A Recursive Tuning Approach for the Model-Free PID Controller Design

Jiadong Wang [*] **Yanhui Zhang** [*] **Xiaoming Jin** [*] **Hongye Su** [**]

[*] *Zhejiang SUPCON Software Co. Ltd., Hangzhou, 310053, China (e-mail: wangjiadong1@supcon.com).*
[**] *Zhejiang University, Hangzhou, China, (e-mail: suhongye@zju.com)*

Abstract: This paper presents a new model-free tuning approach for the PID controller tuning with the employment of the refined recursive instrumental variable (RIV) method. The proposed approach can be applied to solve the control loop tuning problem without identification of the plant or process model and it can be implemented in the online manner. Also, the colored measurement noise has been taken into account. For PI and PID control, the implementation details and the step by step procedures are provided respectively. Two simulation examples are provided to validate the effectiveness of the proposed approach.

Keywords: PID, Controller Tuning, Refined Recursive Instrumental Variable, Model-free, Data-driven.

1. INTRODUCTION

Since the middle of last century, the proportional-integral-derivative (PID) controllers have become the most popular controllers in different process industries and nowadays PID controllers still dominate the industrial control area. Over the decades, people have made great efforts on the research of PID controllers; numerous books and papers can be referenced on how to design a PID controller. Moreover, many commercial software, such as, Emerson DeltaV Insight, Honeywell Profit PID, Supcon PID-Suite, etc., were published to assist engineers in PID controller design. Despite the progress made in the previous work, PID tuning is still often an non-trivial task in industry, especially for the users without control background. One important reason is that the traditional PID tuning approaches are either model based or rely on the testing data with sufficient input excitation. However, in many situations, people are not allowed or impossible to carry out the required tests, which brings with the development of model-free tuning approaches for the PID controller design.

Regarding to the model-free tuning methods, some important work can be referenced. For example, Hjalmarsson et al. (1998) formulated the iterative PID tuning task as a control parameter optimization problem and solved the problem by using a Gauss-Newton scheme; Guardabassi et al. (2000) proposed an off-line virtual reference feedback tuning (VRFT) approach, the idea of which is to interpret the open-loop I/O data as closed-loop data produced by a virtual reference signal; based on this work, Campi et al. (2002) refined the VRFT method by giving more emphasis on the implementation details and performance issues; Hayashi et al. (2011) proposed a so-called one-shoot tuning scheme with simplified procedure, which enables users to tune PID gains with only operating I/O data. Gao et al. (2017) combined controller tuning with performance assessment and convert the tuning problem to be a convex optimization problem.

This paper focuses on solving the model-free tuning problem from the industrial perspectives. In practice, one fact is that some industrial processes are intrinsically time-varying and a few of them are fast-varying systems. Thus, it is necessary to update the PID parameters as soon as the process deviates significantly. Another fact is that there exist practical needs to tune a large number of control loops at the same time, which would consume great computational resource if every single tuning process is time-consuming and occupies large amount of computational resource.

In this paper, we aims at solving these issues by proposing a model-free tuning approach with recursive implementation procedures. Following the idea of the VRFT methods, we first formulate the tuning problem as an optimization problem; then we discuss and show the characteristics of the colored measurement noise. To reduce the negative effect of the noise, we employ the refined instrumental variable (RIV) method to solve the parameter optimization problem. For PI and PID control, different notations and implementation steps are also provided.

The remaining of this paper is organized as follows: Section 2 addresses the tuning objective and presents the idea of the proposed model-free tuning approach; Section 3 details the recursive implementation of the tuning approach; Section 4 shows some simulation results to demonstrate the effectiveness of the algorithms. Section 5 concludes this research work.

