IFAC Conference on Advances in PID Control
PID'12
Brescia (Italy), March 28-30, 2012 WeA2.3

Online Controller Tuning via FRIT and Recursive Least-Squares

Yuji Wakasa * **Kanya Tanaka** ** **Yuki Nishimura** ***

* *Graduate School of Science and Engineering, Yamaguchi University, 2-16-1 Tokiwadai, Ube, Yamaguchi 755-8611, Japan (e-mail: wakasa@yamaguchi-u.ac.jp).*
** *Graduate School of Science and Engineering, Yamaguchi University, 2-16-1 Tokiwadai, Ube, Yamaguchi 755-8611, Japan (e-mail: ktanaka@yamaguchi-u.ac.jp).*
*** *Graduate School of Science and Engineering, Yamaguchi University, 2-16-1 Tokiwadai, Ube, Yamaguchi 755-8611, Japan (e-mail: yunishi@yamaguchi-u.ac.jp).*

Abstract: This paper proposes an online type of controller parameter tuning method by modifying the standard fictitious reference iterative tuning method and by utilizing the so-called recursive least-squares (RLS) algorithm, which can cope with variation of plant characteristics adaptively. As used in many applications, the RLS algorithm with a forgetting factor is also applied to give more weight to more recent data, which is appropriate for adaptive controller tuning. Moreover, we extend the proposed method to online tuning of the feed-forward controller of a two-degree-of-freedom control system. Finally, numerical examples are provided to illustrate the effectiveness of the proposed method.

Keywords: Fictitious reference iterative tuning, online tuning, PID control, recursive least-squares algorithm, adaptive algorithm.

1. INTRODUCTION

For the last decade, some direct tuning methods of controller parameters such as proportional-integral-derivative (PID) gains have been investigated (Hjalmarsson (2002); Campi and Savaresi (2006); Souma et al. (2004)). These methods *directly* use experimental input and output data of a plant to tune controller parameters. They are therefore more practical than indirect methods which require a plant model identified by using the input and output data.

Among the representative direct controller parameter tuning methods, iterative feedback tuning (IFT) proposed in Hjalmarsson (2002) requires iterative experiments. In contrast, virtual reference feedback tuning (VRFT) proposed in Campi and Savaresi (2006) and fictitious reference iterative tuning (FRIT) proposed in Souma et al. (2004) are performed based on input and output data obtained from only a one-shot experiment, which means that VRFT and FRIT are more practical than IFT. Moreover, although FRIT and VRFT are based on a similar idea, FRIT is more intuitively understandable and simple than VRFT as stated in Kaneko et al. (2011). For these reasons, FRIT has received much attention recently as a practical and useful method, and its extended methods have been studied (see, e.g., Tasaka et al. (2009); Masuda (2010); Wakasa et al. (2011)).

The standard FRIT is basically performed offline. This means that once plant characteristics change, the control performance may be deteriorated, and therefore, FRIT has to be re-performed offline. To cope with this problem, online methods of FRIT have been proposed in Masuda (2010); Yamashina et al. (2011). In general, an optimization problem in the standard FRIT is not a convex programming problem, which leads to relatively long computation time to be solved. To avoid this difficulty, the standard FRIT is modified in Masuda (2010); Yamashina et al. (2011) so that the resulting optimization problem becomes a form of least-squares problem. However, these online methods based on the least-squares method still can be improved from a computational viewpoint. Moreover, in the method in Masuda (2010), controller parameters have to be updated periodically, so that the controller parameters may change considerably, thereby leading to control performance deterioration.

This paper proposes an online type of controller parameter tuning method by utilizing the so-called recursive least-squares (RLS) algorithm (see, e.g., Haykin (2002)) which takes less computational complexity than the standard least-squares algorithm. As used in many applications, the RLS algorithm with a forgetting factor is applied to give more weight to more recent data, which is appropriate for adaptive controller tuning. We also introduce a filter to avoid abrupt variation of controller parameters. Moreover, we extend the proposed method to online tuning of the feed-forward controller of a two-degree-of-freedom (2DOF) control system. Finally, numerical examples are provided to illustrate the effectiveness of the proposed method.

