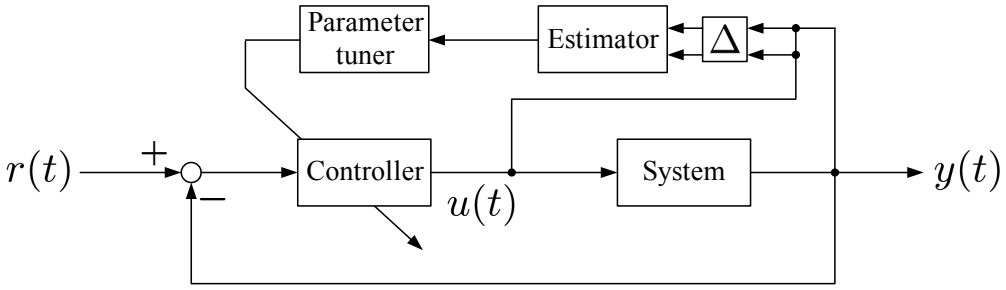

Figure 4.7: Block diagram of self-tuning controller.

at $t = 3500$[step], the controller cannot adapt to the varying system because of the fixed δ, thereby allowing the control system to fall into an unstable state.

Next, the conventional self-tuning controller shown in Figure 4.7 is employed. Note that the controller is designed based on the pole-assignment strategy in (4.1)-(4.14) and δ is fixed at 2.0. Furthermore, the system parameters are estimated by the Estimator based on the WRLS method in (4.19)-(4.23), and the control parameters are iteratively tuned by the Parameter tuner using these parameters. The control result and trajectories of the control parameters are shown in Figure 4.8 and Figure 4.9. In addition, the trajectories of the true and estimated system parameters are shown in Figure 4.10. The self-tuning controller can attain stable control performance

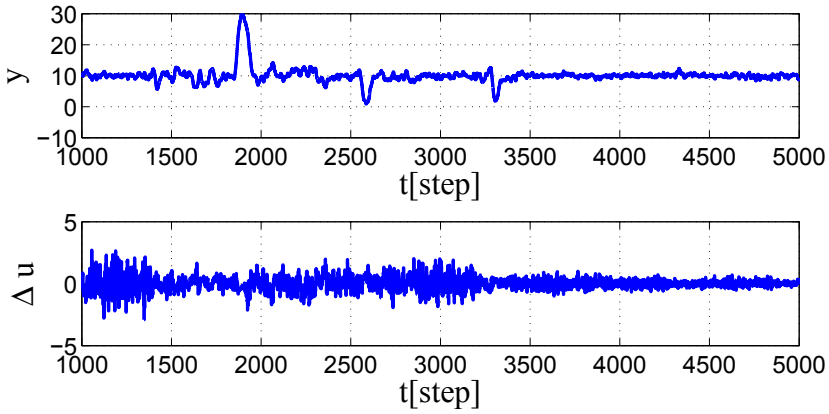

Figure 4.8: Control result obtained conventional self-tuning control with $\delta = 2.0$.

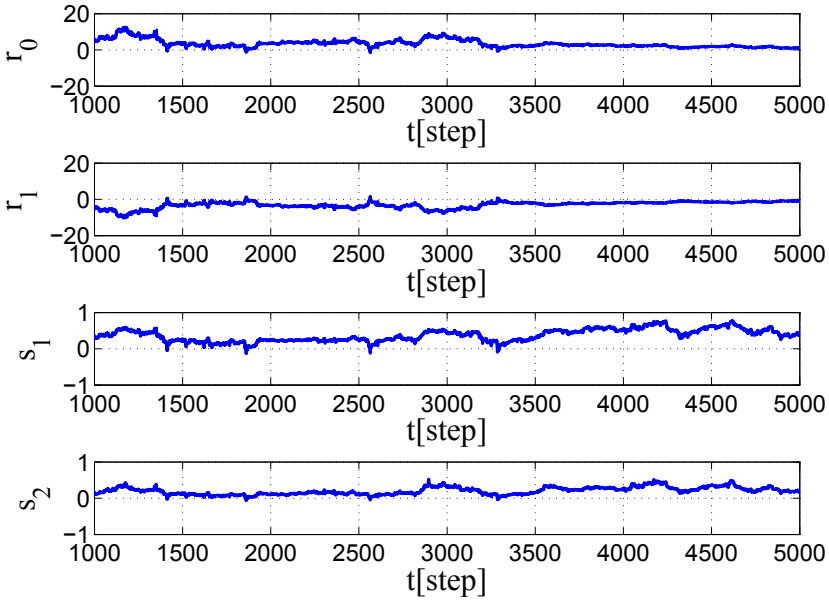

Figure 4.9: Trajectories of control parameters corresponding to Figure 4.8.

because the control parameters are adaptively-tuned based on the estimated system parameters. However, the control performance depends directly on the accuracy of these estimated system parameters, causing the control performance to degrade around $t = 1900$[step], 2600[step], and 3300[step] because of inaccuracies in the estimations.

The proposed method is applied to mitigate this problem of poor estimation of the parameters. The initial user design parameters in the method are

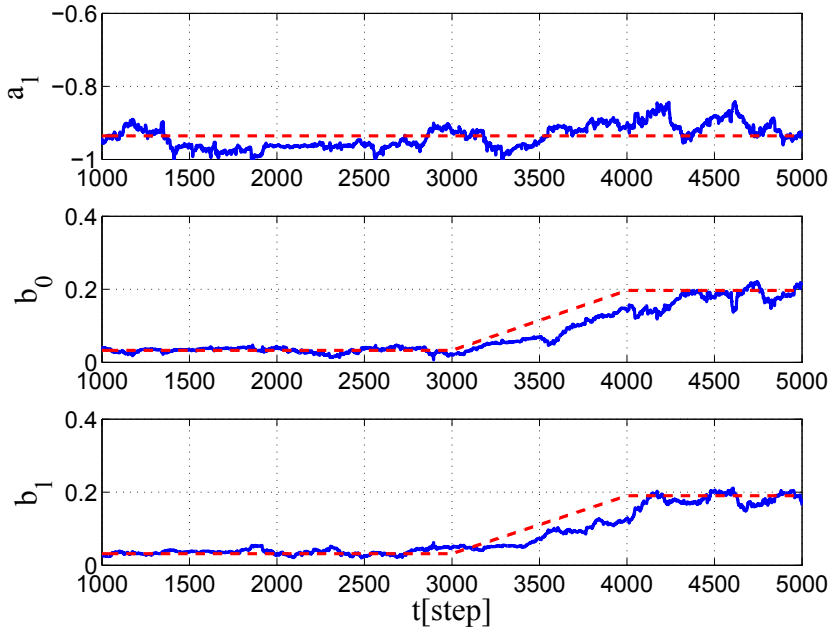

Figure 4.10: Trajectories of system parameters as used in controller corresponding to Figure 4.8 . (Dash line: True value of system parameters; solid line: Estimated parameters using weighted recursive least squares.)