2. THE MODEL-FREE PID TUNING APPROACH

2.1 Tuning Objective

In this paper, we consider the control systems as described in Fig. 2.1, see below, where G_{cl} and G^{*}_{cl} are respectively

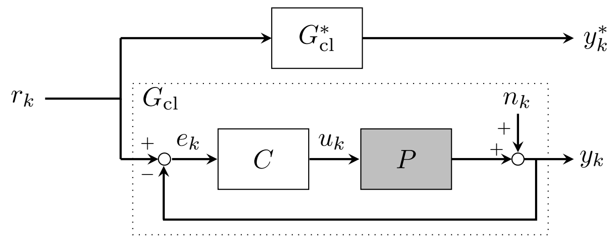

Fig. 1. The control system diagram.

the actual and desired closed loop control systems. P is short for plant, which is unknown and could be difficult or impossible to be modeled accurately with limited number of operating data. C here represents the standard PID controller, which can be written in the following discrete-time form,

$$C(z^{-1}) = K + T_i\frac{T_s z^{-1}}{1-z^{-1}} + T_d\frac{1-z^{-1}}{T_s z^{-1}}, \tag{1}$$

where K, T_i, T_d are the controller parameters.The tuning objective is to design the three parameters so that G_{cl} approaches G^*_{cl} as close as possible. Note that, we here focus on the development of a tuning approach without *a priori* knowledge of P or estimating a plant model beforehand. Due to the lack of the information of G_{cl}, we aim at minimizing $\|y_k - y^*_k\|_\ell$ instead of minimizing $\|G_{\text{cl}} - G^*_{\text{cl}}\|_\ell$. The notation $\|\cdot\|_\ell$ stands for ℓ-norm and we let $\ell = 2$ in our work.

2.2 Tuning Approach

Bearing the tuning objective in mind, we next show the main idea of designing the PID controller in (1), that is, selecting appropriate parameters K, T_i and T_d, in the least squares sense. We start by writing the reference signal r_k in the following form,

$$\begin{aligned} r_k &= \Big(K + T_i\frac{T_s z^{-1}}{1-z^{-1}} + T_d\frac{1-z^{-1}}{T_s z^{-1}}\Big)^{-1} u_k + y_k, \\ &= \Big[\frac{K(1-z^{-1})T_s z^{-1} + T_i T_s^2 z^{-2} + T_d(1-z^{-1})^2}{T_s z^{-1}(1-z^{-1})}\Big]^{-1} u_k + y_k, \\ &= \frac{T_s z^{-1} - T_s z^{-2}}{T_d + (KT_s - 2T_d)z^{-1} + (T_i T_s^2 + T_d - KT_s)z^{-2}} u_k + y_k. \end{aligned} \tag{2}$$

Substituting (2) in $y^*_k = G^*_{\text{cl}} r_k$, the desired output y^*_k then becomes,

$$y^*_k = \frac{G^*_{\text{cl}}(T_s z^{-1} - T_s z^{-2})}{T_d + (KT_s - 2T_d)z^{-1} + (T_i T_s^2 + T_d - KT_s)z^{-2}} u_k + G^*_{\text{cl}} y_k. \tag{3}$$

Before proceeding, we define two filtered input and output signals,

$$\begin{aligned} u^f_k &\triangleq G^*_{\text{cl}}(1-z^{-1})u_k, \\ y^f_k &\triangleq (1-G^*_{\text{cl}})y_k, \end{aligned} \tag{4}$$

and a filter that contains K, T_i and T_d,

$$\begin{aligned} F_c(z^{-1}) &\triangleq f_0 + f_1 z^{-1} + f_2 z^{-2}, \\ &\triangleq T_d + (KT_s - 2T_d)z^{-1} + (T_i T_s^2 + T_d - KT_s)z^{-2}. \end{aligned} \tag{5}$$

Applying the definitions in (4) and (5), $\|y_k - y^*_k\|_2^2$ is then given by,

$$\|y_k - y^*_k\|_2^2 = \|y^f_k - \frac{T_s z^{-1}}{F_c(z^{-1})}u^f_k\|_2^2.$$

Therefore, the tuning problem can be converted to the least squares format,

$$\min_{f_0,\ f_1,\ f_2} \|y^f_k - \frac{T_s z^{-1}}{F_c(z^{-1})}u^f_k\|_2^2. \tag{6}$$