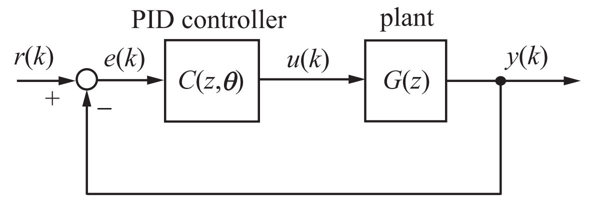

Fig. 1. System configuration.

2. SYSTEM CONFIGURATION

Consider a system configuration shown in Fig. 1. In the figure, $G(z)$ is a plant modeled as a discrete-time single-input and single-output linear system, $C(z, \boldsymbol{\theta})$ is a parameterized controller such as a PID controller, and $\boldsymbol{\theta}$ denotes a parameter vector to be tuned in the controller. Also, $u(k)$, $y(k)$, $r(k)$, and $e(k)$ denote the control input, control output, reference signal, and tracking error, respectively.

We assume that the controller $C(z, \boldsymbol{\theta})$ is linearly parameterized with respect to $\boldsymbol{\theta}$. For example, denoting

$$\boldsymbol{\theta} = [K_P,\ K_I,\ K_D]^T$$
$$\boldsymbol{\phi}_c(s) = \left[1,\ \frac{1}{s},\ \frac{s}{\tau s+1}\right]^T,$$

we can express a continuous-time transfer function of a PID controller as

$$C_c(s, \boldsymbol{\theta}) = \boldsymbol{\theta}^T \boldsymbol{\phi}_c(s),$$

where K_P, K_I, and K_D are the proportional, integral, and derivative gains, respectively, $\boldsymbol{\theta} = [K_P, K_I, K_D]^T$ includes the PID gains to be tuned, and τ is the filter time constant of the approximate derivative. In this paper, we denote a discretized model of $\boldsymbol{\phi}_c(s)$ by $\boldsymbol{\phi}(z)$ and use the following discrete-time PID controller as a typical case:

$$C(z, \boldsymbol{\theta}) = \boldsymbol{\theta}^T \boldsymbol{\phi}(z).$$

3. STANDARD FRIT

In the standard FRIT (Souma et al. (2004)), we first perform a closed-loop experiment to obtain input/output data $u_0(k)$, $y_0(k)$, $k = 1, \ldots, N$, for an initial controller parameter $\boldsymbol{\theta}_0$ and a reference signal $r(k)$. Then the fictitious reference signal is calculated by

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

Based on the fictitious reference signal, we tune the controller parameter $\boldsymbol{\theta}$ so that the following performance index is minimized:

$$J(\boldsymbol{\theta}) = \sum_{k=1}^{N} (y_0(k) - M(z)\tilde{r}(\boldsymbol{\theta}, k))^2,$$

where $M(z)$ is a given reference model that can express an ideal closed-loop system. The abovementioned tuning procedure is performed offline.

4. ONLINE TUNING VIA RECURSIVE LEAST-SQUARES METHODS

One of the reasons why the standard FRIT is performed offline is that J is usually not convex with respect to $\boldsymbol{\theta}$, and therefore, this computation cannot be efficiently carried out.

To cope with this difficulty, we first assume that $\boldsymbol{\theta}$ satisfies an ideal case:

$$y_0(k) - M(z)\tilde{r}(\boldsymbol{\theta}, k) = 0.$$

It follows from this assumption that

$$C(z, \boldsymbol{\theta}) y_0(k) = M(z) u_0(k) + C(z, \boldsymbol{\theta}) M(z) y_0(k).$$

By focusing on the above relationship, the tuning method by minimizing the following performance index has been proposed in (Masuda (2010)):

$$\hat{J}(\boldsymbol{\theta}) = \sum_{k=1}^{N} \hat{e}(k)^2,$$

where

$$\hat{e}(k) = C(z, \boldsymbol{\theta})(1 - M(z)) y_0(k) - M(z) u_0(k). \tag{1}$$

In this case, the minimization problem of $\hat{J}$ is regarded as a least-squares problem because $\hat{e}(k)$ is linear with respect to $\boldsymbol{\theta}$. In Masuda (2010), a period for evaluating the performance index is defined, and the normal equation corresponding to the least-squares problem is solved at each period to update controller parameters. Although this tuning procedure is carried out online, the controller parameters can be abruptly updated at a definite period of time, so that the control performance may be deteriorated. Moreover, the computational complexity of solving the normal equation is relatively large.