Table 4.1: Initial design parameters in proposed method.

Initial value of δ	$\delta_f = 2.0$
Amount of δ changing	$\delta_\Delta = 0.001$
Dead zone	$\gamma = 2.0$
Range of δ movement	$\delta_{lower} = 1.5$
	$\delta_{upper} = 2.5$

shown in Table 4.1. The control result and trajectory of δ are shown in Figure 4.11, trajectories of the control parameters are depicted in Figure 4.12, the trajectories of the true and estimated system parameters are shown in Figure 4.13, and the transition of $J(t)$, which is the control-performance-assessment index, is shown in Figure 4.14. In Figure 4.11, good control performance can be maintained by adjusting δ based on the control-performance evaluation even if the system gain changes. Additionally, the variances of $y(t)$ and $\Delta u(t)$ are continually repressed constant values. At $t = 3625$[step], the system parameters are reidentified because δ falls below δ_{lower}. As a result, the

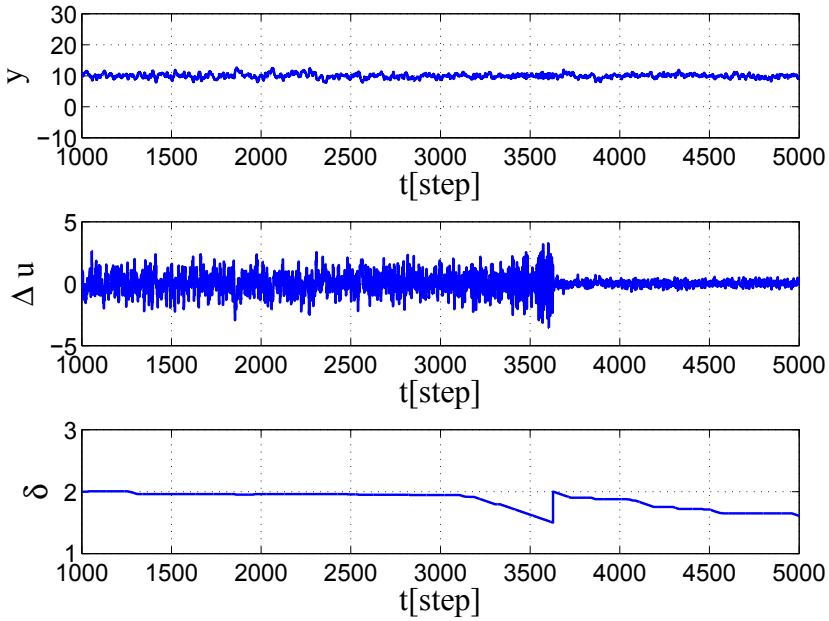

Figure 4.11: Control result as applied to proposed method and trajectory of δ corresponding to system output.

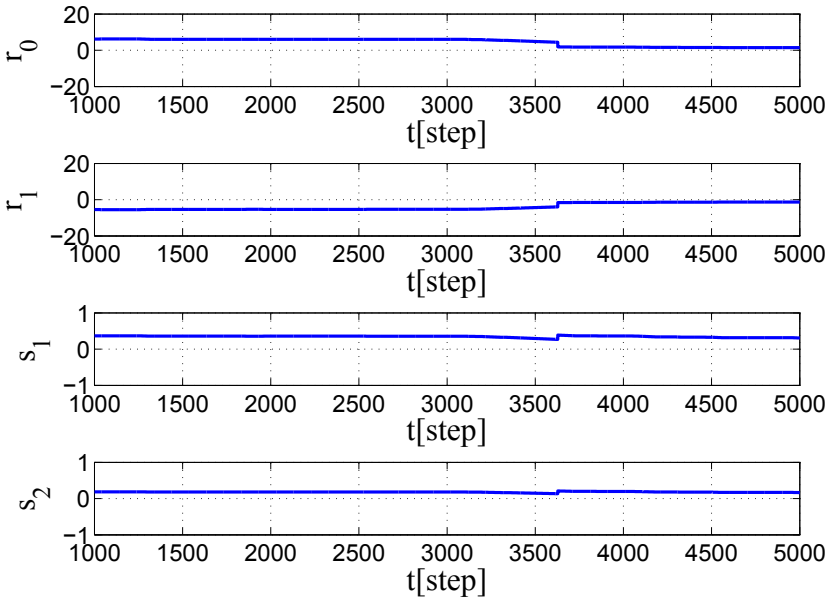

Figure 4.12: Trajectories of control parameters corresponding to Figure 4.11.

estimated parameters in the controller are updated, and subsequently control parameters are updated at $t = 3625$[step] (see Figure 4.12 and Figure 4.13). Moreover, after the system reidentification, the estimated system parameters K, T, and L are updated according to (4.24)-(4.26) and γK^2 and K^2/γ are

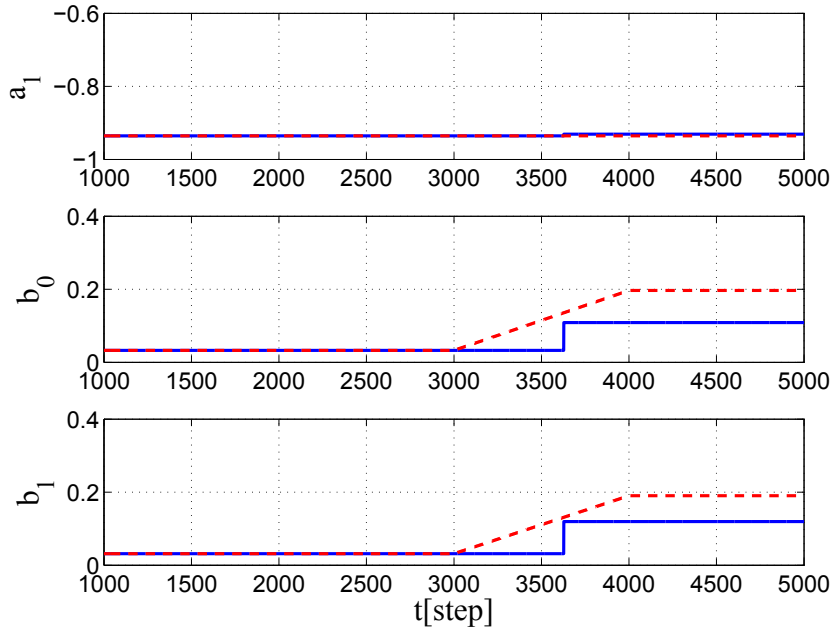

Figure 4.13: Trajectories of system parameters as used in controller corresponding to Figure 4.11 . (Dash line: True value of system parameters; solid line: Estimated parameters using weighted recursive least squares.)

