FRIT and RLS-Based Online Controller Tuning and Its Experimental Validation

Yuji Wakasa, Azakami Ryo, Kanya Tanaka, and Shota Nakashima
Graduate School of Science and Engineering, Yamaguchi University
2-16-1 Tokiwadai, Ube, Yamaguchi 755-8611, Japan
Email: {wakasa, s049vk, ktanaka, s-naka}@yamaguchi-u.ac.jp

Abstract—**This paper proposes an online type of controller parameter tuning method by modifying the standard fictitious reference iterative tuning (FRIT) 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 feedforward controller of a two-degree-of-freedom control system. Finally, experimental results are provided to demonstrate the effectiveness of the proposed FRIT and RLS-based online controller tuning method.**

I. Introduction

For the last decade, some direct tuning methods of controller parameters such as proportional-integral-derivative (PID) gains have been investigated [1], [2], [3]. 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 [1] requires iterative experiments. In contrast, virtual reference feedback tuning (VRFT) proposed in [2] and fictitious reference iterative tuning (FRIT) proposed in [3] 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 [4]. 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., [5], [6], [7]).

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 [6], [8]. 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 [6], [8] 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 [6], controller parameters have to be updated periodically, so that the controller parameters may change considerably, thereby leading to control performance deterioration.

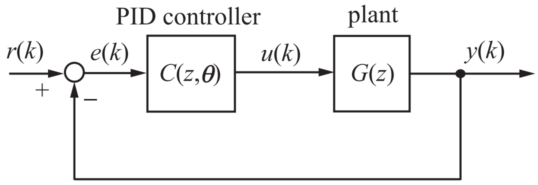

Fig. 1. System configuration.

In [9], the authors have proposed an online method which can resolve the abovementioned problems while experimental validation about the method has not been conducted yet.

This paper presents an online type of controller parameter tuning method by utilizing the so-called recursive least-squares (RLS) algorithm (see, e.g., [10]) 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 feedforward controller of a two-degree-of-freedom (2DOF) control system. Finally, experimental results are provided to illustrate the effectiveness of the proposed method.

II. Standard FRIT

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 $\boldsymbol{\theta} \in \mathbb{R}^n$ 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

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

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

978-1-4673-5769-2/13/$31.00 ©2013 IEEE

model of $\phi_c(s)$ by $\phi(z)$ and use the following discrete-time PID controller as a typical case:

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

In the standard FRIT [3], we first perform a closed-loop experiment to obtain input/output data $u_0(k)$, $y_0(k)$, $k = 1,\dots,N$, 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.

III. 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, i.e., $J(\boldsymbol{\theta}) = 0$:

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

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 [6]:

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

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 [6], 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 [10] 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:

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

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 $k-1$ [10]. 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., [10]).

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 as $\boldsymbol{P}(0) = \gamma I$, 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 $\boldsymbol{\theta}(k)$:

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

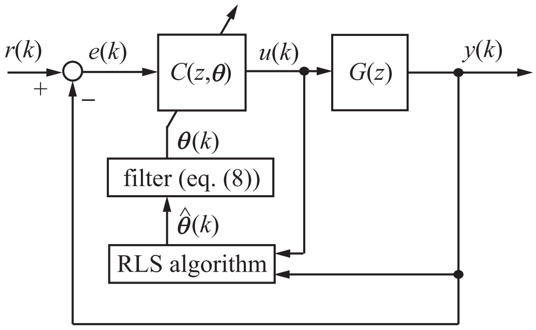

Fig. 2. Block diagram of online tuning.

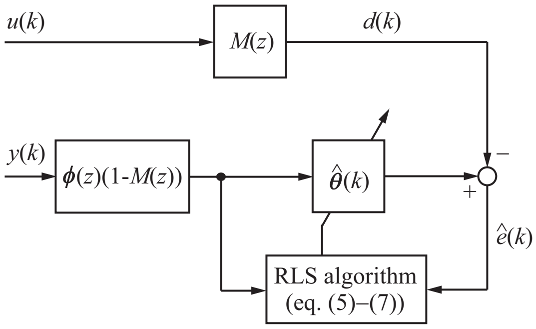

Fig. 3. Block diagram of RLS algorithm.

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 stability of the control system by utilizing information on a pre-experiment or plant model.