To resolve these problems, in this paper, we utilize the so-called RLS method (Haykin (2002)) and propose a method for reducing the variations of controller parameters.

We first replace the initial data $u_0(k)$, $y_0(k)$ with $u(k)$, $y(k)$ for (1) and define the following signals:

$$\boldsymbol{\xi}(k) = \boldsymbol{\phi}(z)(1 - M(z)) y(k) \tag{2}$$
$$d(k) = M(z) u(k). \tag{3}$$

Then we can describe the error as

$$\hat{e}(k) = \boldsymbol{\theta}^T \boldsymbol{\xi}(k) - d(k). \tag{4}$$

Therefore, when we aim to tune the controller parameters based on the data up to time k, we can express the performance index to be minimized as follows:

$$\hat{J}_k(\boldsymbol{\theta}) = \sum_{i=1}^{k} \hat{e}(i)^2 = \left\| \begin{bmatrix} \boldsymbol{\xi}(1)^T \\ \vdots \\ \boldsymbol{\xi}(k)^T \end{bmatrix} \boldsymbol{\theta} - \begin{bmatrix} d(1) \\ \vdots \\ d(k) \end{bmatrix} \right\|^2.$$

The RLS algorithm is an algorithm which recursively finds the optimal estimate $\hat{\boldsymbol{\theta}}(k)$ of controller parameters by using $\hat{\boldsymbol{\theta}}(k-1)$ at the previous time $k-1$ (Haykin (2002)). Since the standard RLS uses all data $u(k)$, $y(k)$ from the initial time to the current time, it cannot cope with characteristic variations of the plant. Therefore the RLS with a forgetting factor $\lambda (0 < \lambda < 1)$ is appropriate for such a case. The forgetting factor is a weighting factor which is introduced into the performance index as follows:

$$\hat{J}_k(\boldsymbol{\theta}) = \sum_{i=1}^{k} \lambda^{k-i} \hat{e}(i)^2.$$

The forgetting factor gives exponentially less weight to older error samples. When $\lambda = 1$, we have the standard RLS algorithm. The inverse of $1 - \lambda$ is, roughly speaking, a measure of the memory of the algorithm. Therefore, the special case $\lambda = 1$ corresponds to infinite memory (see for the details, e.g., Haykin (2002)).

The RLS algorithm with a forgetting factor is as follows:

$$\boldsymbol{h}(k) = \frac{\boldsymbol{P}(k-1)\boldsymbol{\xi}(k)}{\lambda + \boldsymbol{\xi}(k)^T \boldsymbol{P}(k-1)\boldsymbol{\xi}(k)} \tag{5}$$

$$\boldsymbol{P}(k) = (\boldsymbol{P}(k-1) - \boldsymbol{h}(k)(\boldsymbol{\xi}(k)^T \boldsymbol{P}(k-1)))/\lambda \tag{6}$$

$$\hat{\boldsymbol{\theta}}(k) = \hat{\boldsymbol{\theta}}(k-1) + \boldsymbol{h}(k)(d(k) - \boldsymbol{\xi}(k)^T \hat{\boldsymbol{\theta}}(k-1)). \tag{7}$$

To initialize the RLS algorithm, we need to specify the initial controller parameter $\hat{\boldsymbol{\theta}}(0)$ and the initial correlation matrix $\boldsymbol{P}(0)$. Usually we set the matrix

$$\boldsymbol{P}(0) = \gamma I,$$

where $\gamma > 0$ is set to be a large constant for high signal-to-noise ratio.

According to the abovementioned RLS algorithm, the controller parameter $\hat{\boldsymbol{\theta}}(k)$ is updated at each time. This variation of the controller parameter may be large at the beginning of the algorithm, at the time when plant characteristics change abruptly, and at the time when the set-point reference is changed. Due to this, the control performance can be deteriorated, and the system may fail to be stable in the worst case. Thus, to reduce the variation of the controller parameter, we propose the following update rule of the implemented controller parameter $\boldsymbol{\theta}(k)$:

$$\boldsymbol{\theta}(k) = (1 - \alpha)\boldsymbol{\theta}(k-1) + \alpha\hat{\boldsymbol{\theta}}(k-1), \tag{8}$$

where α is a sufficiently small positive constant. By the update rule (8), $\hat{\boldsymbol{\theta}}$ is filtered by the low path filter $\alpha/(z + \alpha - 1)$, so that $\boldsymbol{\theta}$ is changed moderately.