For given f_0, f_1 and f_2, the controller parameters, *i.e.*, K, T_i and T_d, are computed by,

$$\begin{bmatrix} K \\ T_i \\ T_d \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ T_s & 0 & -2 \\ -T_s & T_s^2 & 1 \end{bmatrix}^{-1} \begin{bmatrix} f_0 \\ f_1 \\ f_2 \end{bmatrix}. \tag{7}$$

It is remarked that the filtered signals u^f_k and y^f_k are both known and the defined filter $F_c(z^{-1})$ contains all the controller parameters to design. Moreover, the matrix in (7) is always non-singular for $T_s > 0$.

3. THE RECURSIVE IMPLEMENTATION

Based on the proposed tuning approach, we will now provide the recursive procedure to solve the optimization problem in (6) with consideration of measurement noise added in the output channel, see Fig. 2.1. We shall first look at the characteristics of noise and then introduce the refined recursive instrumental variables (RIV) algorithm to calculate the tuning parameters.

3.1 Characteristics of Noise

Taking the measurement noise into account, we can write y_k as the sum of the following two terms:

$$y_k = \frac{PC}{PC+1}r_k + \frac{1}{PC+1}n_k, \tag{8}$$

where n_k is supposed equal to $\frac{L(z^{-1})}{M(z^{-1})}\varepsilon_k$ and ε_k is a white noise. From the definition in (4), it is clear that the noise part of y^f_k can be represented in the colored noise form of $\frac{L'(z^{-1})}{M'(z^{-1})}\varepsilon_k$, where the noise filters $L'(z^{-1})$ and $M'(z^{-1})$ are defined as

$$\begin{aligned} \frac{L'(z^{-1})}{M'(z^{-1})} &\triangleq \frac{(1-G^*_{\text{cl}})L(z^{-1})}{(PC+1)M(z^{-1})}, \\ &\triangleq \frac{l_0 + l_1 z^{-1} + l_2 z^{-2} + \cdots + l_{n_l} z^{-n_l}}{1 + m_1 z^{-1} + m_2 z^{-2} + \cdots + m_{n_m} z^{-n_m}}. \end{aligned} \tag{9}$$

Note that, in the presence of the colored noise, the standard least squares methods cannot maintain the estimation consistency in solving the optimization problem in (6) and the estimated controller parameters may give unsatisfactory control performance. Therefore, in this case, we'll consider an alternative method, *i.e.*, the refined IV method, to estimate tuning parameters.

3.2 Definitions

Before proceeding, below we define two *parameter vectors* to be used later. For the time being, we consider the case of PID control, *i.e.*, $f_0 \neq 0$.

$$\theta \triangleq \left[\frac{f_1}{f_0},\, \frac{f_2}{f_0},\, \frac{T_s}{f_0}\right]^{\mathrm{T}}, \tag{10}$$

$$\eta \triangleq [m_1,\, \cdots,\, m_{n_m},\, l_0,\, l_1,\, \cdots,\, l_{n_l}]^{\mathrm{T}}. \tag{11}$$

θ contains all the controller parameters and η contains all the noise filter parameters. The estimated parameter vector at time k will be noted as $\hat{(\cdot)}_k$, *e.g.*, $\hat{\theta}_k$ is the estimate of θ at time k.

In addition, we introduce another two pre-filtered input and output signals: u_k^{f*} and y_k^{f*}, which are expressed as,

$$u_k^{f^*} \triangleq \frac{\hat{M}'(z^{-1})}{\hat{L}'(z^{-1})\hat{F}_c(z^{-1})} u_k^f, \tag{12}$$

$$y_k^{f^*} \triangleq \frac{\hat{M}'(z^{-1})}{\hat{L}'(z^{-1})\hat{F}_c(z^{-1})} y_k^f. \tag{13}$$

The purpose of this filtering is to decompose the estimation of θ and η into two parts, which will be seen soon after. The input and output signals are both pre-filtered by the noise filters and so the noise effect is filtered out at each recursive step.