Remark 2: As is well-known, the RLS method requires $O(n^2)$ arithmetic operations per iteration, while the (non-recursive) least squares method, i.e., solving the normal equation corresponding to $\hat{J}_k(\boldsymbol{\theta})$, requires $O(n^3)$ complexity [10]. Therefore, the proposed method is more effective for a larger n, although there is no great difference between the arithmetic operations in both methods when the PID controller is used, i.e., $n = 3$.

IV. Extension to 2DOF Control Systems

In this section, we extend the proposed method to online tuning of the feedforward controller in a 2DOF control system by applying the results in [4].

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 [4]. Also, $C_{\text{ff}}(z,\boldsymbol{\theta})$ is a feedforward 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

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

As stated in Section II, 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 [4], however, the error $y(k) - M(z)\tilde{r}(\boldsymbol{\theta}, k)$ evaluated in the performance index is modified as

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

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 feedforward controller tuning in the same way. In this case, the RLS with a forgetting factor is

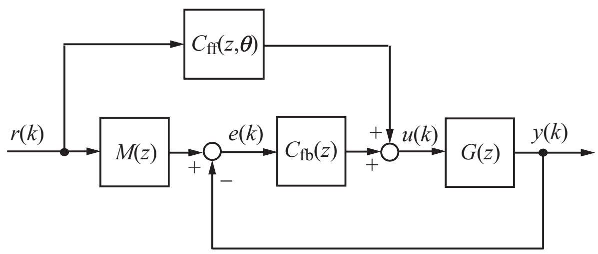

Fig. 4. 2DOF control system.

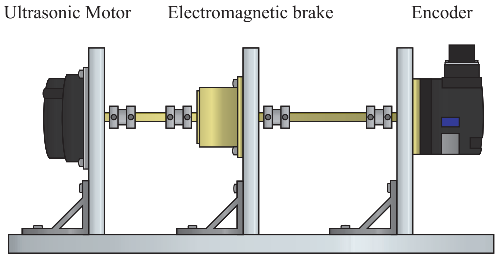

Fig. 5. Experimental ultrasonic motor system.

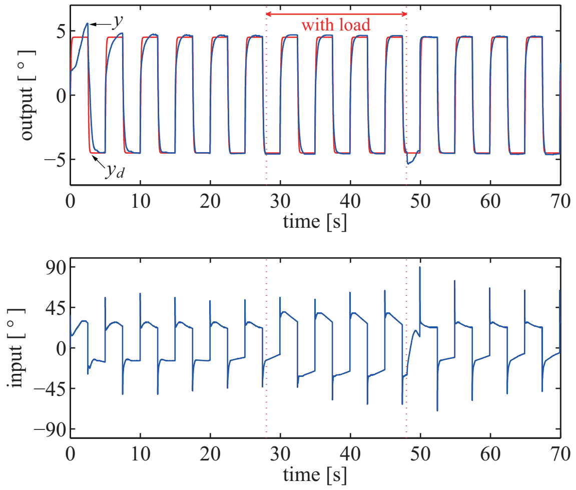

Fig. 6. Control input and output in the 1DOF case.

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.

V. Experimental Validation

In order to demonstrate the effectiveness of the methods presented in Sections III and IV, we carried out experiments using an ultrasonic motor system. In the experimental system, an ultrasonic motor (Canon UA60), electromagnetic brake, and encoder are connected to the same shaft, as shown in Fig. 5. The rotation angle of the ultrasonic motor (i.e., the control output) is obtained using the encoder, and the ultrasonic motor is driven by means of a phase difference scheme. The phase difference (i.e., the control input) can be adjusted between -90° and 90°, in steps of 1.406°. A load of 0.1 N·m can be generated using the electromagnetic brake. We suppose that this load can cause a characteristic change of the ultrasonic motor system.

We set sampling time as 0.001 s and a reference model $M(z)$ as a discretized system of $1/(0.05s+1)^2$ with a zero-order hold. The reference signal $r(k)$ is given by a rectangle wave taking values ± 4.5 with a period of 5 s. Accordingly, the desired output is denoted by $y_d(k) = M(z)r(k)$. We set the parameters in the algorithm as $\gamma = 10^2$, $\lambda = 1 - 5 \cdot 10^{-4}$, and $\alpha = 10^{-4}$.