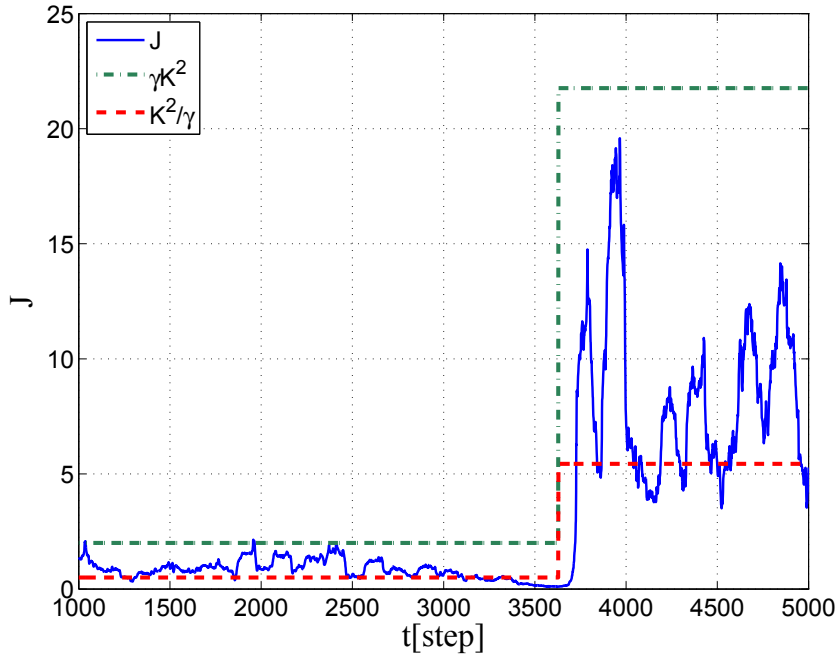

Figure 4.14: Trajectories of control-performance-assessment index J corresponding to Figure 4.11.

recalculated, with the result shown in Figure 4.14. In addition, the control-performance-evaluation index J is maintained at the threshold value by only

Figure 4.15: Appearance of weigh feeder.

adjusting δ after 3625[step].

4.5 Application to Weigh Feeder

In this section, the proposed controller is applied to a weigh feeder, as shown in Figure 4.15.

4.5.1 Control object

The operation of a weigh feeder can be summarized as follows. First, the weigh feeder converts the input DC voltage into AC voltage using an inverter. Then, the disk in the weigh feeder is rotated by the AC motor and a certain amount of powder in the hopper is released according to the number of rotations of the disk. The amount of released powder is measured by the loss-in-weight method [68]. The characteristics of the weigh feeder may vary every hour because of the type of powders being used, and the amount of powder remaining in the hopper. Thus, in order to maintain the desired level of control performance, the control system must be reconstructed based on a system property. In this case, the controlled variable $y(t)$ is the amount of

released powder in time units and the control input $u(t)$ is the DC voltage to the inverter. The range of the input DC voltage is limited to 9.8 V.

4.5.2 Control result

The initial system parameters are calculated using a Genetic Algorithm (GA) [41], as the least squares method cannot accurately estimate the system parameters (i.e., the time constant T is computed as a negative number). Using the GA, the system model is given by

$$(1 - 0.893z^{-1})y(t) = z^{-1}(0.199 + 0.218z^{-1})u(t) + \frac{\xi(t)}{\Delta}, \qquad (4.29)$$

where the sampling time T_s is 1[s] and the reference signal $r(t)$ is set as 0.01[kg/s].

In this demonstration, the following three methods are employed for the same simulation:

- Conventional pole-assignment controller constructed by (4.1)-(4.14).
- Conventional self-tuning control.
- The proposed method.

The conventional self-tuning control and the proposed method are applied after $t = 60$[step].

First, the control result obtained the pole-assignment control with $\delta = 0.6$ is shown in Figure 4.16. The result shows that the control system falls into an unstable state because the initial estimated system parameters are not good and so the constructed controller cannot adapt to the system.

Next, the conventional self-tuning controller is employed with δ set at 0.6. The control result and trajectories of the control parameters are shown in Figure 4.17 and Figure 4.18. In addition, the trajectories of the estimated system parameters under the WRLS method are shown in Figure 4.19. The results show that the conventional self-tuning controller can attain a more stable

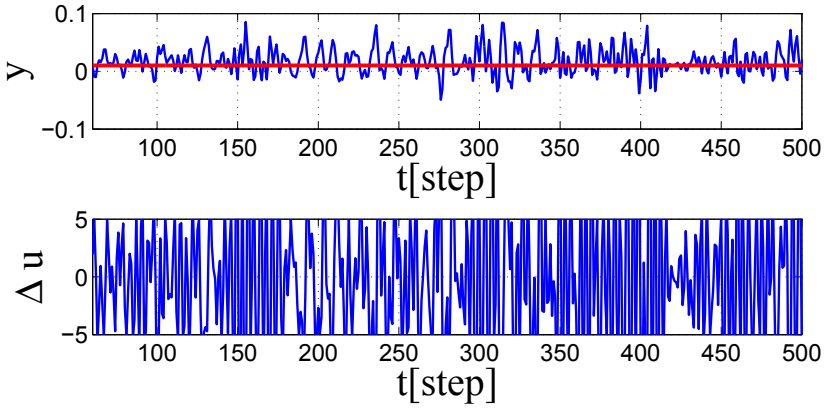

Figure 4.16: Control result obtained conventional pole-assignment control with $\delta = 0.6$.

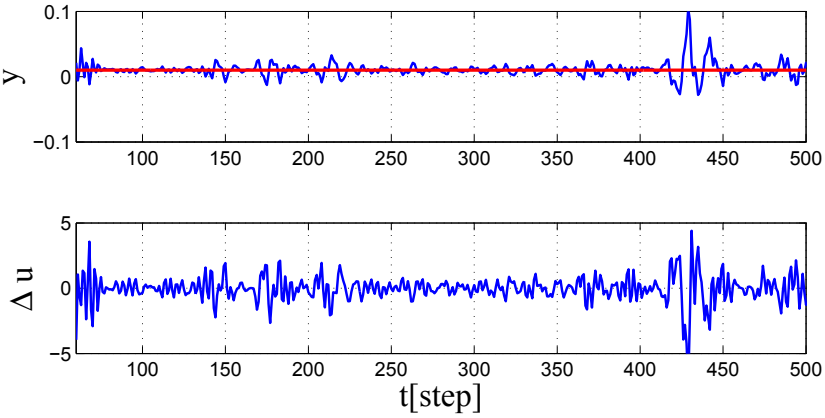

Figure 4.17: Control result obtained conventional self-tuning controller.

control performance than the conventional pole-assignment controller. This is because in the conventional self-tuning control, the parameters are changed immediately after being applied to the controller. However, at 430[step], the control performance degrades because system parameters are continually estimated even though a good control performance is obtained.