Finally, we define several *data vectors* to simplify notations.

$$\psi_k \triangleq \left[-\hat{x}^*_{k-1},\, -\hat{x}^*_{k-2},\, u^{f^*}_{k-1}\right], \tag{14}$$

$$\phi_k \triangleq \left[-y^{f^*}_{k-1},\, -y^{f^*}_{k-2},\, u^{f^*}_{k-1}\right], \tag{15}$$

$$\omega_k \triangleq [-\hat{\varepsilon}_{k-1},\, \ldots,\, -\hat{\varepsilon}_{k-n_m},\, \hat{e}_k,\, \ldots,\, \hat{e}_{k-n_l}], \tag{16}$$

where $\hat{x}_k^*$ is the so-called instrumental variable (IV) defined as

$$\hat{x}^*_k \triangleq \frac{T_s z^{-1}}{\hat{F}_c(z^{-1})} u_k^{f^*} = \psi_k \hat{\theta}_k,$$

and $\hat{\varepsilon}_k \triangleq y_k^f - \hat{x}_k$.

3.3 The Recursive RIV Algorithm

The recursive RIV algorithm applied here stems from the work in Young et al. (1980). The algorithm below includes the initialization and iteration two stages. At the iteration stage, PID controller parameters will be firstly estimated (see steps 3-6) and the noise filter parameters are then calculated (see steps 7-9).

Algorithm 1 PID Control

Initialization:

1: Generate initial parameter vector $\hat{\theta}_0$ with existing PID controller parameters and generate $\hat{\eta}_0$ with random parameters.

2: Set the initial covariance matrices $P_0 = p\mathrm{I}$ and $Q_0 = q\mathrm{I}$ with $p, q \gg 1$.

Iteration:

3: Compute $u_k^{f^*}$ and $y_k^{f^*}$ with (4), (12) and (13).

4: Compute the IV variable $\hat{x}_k^*$.

5: Construct the data vectors ψ_k, ϕ_k using (14)-(15).

6: Update the PID controller parameters,

$$\hat{e}_k = \hat{F}_c(z^{-1}) y_k^{f^*} - T_s u_{k-1}^{f^*},$$

$$P_k = P_{k-1}\Big(\mathrm{I} - \frac{\psi_k^T \phi_k P_{k-1}}{1 + \phi_k P_{k-1} \psi_k^T}\Big),$$

$$\hat{\theta}_k = \hat{\theta}_{k-1} + P_k \psi_k^T \hat{e}_k.$$

7: Compute $\hat{\varepsilon}_k = y_k^f - \hat{x}_k$.

8: Construct the data vectors ω_k using (16).

9: Update the noise filter parameters,

$$Q_k = Q_{k-1}\big(\mathrm{I} - \frac{\omega_k^T \omega_k Q_{k-1}}{1 + \omega_k Q_{k-1} \omega_k^T}\big),$$

$$\hat{\eta}_k = \hat{\eta}_{k-1} + Q_k \omega_k^T (\hat{M}'(z^{-1})\hat{\varepsilon}_k - \hat{L}'(z^{-1})\hat{e}_k).$$

3.4 PI Control

The above we have discussed the case of PID control. In the case of PI control, *i.e.*, $f_0 = 0$ and thus $F_c(z^{-1})$ reduces to,

$$F_c(z^{-1}) = T_s z^{-1}[K + (T_i T_s - K)z^{-1}]. \tag{17}$$

By cancelling the common factor of $T_s z^{-1}$ in (6), the optimization problem can be rewritten into,

$$\min_{f_0',\, f_1'} \|y_k^f - \frac{1}{F_c'(z^{-1})} u_k^f\|_2^2. \tag{18}$$

where F_c' is defined as

$$\begin{aligned} F_c'(z^{-1}) &\triangleq f_0' + f_1' z^{-1}, \\ &\triangleq K + (T_i T_s - K)z^{-1}. \end{aligned} \tag{19}$$