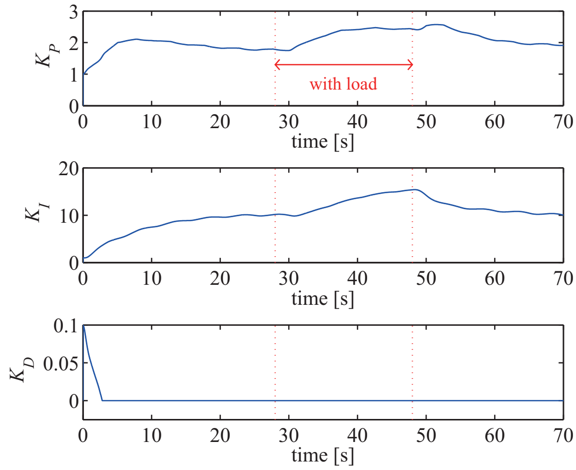

Fig. 7. PID gains in the 1DOF case.

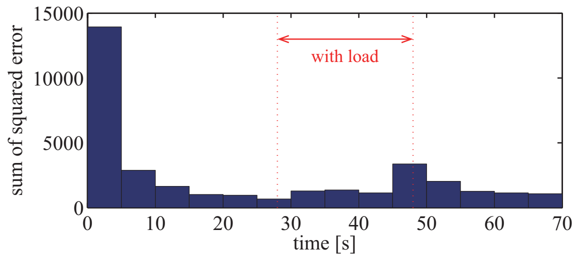

Fig. 8. Sum of squared errors in the 1DOF case.

A. Case of the 1DOF Control System

We first deal with the online PID controller tuning in the 1DOF control system configuration which is proposed in Section III. We set initial PID gains as $\boldsymbol{\theta}(0) = [1.0, 1.0, 0.1]^T$ and a filter time constant of the approximate derivative as $\tau = 0.1$. The PID gains $\boldsymbol{\theta}(k)$ are limited so that they should take non-negative values. In order to evaluate the adaptation to characteristic variation, we consider the case where the load is imposed between 28 s and 48 s.

The control input and output by the proposed method are shown in Fig. 6, and the transition of each PID gain is shown in Fig. 7. Moreover, the transition of the sum of squared errors between y_d and y at each period is shown in Fig. 8. We see from the figures that, although the overshoot of the control output occurs at the beginning of the experiment, the tracking performance is gradually improved. The PID gains converge to almost constant values up to about 20 s. The deterioration of the control output cannot be seen after the load is imposed at 28 s, while the PID gains are gradually adjusted. Although the performance deterioration can be seen when the load is canceled at 48 s, we once again obtain good tracking performance around 60 s. It is evident from the abovementioned observations that the proposed method is effective to the change of the plant characteristics.

B. Case of the 2DOF Control System

We next consider the online tuning of the feedforward controller in the 2DOF control system configuration which is proposed in Section IV. We provide a proportional feedback

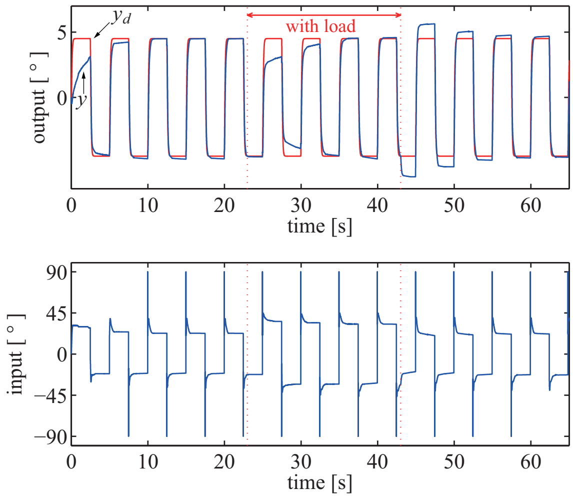

Fig. 9. Control input and output in the 2DOF case.

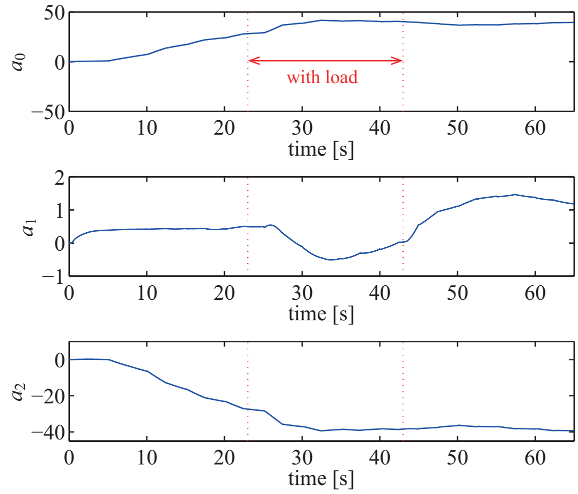

Fig. 10. Controller coefficients in the 2DOF case.