To sum up, the proposed online controller parameter tuning algorithm is described as follows:

Online controller parameter tuning algorithm

Step 1. Set an initial controller parameter $\hat{\boldsymbol{\theta}}(0) = \boldsymbol{\theta}(0) = \boldsymbol{\theta}_0$ and parameters γ, λ, and α. For each time, perform Steps 2–4.
Step 2. From (2) and (3), compute $\boldsymbol{\xi}(k)$ and $d(k)$.
Step 3. Obtain $\hat{\boldsymbol{\theta}}(k)$ by computing (5)–(7).
Step 4. Obtain $\boldsymbol{\theta}(k)$ according to (8) and implement it into the controller.

We show the block diagram of the online controller parameter tuning algorithm in Fig. 2. Moreover, a more detailed mechanism of the RLS algorithm is illustrated in Fig. 3.

Remark 1. As in the standard offline FRIT, the proposed online FRIT does not ensure the stability of the control system. A remedy for this problem is to restrict $\boldsymbol{\theta}(k)$ within the range aimed at ensuring the stabiliy of the control system by utilizing information on a pre-experiment or plant model.

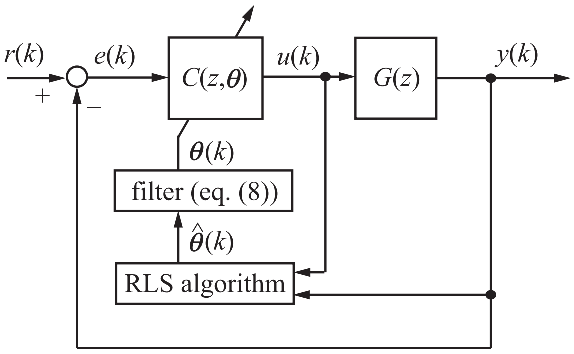

Fig. 2. Block diagram of online tuning.

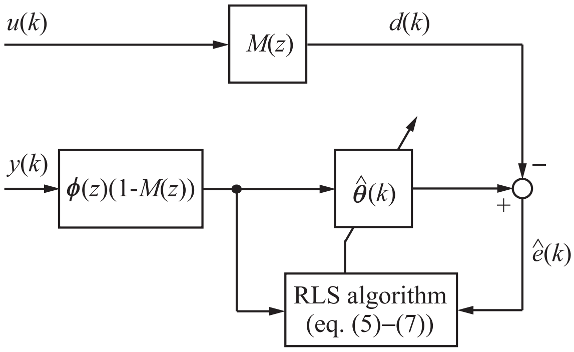

Fig. 3. Block diagram of RLS algorithm.

5. EXTENSION TO 2DOF CONTROL SYSTEMS

In this section, we extend the proposed method to online tuning of the feed-forward controller in a 2DOF control system by applying the results in Kaneko et al. (2011).

We consider the 2DOF control system illustrated in Fig. 4. In the figure, $C_{\text{fb}}(z)$ is a feedback controller which is assumed to be implemented so as to stabilize the closed-loop as in Kaneko et al. (2011). Also, $C_{\text{ff}}(z, \boldsymbol{\theta})$ is a feed-forward controller which is assumed to be linearly parameterized as follows:

$$C_{\text{ff}}(z, \boldsymbol{\theta}) = \boldsymbol{\theta}^T \boldsymbol{\psi}(z),$$

where $\boldsymbol{\theta} \in \mathbb{R}^n$ is a tunable parameter vector and $\boldsymbol{\psi}(z)$ is an n-dimensional rational function with real coefficients given by the designer.