Moreover, the corresponding parameter vector of θ needs to be redefined as,

$$\theta' \triangleq \left[\frac{f_1'}{f_0'},\, \frac{1}{f_0'}\right]^{\mathrm{T}}, \tag{20}$$

and likewise ψ_k and ϕ_k are replaced by,

$$\psi_k' \triangleq \left[-\hat{x}^*_{k-1},\, u_k^{f^*}\right], \tag{21}$$

$$\phi_k' \triangleq \left[-y^{f^*}_{k-1},\, u_k^{f^*}\right]. \tag{22}$$

For the case of PI control, the recursive RIV algorithm can be obtained by replacing θ, ψ_k and ϕ_k with their counterparts as above defined. The algorithm details are similar as Algorithm 1 and hence omitted here for brevity.

4. SIMULATION RESULTS

In this section, we show two examples to validate the effectiveness of the proposed approach. The first example is a dynamic system written in the form of the first-order-plus-dead-time (FOPDT) model,

$$G_0(s) = \frac{0.2}{10s + 1} e^{-5s},$$

where the time constant is 10 and the time delay is 5. The noise filter is,

$$\frac{L(z^{-1})}{M(z^{-1})} = \frac{0.6993 - 0.1107z^{-1}}{1 - 0.7555z^{-1} + 0.04979z^{-2}}.$$

where the $e(s)$ is a white noise signal with zero mean and its variance equal to 1. The signal to noise ratio (SNR) is approximately 30dB.

The aim is to re-tune the PID parameters such that the closed loop system settles down within 80 seconds. To achieve this goal, we pick the target system G^*_{cl} as

$$G^*_{\mathrm{cl}} = \frac{0.0059}{1 - 1.846z^{-1} + 0.8521z^{-2}}, \tag{23}$$

The input and output signals are collected and displayed in Fig. 2. Fig. 3 displays the filtered input and output signals used in the recursive RIV algorithm.

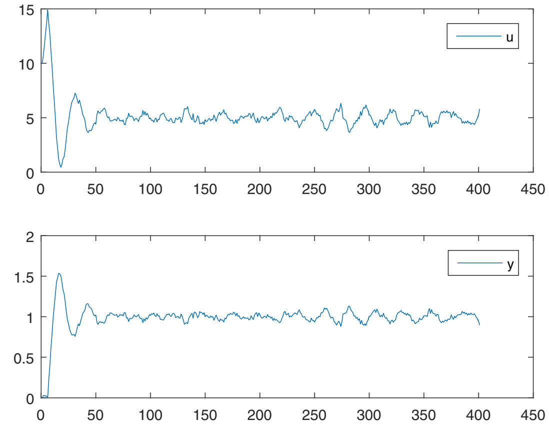

Fig. 2. The input signals and output signals

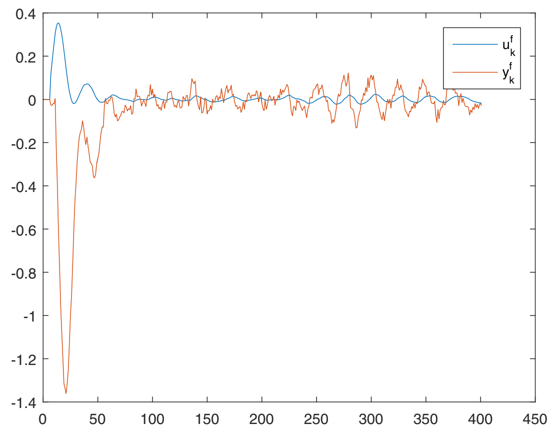

Fig. 3. The filtered signals of u^f_k and y^f_k

Fig. 4 shows the step response of the closed loop system with the initial and updated controller parameters. By using the new control parameters, the closed loop system performs as desired, although the collected data is relatively noisy and the system has a time delay.