Finally, Figure 4.20 shows the result when the proposed method is applied. The initial user design parameters in this method are shown in Table 4.2. Figure 4.20 shows that δ is appropriately changed by control-performance evaluation. As a result, good control performance can be maintained using the proposed method. Moreover, after 100[step] the control pa-

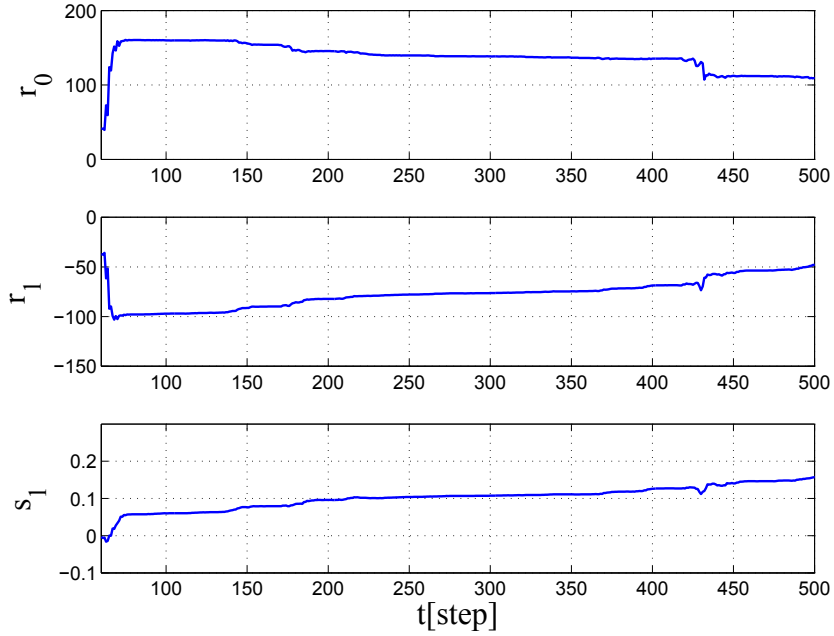

Figure 4.18: Trajectories of control parameters corresponding to Figure 4.17.

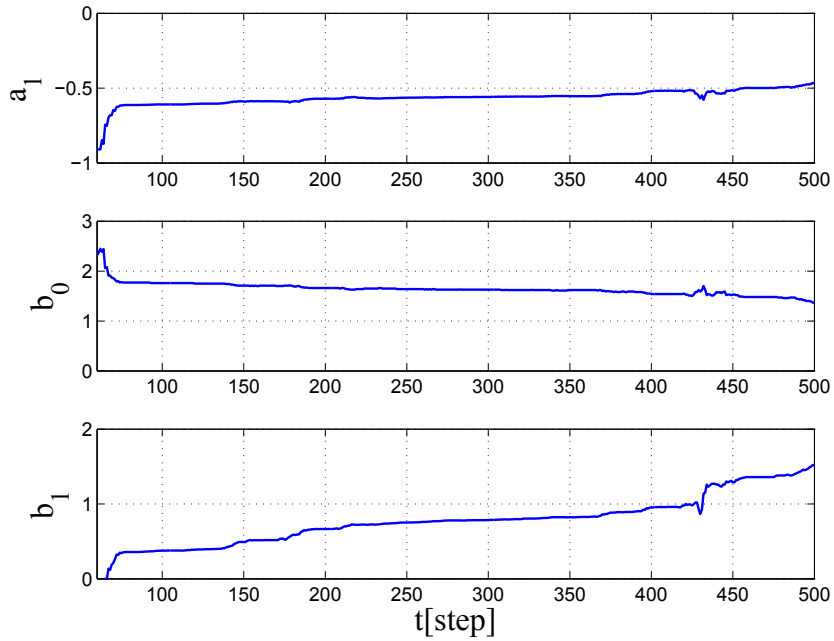

Figure 4.19: Trajectories of estimated system parameters used in controller corresponding to Figure 4.17.

rameters converge to maintain steady values along with δ, implying-that the system output is stationary, and system reidentification is not needed because δ does not fall out of the δ_{upper} and δ_{lower} range.

Table 4.2: Design parameters in proposed method for weigh feeder.

Initial value of δ	$\delta_f = 0.6$
Amount of δ changing	$\delta_\Delta = 0.005$
Dead zone	$\gamma = 2.0$
Range of δ movement	$\delta_{lower} = 0.3$
	$\delta_{upper} = 0.9$

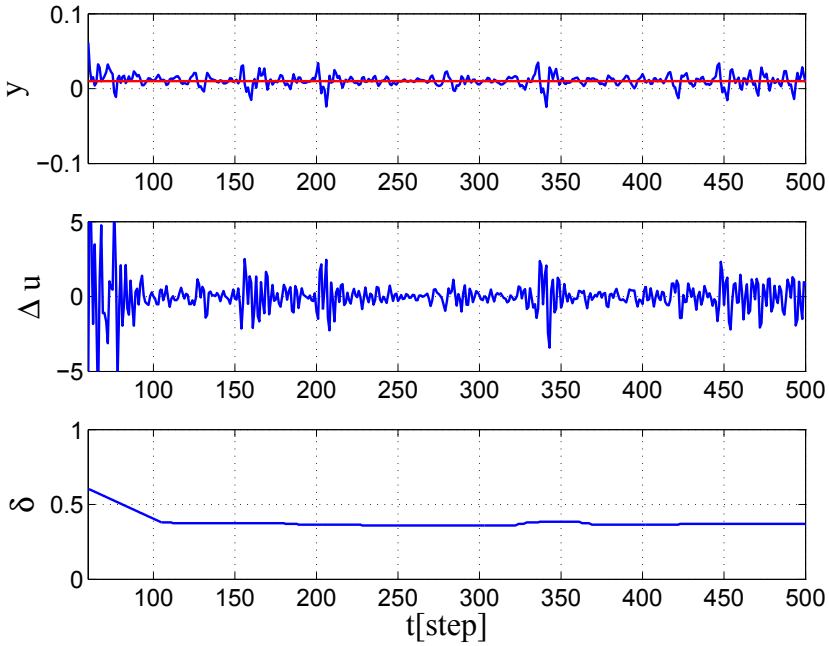

Figure 4.20: Control result obtained using proposed method and trajectories of δ corresponding to system output.