In this system configuration, using the obtained input and output data $u(k)$, $y(k)$, $k = 1, \ldots, N$, we can express the fictitious reference signal as

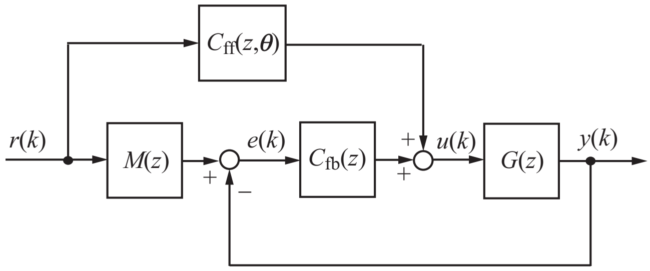

Fig. 4. 2DOF control system.

$$\tilde{r}(\boldsymbol{\theta}, k) = \frac{u(k) + C_{\mathrm{fb}}(z)y(k)}{C_{\mathrm{ff}}(z, \boldsymbol{\theta}) + M(z)C_{\mathrm{fb}}(z)}.$$

As stated in Section 3, the performance index to be minimized in FRIT is

$$J(\boldsymbol{\theta}) = \sum_{k=1}^{N} (y(k) - M(z)\tilde{r}(\boldsymbol{\theta}, k))^2.$$

In Kaneko et al. (2011), however, the error $y(k) - M(z)\tilde{r}(\boldsymbol{\theta}, k)$ evaluated in the performance index is modified as

$$\begin{aligned}\tilde{e}(k) &= (C_{\mathrm{ff}}(z, \boldsymbol{\theta}) + M(z)C_{\mathrm{fb}}(z))\, y(k) \\ &\quad - M(z)\,(u(k) + C_{\mathrm{fb}}(z)y(k)) \\ &= C_{\mathrm{ff}}(z, \boldsymbol{\theta})y(k) - M(z)u(k)\end{aligned}$$

and the following modified performance index is considered:

$$\tilde{J}(\boldsymbol{\theta}) = \sum_{k=1}^{N} \tilde{e}(k)^2.$$

Defining

$$\boldsymbol{\xi}(k) = \boldsymbol{\psi}(z)y(k), \tag{9}$$

we can express the modified error $\tilde{e}(k)$ as

$$\tilde{e}(k) = \boldsymbol{\theta}^T \boldsymbol{\xi}(k) - d(k),$$

which is the same form as (4). Therefore we can apply the online controller parameter tuning algorithm described in the previous section to the feed-forward controller tuning in the same way. In this case, the RLS with a forgetting factor is used, and therefore, the performance index to be minimized is described as follows:

$$\tilde{J}_k(\boldsymbol{\theta}) = \sum_{i=1}^{k} \lambda^{k-i} \tilde{e}(i)^2.$$

Also, (9) is used instead of (2) in the online controller parameter tuning algorithm in the previous section.

6. NUMERICAL EXAMPLE

We consider the following plant used in a benchmark problem in Suda (1992):

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

Let the sampling time be 0.001 s and the reference model be

$$M(z) = \frac{1.997 \cdot 10^{-6} z + 1.995 \cdot 10^{-6}}{z^2 - 1.996z + 0.996}$$

which is a discretized system of $M_c(s) = 1/(0.5s + 1)^2$ with a zero-order hold. The reference signal $r(k)$ is given by a rectangle wave taking values 0 and 1. Accordingly, the desired output is $y_d(k) = M(z)r(k)$. We apply the method proposed in Section 4 and set the initial controller parameter and the filter time constant of the approximate derivative by $\boldsymbol{\theta}(0) = [3, 0.5, 0]^T$ and $\tau = 0.1$, respectively. We provide white noise with zero mean and standard deviation 10^{-4} as measurement noise. We set the parameters in the algorithm as $\gamma = 10^2$, $\lambda = 1 - 5 \cdot 10^5$, and $\alpha = 10^{-4}$.

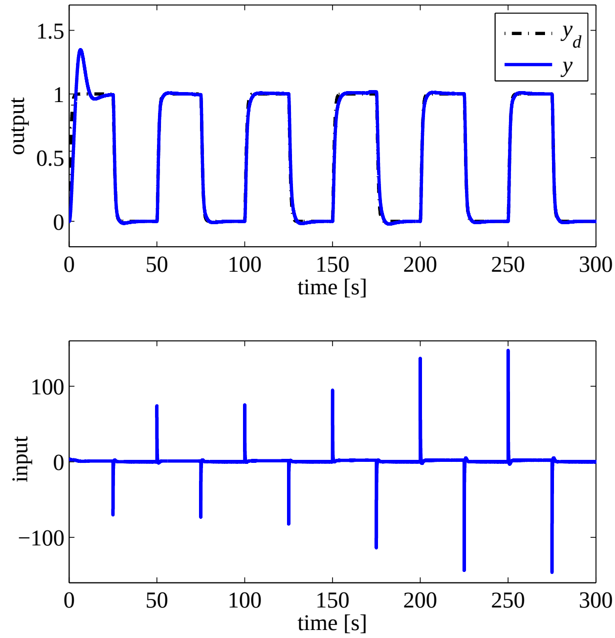

Fig. 5. Output and input in Case 1.