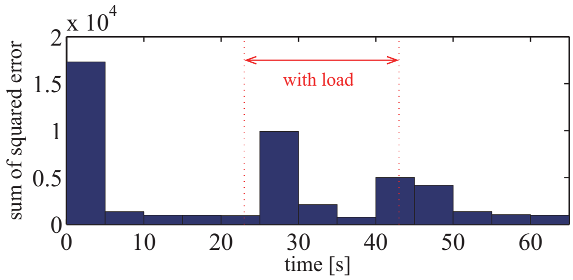

Fig. 11. Sum of squared errors in the 2DOF case.

controller $C_{\text{fb}}(z) = K_P = 2.0$ such that the resulting close-loop system is stabilized. However, the transient performance of the closed-loop system is not sufficiently good by means of the above proportional control. To improve the tracking performance, we add the feedforward controller, as shown in Fig. 4, and set its structure as the following FIR model:

$$C_{\text{ff}}(z, \boldsymbol{\theta}) = a_0 + \frac{a_1}{z} + \frac{a_2}{z^2},$$

where $\boldsymbol{\theta} = [a_0, a_1, a_2]^T$. We set initial coefficients as $\boldsymbol{\theta}(0) = [0, 0, 0]^T$. In order to evaluate the adaptation to characteristic variation, we impose the load between 28 s and 43 s.

The control input and output by the proposed method are shown in Fig. 9, and the transition of each PID gain is shown in Fig. 10. Moreover, the transition of the sum of squared errors between y_d and y at each period is shown in Fig. 11. It is seen from the figures that the tracking performance is not good at the beginning of the experiment and just after the imposition and cancellation of the load. After these occasions, however, the tracking performance is gradually improved. It is evident from this observation that the proposed method for the 2DOF control system configuration is also effective to the change of the plant characteristics.

VI. Conclusion

In this paper, we have presented 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 feedforward controller in a 2DOF control system. We have shown some experimental results to verify the effectiveness of the proposed method.

We need to appropriately set some setting parameters in the proposed algorithm. However, 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.

References

[1] H. Hjalmarsson, "Iterative feedback tuning – An overview," *International Journal of Adaptive Control and Signal Processing*, vol. 16, pp. 373–395, 2002.

[2] M. C. Campi, A. Lecchini, and S. M. Savaresi, "Virtual reference feedback tuning: a direct method for the design of feedback controllers," *Automatica*, vol. 38, pp. 1337–1346, 2002.

[3] S. Souma, O. Kaneko, and T. Fujii, "A new method of controller parameter tuning based on input-output data – Fictitious reference iterative tuning," *Proc. IFAC Workshop on Adaptation and Learning in Control and Signal Processing (ALCOSP 04)*, pp. 789–794, 2004.

[4] O. Kaneko, Y. Yamashina, and S. Yamamoto, "Fictitious reference tuning of the feed-forward controller in a two-degree-of-freedom control system," *SICE Journal of Control, Measurement, and System Integration*, vol. 4, no. 1, pp. 55–62, 2011.

[5] K. Tasaka, M. Kano, M. Ogawa, S. Masuda, and T. Yamamoto, "Direct PID tuning from closed-loop data and its application to unstable processes," *Trans. Institute of Systems, Control and Information Engineers*, vol. 22, no. 4, pp. 137–144, 2009 (in Japanese).

[6] S. Masuda, "Adaptive PID control based on the online FRIT approach", *Proc. SICE 10th Annual Conference on Control Systems*, 16532, 2010 (in Japanese).

[7] Y. Wakasa, S. Kanagawa, K. Tanaka, and Y. Nishimura, "FRIT for systems with dead-zone and its application to ultrasonic motors," *IEEJ Trans. Electronics, Information and Systems*, vol. 131, no. 6, pp. 1209–1216, 2011.

[8] Y. Yamashina, O. Kaneko, and S. Yamamoto, "Real-time parameter tuning for the feed-forward controller in a two-degree-of-freedom control system with FRIT," *Proc. SICE 11th Annual Conference on Control Systems*, 18412, 2011 (in Japanese).

[9] Y. Wakasa, K. Tanaka, and Y. Nishimura, "Online controller tuning via FRIT and recursive least-squares," *Proc. IFAC Conf. Advances in PID Control*, 2012.

[10] S. Haykin, *Adaptive Filter Theory*, Prentice-Hall, 4th edition, 2002.