4.6 Conclusions

In this paper, the design of a performance-driven controller based on evaluation and design is presented. The pole-assignment controller is first designed, where the design polynomial that can determine the control parameters has only one user-specified parameter δ and the sensitivity of the controller is tuned by adjusting δ. While controlling, δ is adjusted using a control-performance assessment, and the method is called one-parameter tuning. The one-parameter tuning has following features:

i) The desired control performance can be obtained by adjusting only

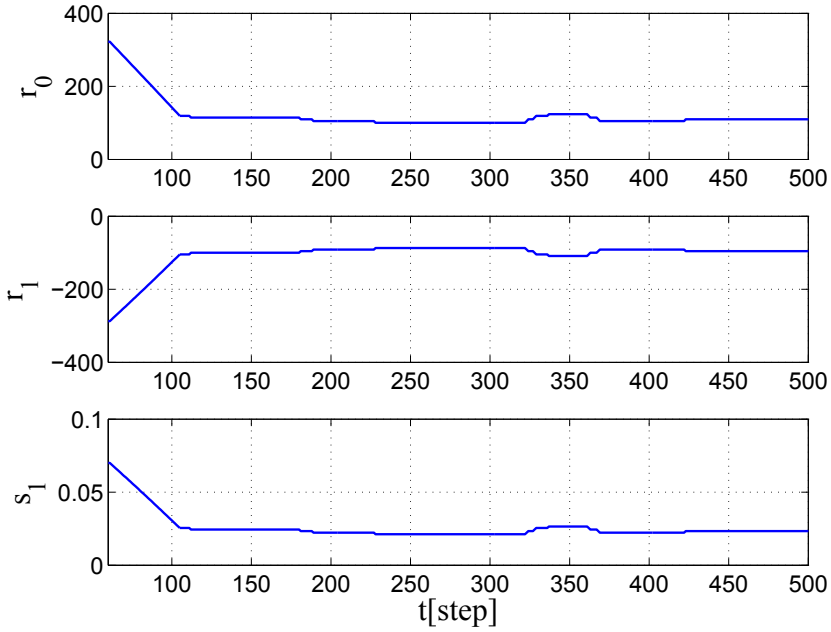

Figure 4.21: Trajectories of control parameters corresponding to Figure 4.20.

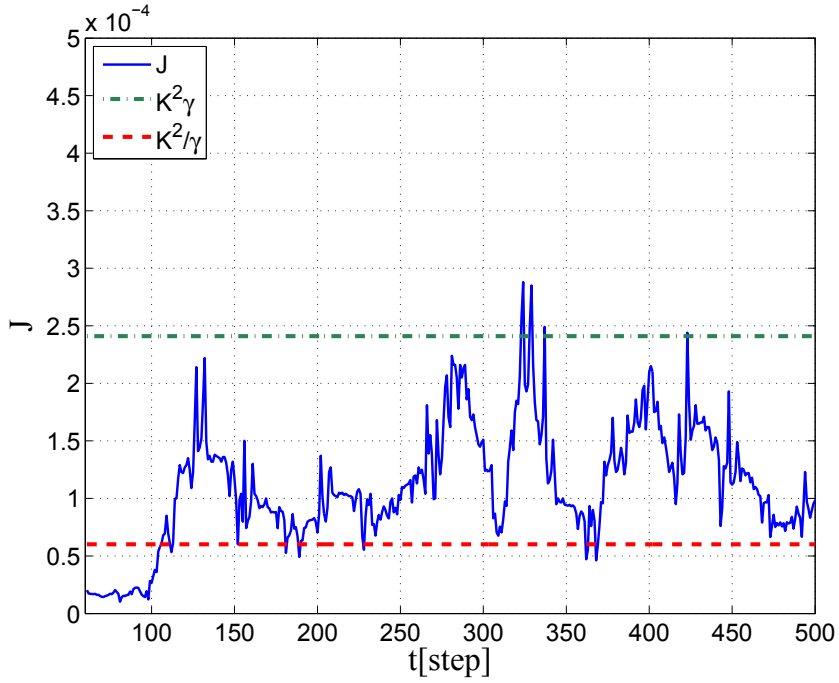

Figure 4.22: Trajectories of control performance assessment index J corresponding to Figure 4.20.

one user-specified parameter, δ.

ii) It is not necessary to identify system parameters recursively. The

user-specified parameter δ becomes the measure whether the system parameters need to be reidentified.

iii) The desired control performance is maintained by adjusting δ regardless of the accuracy of the system identification.

The effectiveness of the proposed method is verified using a numerical example and a real process system. Moreover, in the experiment using the weigh feeder, δ is appropriately adjusted, confirming that the proposed method can obtain excellent control performance.

Chapter 5

Conclusions

In this thesis, in order to establish a design method for data-oriented controllers, three design methods, from different perspectives, that use closed-loop data for controller designs are presented, and the respective effectiveness of each method is verified by simulation and experimental results. The three methods are summarized as follows:

i) Direct controller design method using closed-loop data

ii) Intelligent controller design method using closed-loop data

iii) Performance-driven controller design method that unifies control-performance assessment and controller design

In Chapter 1, the research background and necessity of data-oriented controller designs were discussed.

In Chapter 2, a proportional-integral-derivative (PID) controller design method based on GMVC was presented corresponding to i). According to this method, a PID controller can be designed without the need for system identification by converting control parameters that are approximated by the implicit GMVC controller to a PID controller. Moreover, the controller considers the control performance that can be designed by tuning the controller's sensitivity by adjusting a user-specified parameter λ. Furthermore, it was demonstrated that the proposed method is easily extended to MIMO systems by expanding a system description to a $p \times p$ expression.

In Chapter 3, an intelligent controller design method that introduced the idea of the FRIT method for CMAC learning (CMAC-FRIT) was proposed corresponding to ii). In the proposed method, offline learning can be performed by using a fictitious reference signal in the FRIT method to learn the CMAC-PID controller. This controller is effective for nonlinear systems. According to the proposed method, the problem of online learning implementing the CMAC algorithm is resolved, and the usefulness of the CMAC-PID controller is improved. Furthermore, in this chapter, it was also shown that the learned CMAC-PID tuner could be converted to a GMDH-PID tuner using the GMDH network in order to reduce the required memory for implementing the CMAC-PID tuner. Therefore, the nonlinear PID tuner could be implemented on a general-purpose microcomputer, thereby extending the application range.

In Chapter 4, as a design method for iii), a one-parameter tuning method that adjusts its one user-specified parameter based on control-performance assessment was proposed. According to this method, control-performance assessment and controller redesign is unified by only one user-specified parameter; thus, the performance-driven controller design scheme becomes clearer.