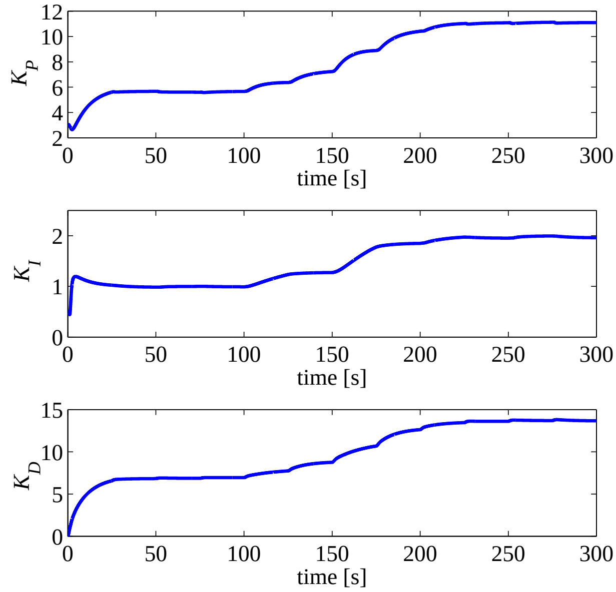

Fig. 6. PID gain in Case 1.

6.1 Case 1: Slow Change of Plant Characteristics

In order to evaluate the adaptation to characteristic variation, we consider the case where the gain of the plant is proportionally reduced to its half from 70 s to 170 s, i.e., the plant is changed according to

$$\left(-\frac{1}{200}(t - 70) + 1\right) G(s),$$

where t denotes the unit of time, the second.

The control input and output by the proposed method are shown in Fig. 5 and the transition of the PID gains is shown in Fig. 6. The tuned PID gains converge to constant values up to about 25 s, and good tracking performance is achieved. The PID gains obtained at 25 s are almost the same as those by the standard (offline) FRIT. After 70 s,

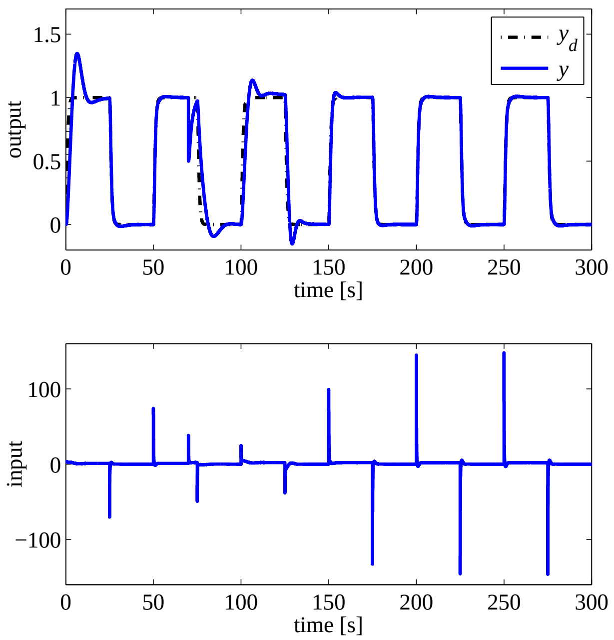

Fig. 7. Output and input in Case 2.

the PID gains are tuned adaptively in accordance with the variation of the plant characteristics, and good tracking performance is maintained. We see from the figures that the proposed method is effective to the slow change of the plant characteristics.

6.2 Case 2: Abrupt Change of Plant Characteristics

Next we consider the case where the gain of the plant is halved abruptly at 70 s, i.e., the plant is $1/2G(s)$ after 70 s. We have the simulation results shown in Figs. 7 and 8. Due to the characteristic variation of the plant at 70 s, the output is disturbed. After that, however, the PID gains are gradually tuned, and finally, good tracking performance is achieved again. It is apparent from the figures that the proposed method is effective to the abrupt change of the plant characteristics.

7. CONCLUSION

In this paper, we have proposed an online controller parameter tuning method by applying the standard FRIT and the RLS algorithm. Moreover, we have extended the proposed method to online tuning of a feed-forward controller in a 2DOF control system. We have shown some numerical examples to verify the effectiveness of the proposed method.

In comparison with the conventional FRIT which is offline tuning, the proposed algorithm can be performed continuously without stopping the control operation or without changing the controller parameter abruptly, which implies that the proposed method is more practical than the conventional one. However, we need to appropriately set some setting parameters in the algorithm. Future research directions are to investigate some guidelines of such an appropriate parameter setting and to verify the effectiveness of the proposed method by carrying out some experiments.

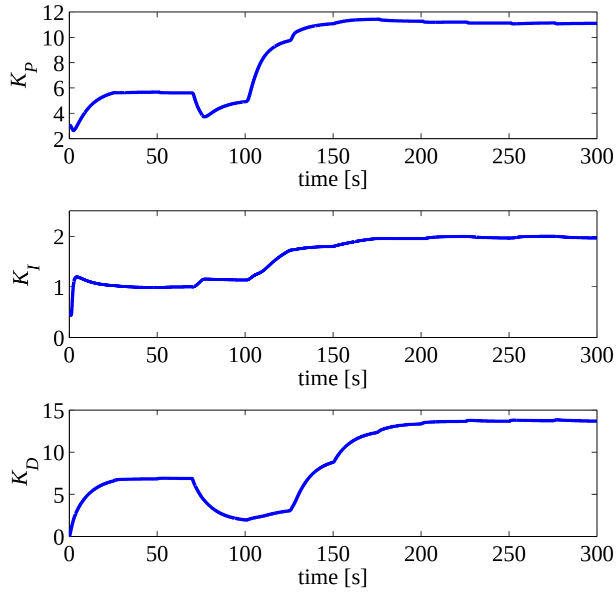

Fig. 8. PID gain in Case 2.

REFERENCES

Campi, M. and Savaresi, S. (2006). Direct nonlinear control design: the virtual reference feedback tuning (VRFT) approach. *IEEE Transactions on Automatic Control*, 51(1), 14–27.

Haykin, S. (2002). *Adaptive Filter Theory*. Prentice-Hall, Upper Saddle River, NJ, 4th edition.

Hjalmarsson, H. (2002). Iterative feedback tuning–an overview. *International Journal of Adaptive Control and Signal Processing*, 16, 373–395.

Kaneko, O., Yamashina, Y., and Yamamoto, S. (2011). Fictitious reference tuning of the feed-forward controller in a two-degree-of-freedom control system. *SICE Journal of Control, Measurement, and System Integration*, 4(1), 55–62.

Masuda, S. (2010). Adaptive PID control based on the online FRIT approach. *Proceedings of SICE 10th Annual Conference on Control Systems*, 16532 (in Japanese).

Souma, S., Kaneko, O., and Fujii, T. (2004). A new method of controller parameter tuning based on input-output data, –fictitious reference iterative tuning. *Proceedings of IFAC Workshop on Adaptation and Learning in Control and Signal Processing*, 789–794.

Suda, N. (1992). *PID Control*. Asakura Publishing Co., Ltd., Tokyo, Japan (in Japanese).

Tasaka, K., Kano, M., Ogawa, M., Masuda, S., and Yamamoto, T. (2009). Direct PID tuning from closed-loop data and its application to unstable processes. *Transaction of the Institute of Systems, Control and Information Engineers*, 22(4), 137–144 (in Japanese).

Wakasa, Y., Kanagawa, S., Tanaka, K., and Nishimura, Y. (2011). FRIT for systems with dead-zone and its application to ultrasonic motors. *IEEJ Transactions on Electronics, Information and Systems*, 131(6), 1209–1216.

Yamashina, Y., Kaneko, O., and Yamamoto S. (2011). Real-time parameter tuning for the feed-forward controller in a two-degree-of-freedom control system with FRIT. *Proceedings of the 10th SICE Control Division Conference*, 18412 (in Japanese).