The most significant weakness of the data-oriented control design is the stability of a controller cannot be analytically guarantied. This is because conventional stability analysis approaches such as a robust stability theory cannot be applied to the data-oriented controller design method, because it does not explicitly use a system model. For instance, the one-parameter tuning method in Chapter 4 can achieve the stability of a closed-loop system when the user-specified parameter is set to a small value. Therefore, if the stability of a closed-loop system can be discriminated by the control-performance-evaluation mechanism, the stability is consistently maintained by adjusting the user-specified parameter. A stability discriminant method is identified as a future work.

Finally the future of data-oriented controller design scheme is discussed. In recent years, information throughout the world has increased explosively. These reasons for this include the technical advances of computers and networking maintenance, the implementation of supercomputers, and in the industrial world, the technical advancements of measurement hardware systems. Moreover, in industrial systems, the large data processing techniques that store information in databases, analyze, and leverage the data, have attracted much attention. These techniques have already influenced the marketing field. In the research field of control, data extracted from databases is being used for determining an input signal or controller design [69], typified by just-in-time (JIT) [70, 71] and lazy learning[72]. Moreover, a data-driven PID control design method [73, 74, 75] is also proposed. However, these methods use only the I/O relationships of data. It is hard to say that these methods enough use information of a database. The substance of a database should be treated multilaterally using data mining techniques, and a control system should be adjusted based on this information. The design of a controller should not be limited only to I/O data. For instance, many deterioration/abnormal diagnosis methods using data mining have been proposed, however, the results of these studies have not been applied to controller design. It is agreed that system operation must be stopped if a system becomes heavily damaged. However, if a system becomes degraded (for example, battery capacitance degradation), the system performance can be used to economically redesign a controller so that it can maximize the performance during the degradation. Hereafter, all information can become data that is stored in a large database. It proposed that a data-oriented method that analyzes this data multilaterally and maintains all aspects of control objects (for example, human, product, or economy) at an optimal state will be required. Therefore, the fusion of data mining and controller design is advanced by our research.

Bibliography

[1] L.Ljung, ***System Identification: Theory for the user***. Prentice-Hall, 1999.

[2] S.Adachi and A.Sano, "System indentification : From fundamentals to recent topics[2] (in japanese)," ***Journal of the Japan Society For Simulation Technology***, vol. 12, no. 2, pp. 128–135, 1993.

[3] P.Dorato, "Historical review of robust control," ***IEEE Control Systems Magazine***, vol. 7, no. 2, pp. 44–47, 1987.

[4] H.Hjalmarsson, S.Gunnarson, and M.Gevers, "A convergent iterative restricted complexity control design scheme," ***Proc. of the 33rd IEEE conference on decision and control***, vol. 2, pp. 1735–1740, 1994.

[5] H.Hjalmarsson, M.Gevers, S.Gunnarsson, and O.Lequin, "Iterative feedback tuning : theory and applications," ***IEEE Control System Magazine***, vol. 18, no. 4, pp. 26–41, 1998.

[6] H.Hjalmarsson, "Iterative feedback tuning –an overview," ***Int. J. Adapt. Control Signal Processing***, vol. 16, no. 5, pp. 373–395, 2002.

[7] K.Hamamoto, T.Fukuda, and T.Sugie, "Iterative feedback tuning of controllers for a two-mass-spring system," ***Control Engineering Practice***, vol. 11, pp. 1061–1068, 2003.

[8] M.Nakamoto, "Direct pid parameter tuning method using experimental data (¡special issue¿parameters tuning in pid control) (in japanese)," ***Trans. of the Institute of Systems, Control and Information Engineers***, vol. 50, no. 12, pp. 447–452, 2006.

[9] R.Hildebrand, A.Lecchini, G.Solari, and M.Gevers, "Asymptotic accuracy of iterative feedback tuning," ***IEEE Trans. on Automat. and Contr.***, vol. 50, no. 8, pp. 1182–1185, 2005.

[10] M.C.Campi, "Virtual reference feedback tuning (vrft) : A direct method for the design of feedback controllers," ***Automatica***, vol. 38, no. 8, pp. 1337–1346, 2002.

[11] A.Lecchini, M.C.Campi, and S.M.Savaresi, "Virtual reference feedback tuning for two degree of freedom controllers," ***International Journal of Adaptive Control and Signal Processing***, vol. 16, no. 5, pp. 355–371, 2002.

[12] M.C.Campi, A.Lecchini, and S.M.Savaresi, "An application of the virtual reference feedback tuning (vrft) method to a benchmark active suspension system," ***European Journal of Control***, vol. 9, pp. 66–76, 2003.

[13] M.C.Campi, A.Lecchini, and S.M.Savaresi, "An application of the virtual reference feedback tuning method to benchmark problem," ***European Journal of Control***, vol. 9, no. 1, pp. 66–76, 2003.

[14] F.Previdi, T.Schauer, S.M.Savaresi, and K.J.Hunt, "Data-driven control design for neuroprotheses : A virtual reference feedback tuning (vrft) approach," ***IEEE Trans. on Control Systems Technology***, vol. 12, no. 1, pp. 176–182, 2004.

[15] S.Soma, O.Kaneko, and T.Fujii, "A new approach to parameter tuning of controllers by using one-shot experimental data : A proposal of fictitious reference iterative tuning (in japanese)," ***Trans. of the Institute of Systems, Control and Information Engineers***, vol. 17, no. 12, pp. 528–536, 2004.

[16] 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.

[17] O.Kaneko, K.Yoshida, K.Matsumoto, and T.Fujii, "A new parameter tuning for controllers based on least-squares method by using one-shot closed loop experimental data : An extension of fictitious reference iterative tuning (in japanese)," ***Trans. of the Institute of Systems, Control and Information Engineers***, vol. 18, no. 11, pp. 400–409, 2005.

[18] J.G.Ziegler and N.B.Nichols, "Optimum settings for automatic controllers," ***Trans. ASME***, vol. 64, no. 8, pp. 759–768, 1942.

[19] K.L.Chien, J.A.Hrones, and J.B.Reswick, "On the automatic control of generalized passice systems," ***Trans. ASME***, vol. 74, pp. 175–185, 1952.

[20] N.Suda and etal, ***PID Control (in Japanese)***. Asakura Publishing Company, 1992.

[21] R.Vilanova and A.Visioli, ***PID control in the Third Millennium : lessons learned and new approaches***. Springer, 2012.

[22] S.Adachi and A.Sano, "System indentification : From fundamentals to recent topics[4] (in japanese)," ***Journal of the Japan Society For Simulation Technology***, vol. 12, no. 4, pp. 304–312, 1993.

[23] M.G.Safonov and T.C.Tsao, "Ieee trans. on automat. contr." ***The Unfalsified Control Consept and Learning***, vol. 42, no. 6, pp. 843–847, 1997.

[24] S.Masuda, M.Kano, Y.Yasuda, and G.D.Li, "A fictitious reference iterative tuning method with simulations delay parameter tuning of the reference model," ***IJICIC***, vol. 6, no. 7, pp. 2927–2939, 2010.

[25] Y.Ohnishi and T.Yamamoto, "A design of a frit based nonlinear pid controller," ***10th IFAC International Workshop on the Adaptation and Learning in Control and Signal Processing***, pp. 152–155, 2010.

[26] Y.Wakasa, S.Kanagawa, T.Kanya, and Y.Nishimura, "Frit with deadzone 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.

[27] 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.

[28] 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.

[29] S.Masuda, "Pid controller tuning based on disturbance attenuation frit using one-shot experimental data due to a load change disturbance," ***Proc. of 2nd IFAC Conference on Advances in PID Control***, pp. 92–97, 2012.

[30] 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.

[31] T.Shigemasa, Y.Negishi, and Y.Baba, "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***, pp. 337–342, 2013.

[32] F.Uozumi, O.Kaneko, and S.Yamamoto, "Fictitious reference iterative tuning of disturbance observers for attenuation of the effect of periodic unknown exogenous signals," ***Proc. of 11th IFAC International Workshop on Adaptation and Learning in Control and Signal Processing***, pp. 576–581, 2013.

[33] K.S.Fu, "Learning control systems and intelligent control systems; an tntersection of artifical intelligence and automatic control," ***IEEE Trans. on Automatic Control***, pp. 70–72, 1971.

[34] E.Mishkin and L.Braun, ***Adaptive Control Systems***. McGraw-Hill, 1961.

[35] J.S.Albus, "A new approach to manipulator control: The cerebellar model articulation controller," ***Transaction of the ASME***, vol. 97, no. 3, pp. 270–277, 1975.

[36] B.Bavarian, "Introduction to neural networks for intellignet control," ***IEEE Control Systems Magazine***, vol. 8, no. 2, pp. 3–7, 1988.

[37] S.Omatu, K.Marzuki, and Y.Rubiyah, ***Neuro-Control and Its Applications***. Springer-Verlag, 1995.

[38] T.Yamamoto, R.Kurozumi, and S.Fujisawa, "A design of cmac based intelligent pid controllers," ***Artificial Neural Networks and Neural Information Processing, Lecture Notes in Computer Science***, vol. 2714, pp. 471–478, 2003.

[39] R.Kurozumi, T.Yamamoto, and S.Fujisawa, "Development of training equipment with an adaptive and learning mechanism using balloon actuator-sensor system," ***Proceedings of SMC2007***, pp. 2624–2629, 2007.

[40] E.H.Mamdani, "Application of fuzzy algorithms for control of a simple dynamic plant," ***Proc. of IEE***, vol. 121, no. 12, pp. 1585–1588, 1974.

[41] D.E.Goldberg, ***Genetic Algorithm in Search, Optimization & Machine Learning***. Addison-Wesley, 1989.

[42] J.D.Farmer, N.H.Packard, and A.S.Perelson, "The immune system, adaptation, and machine learning," ***Physica***, vol. 22-D, pp. 184–204, 1986.

[43] J.Kenedy and R.Eberhart, ***Swarm Intelligence***. Morgan Kaufmann Publishers, 2001.

[44] T.J.Harris, "Assessment of closed loop performance," ***Canadian J. Chemical Engineering***, vol. 67, pp. 856–861, 1989.

[45] B.Huang and S.L.Shah, ***Performance Assessment of Control Loops: Theory and Applications***. Springer-Verlag, 1999.

[46] B.Huang, "A pragmatic approach towards assessment of control loop performance," ***International Journal of Adaptive Control and Signal Processing***, vol. 17, pp. 589–608, 2003.

[47] T.Yamamoto and S.L.Shah, "Design of a performance-adaptive pid controller," ***Proc. of the IEEE International Conference on Networking, Sensing and Control***, pp. 547–552, 2007.

[48] K.Takao, Y.Ohnishi, T.Yamamoto, and T.Hinamoto, "Design of a performance adaptive pid controller based on control performance assessment(in japanese)," ***Trans. Society of Instrument and Control Engineers***, vol. 43, no. 2, pp. 110–117, 2007.

[49] T.Yamamoto, "Design of performance-adaptive pid control system based on modelimg performance assessment(in japanese)," ***IEEJ Transactions on Electronics, Information and Systems***, vol. 27, no. 12, pp. 2101–2108, 2007.

[50] K.Fujii and T.Yamamoto, "One-parameter tuning pid control of a distillation process," ***Proc. International Symposium on Advanced Control of Industrial Processes***, pp. 222–225, 2008.

[51] T.Yamamoto, K.Kawada, H.Kugemoto, and Y.Kutsuwa, "Design and industrial applications of a control performance assessment based pid controller," ***Proc. 15th IFAC Symposium on System Identification***, pp. 729–734, 2009.

[52] T.Yamamoto and S.Wakitani, "Design of a performance-driven self-tuning controller via one-parameter tuning," ***Proc. European Control Conference 2009***, pp. 4019–4024, 2009.

[53] M.Tokuda and T.Yamamoto, "A self-tuning pid controller based on control performance evaluations," ***International Journal of Innovative Computing, Information and Control***, vol. 6, no. 8, pp. 3751–3762, 2010.

[54] D.W.Clarke and P.J.Gawthrop, "Self-tuning controller," ***Proc. IEE***, vol. 122, no. 9, pp. 929–934, 1975.

[55] D.W.Clarke and P.J.Gawthrop, "Self-tuning control," ***Proc. IEE***, vol. 126D, pp. 633–640, 1979.

[56] A. G. Ivakhnenko, "The group method of data handling, a rival of the method of stochastic approximation," ***Soviet Automatic Control***, vol. 13, no. 3, pp. 43–55, 1968.

[57] P.E.Wellstead and M.B.Zarrop, ***Self-Tuning Systems : Control and Signal Processing***. John Wiley & Sons, 1991.

[58] T.J.Harris, J.F.MacGregor, and J.D.Wright, "Self-tuning and adaptive controllers: an application to catalytic reactor control," ***Technometrics***, vol. 22, no. 2, pp. 153–164, 1980.

[59] P.R.Belanger, "On type 1 systems and the clarke-gawthrop regulator," ***Automatica***, vol. 19, pp. 91–94, 1983.

[60] M.Mahfouf, D.A.Linkens, and M.F.Abbod, "Adaptive fuzzy tsk model-based predictive control using a carima model structure," ***Trans. IChemE***, vol. 78, no. Part A, pp. 590–596, 2000.

[61] 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.

[62] A.Sakaguchi and T.Yamamoto, "A design of generalized minimum variance controllers using a gmdh network for nonlinear systems," ***IEICE transactions on fundamentals of electronics, communications and computer sciences***, vol. E84-A, no. 11, pp. 2901–2907, 2001.

[63] S.Wakitani, Y.Ohnishi, and T.Yamamoto, "Design of a cmac-based pid controller using operating data," ***Distributed Computing and Artificial Intelligence Advances in Intelligent and Soft Computing***, vol. 151, pp. 545–552, 2012.

[64] S.Wakitani, Y.Ohnishi, and T.Yamamoto, "Design of a cmac-based pid controller using frit for nonlinear systems (in japanese)," ***Trans. of the Society of Instrument and Control Engineers***, vol. 48, no. 12, pp. 847–853, 2012.

[65] D.W.Clarke and P.J.Gawthrop, "Self-tuning control," ***IEE Proc. Control Theory and Aplications***, vol. 126, no. 6, pp. 633–640, 1979.

[66] S.Haykin, ***Adaptive filter theory***. Prentice Hall, 1986.

[67] R.M.Miller, S.L.Shah, and R.K.Wood, "Adaptive predictive control employing on-line time delay estimation," ***Advances in Instrumentation and Control***, pp. 857–864, 1995.

[68] M.Hopkins, "Loss in weight feeder systems," ***Measurement and Control***, vol. 39, no. 8, pp. 237–240, 2006.

[69] D.Inoue and S.Yamamoto, "A memory-based predictive control approach to a braking assist problem," ***Proc. of SICE Annual Conference 2005***, pp. 3186–3189, 2005.

[70] 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.

[71] N. Nakpong and S. Yamamoto, "Just-in-time predictive control for a two-wheeled robot," ***The 10th International Conference on ICT and Knowledge Engineering***, pp. 95–98, 2012.

[72] G.Bontempi, M.Birattari, and H.Bersini, "Lazy learning for local modeling and control design," ***International Journal of Control***, vol. 72, no. 7–8, pp. 643–658, 1999.

[73] K.Takao, T.Yamamoto, and T.Hinamoto, "A design of memory-based pid controllers (in japnaese)," ***Trans. Society of Instrument and Control Engineers***, vol. 40, no. 9, pp. 898–905, 2004.

[74] 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.

[75] K.Hosokawa, S.Wakitani, and T.Yamamoto, "Design of a data-driven model-free type gmv-pid control system (in japanese)," ***Trans. of the Institute of Electrical Engineers of Japan. C***, vol. 133, no. 6, pp. 1103–1108, 2013.

Publication Lists

[1] S.Wakitani, K.Hosokawa and T.Yamamoto, "Design of an Implicit GMV-PID Controller Using Closed Loop Data (in Japanese)", ***Trans. of the Institute of Electrical Engineers of Japan. C***, vol. 132, no. 6, pp. 873–878, 2012.

[2] S.Wakitani, Y.Ohnishi and T.Yamamoto, "Design of a CMAC-Based PID Controller Using FRIT for Nonlinear Systems (in Japanese)", ***Trans. of the Society of Instrument and Control Engineers***, vol. 48, no. 12, pp. 847–853, 2012.

[3] K.Hosokawa, S.Wakitani and T.Yamamoto, "Design of a Data-Driven Model-Free Type GMV-PID Control System (in Japanese)", ***Trans. of the Institute of Electrical Engineers of Japan. C***, vol. 133, no. 6, pp. 1103–1108, 2013.

[4] S.Wakitani and T.Yamamoto, "Design and Experimental Evaluation of a Data-Oriented Multivariable PID Controller", ***Int. Journal of Advanced Mechatronic Systems***, vol. 5, no. 1, pp. 20–26, 2013

[5] S.Wakitani, T.Nawachi, G.R.Martins and T.Yamamoto, "Design and Implementation of a Data-oriented Nonlinear PID Controller", ***Journal of Advanced Computational Intelligence and Intelligent Informatics***, vol. 17, no. 5, pp. 690–698, 2013.

[6] S.Wakitani, T.Yamamoto, T.Sato and N.Araki, "Design of a Performance-Driven Controller by One-Parameter Tuning", ***Control Engineering Practice*** (submitting).

International Conference Papers

[1] T.Yamamoto and S.Wakitani, "Design of a Performance-Driven Self-Tuning Controller via One-Parameter Tuning", ***Proc. of European Control Conference 2009***, pp. 4019–4024, Budapest 2009.

[2] S.Wakitani, T.Yamamoto, T.Sato and N.Araki, "Design and Experimental Evaluation of a Performance-Driven Adaptive Controller", ***Proc. of 18th IFAC World Congress***, pp. 7322–7327, Milano 2011.

[3] Y.Ohnishi, S.Wakitani and T.Yamamoto, "A Design of Nonlinear PID Controller with Neural net based FRIT", ***Proc. of 2nd IFAC Conference on Advances in PID Control, 2012***, pp. 81–85, Brecia 2012.

[4] S.Wakitani, Y.Ohnishi and T.Yamamoto, "Design of a CMAC-Based PID Controller Using Operating Data", ***Distributed Computing and Artificial Intelligence Advances in Intelligent and Soft Computing (Proc. of International Symposium on Distributed Computing and Artificial Intelligence)***, vol. 151, pp. 545–552, Salamanca 2012.

[5] S.Wakitani, S.Hanata and T.Yamamoto, "Design and Application of a Direct GMV-Based PID Controller", ***Proc. of the 2012 International Conference on Advanced Mechatronic System***, pp. 541–546, Tokyo 2012

[6] S.Wakitani, T.Nawachi and T.Yamamoto, "Design of a Data-Oriented PID Controller for Nonlinear Systems", ***Proc. of 19th International Conference on Neural Information***, Prt V, pp. 169–176, Doha 2012.

[7] S.Wakitani, K.Nishida, M.Nakamoto and T.Yamamoto, "Design of a Data-Driven PID Controller using Operating Data", ***Proc. of 11th IFAC International Workshop on Adaptation and Learning in Control and Signal Processing***, pp.587–592, Caen 2013.

[8] Y.Ohnishi, H.Kitagawa, S.Mori, S.Wakitani and T.Yamamoto, "Design of Neural Networks Based FRIT PID Controllers and Its Applications", ***Proc. of 11th IFAC International Workshop on Adaptation and Learning in Control and Signal Processing***, pp.355–359, Caen 2013.

[9] S.Wakitani, G.Martins and T.Yamamoto, "Design of Data-Oriented GMDH-

Based Controller", ***Proc. of IEEE International Conference on Systems, Man, and Cybernetics***, pp.2516–2521, Manchester 2013.