The second example shows the level control of a water tank, see Fig. 5. Water is pumped into the tank at the top at rate of flow of $ku(t)$ $\mathrm{m}^3/\mathrm{sec}$ and the water flows out of the tank through a hole at the bottom at the rate of flow of $a\sqrt{2gy}$ $\mathrm{m}^3/\mathrm{sec}$, where a, y and g are respectively the area of the bottom hole, the water level and the gravitational acceleration. A is the cross-sectional area of the tank.

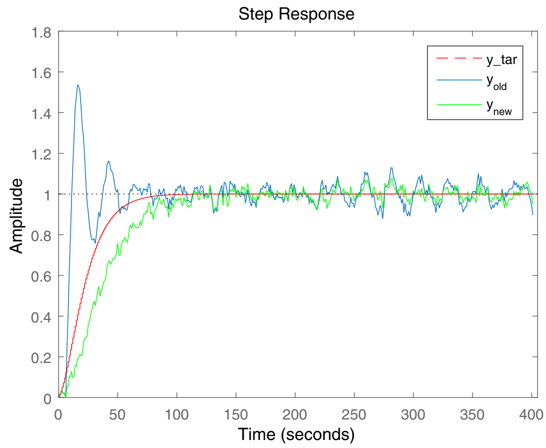

Fig. 4. The step response of the closed loop system with initial and updated controller parameters.

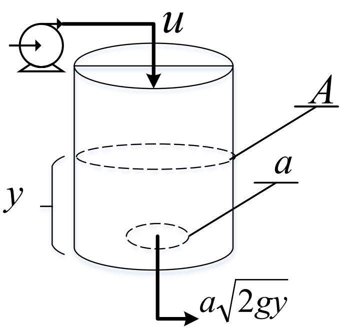

Fig. 5. The tank level control

From the mass conservation principle, we get

$$A\dot{y} = ku - a\sqrt{2gy}. \tag{24}$$

The parameters in the above equation are

- $A = 2.3 \times 10^{-3}$ m^2
- $A = 7.1 \times 10^{-6}$ m^2
- $g = 9.82$ $\mathrm{m}/\mathrm{sec}^2$
- $k = 3.9 \times 10^{-6}$ $\mathrm{m}^3/(\mathrm{sec}\cdot\mathrm{V})$

It can be seen that the target system is nonlinear. We employ a PI controller to control the level. The initial PI parameters are $K = 1 \times 10^{-6}$ and $T_i = 1 \times 10^{-5}$. The sampling time is 1 second. The measurement noise is simulated with the following filter and the signal to noise ratio is around 40 dB,

$$\frac{L(z^{-1})}{M(z^{-1})} = \frac{0.3712 - 0.2011z^{-1}}{1 - 0.1147z^{-1} + 0.1353z^{-2}}. \tag{25}$$

In this example, G^*_{cl} is chosen to be in the following form,

$$G^*_{\mathrm{cl}} = \frac{8.6578 \times 10^{-5}}{1 - 1.9869z^{-1} + 0.9870z^{-2}}, \tag{26}$$

which corresponds to approximately 400 seconds rising time, see Fig. 6.

Fig. 7 shows the step response of the closed-loop response with the initial PI parameters and the newly tuned PI

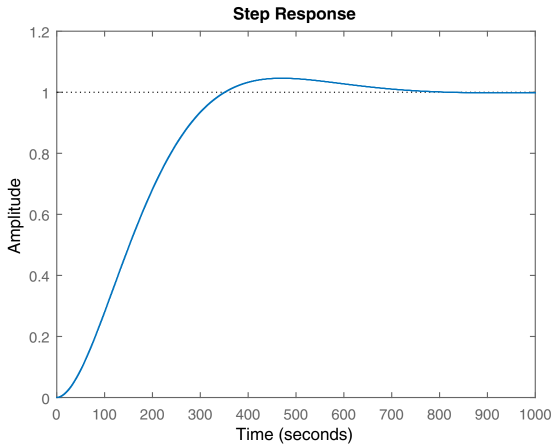

Fig. 6. The step response of the target system

parameters. We observe that the tuned PI controller

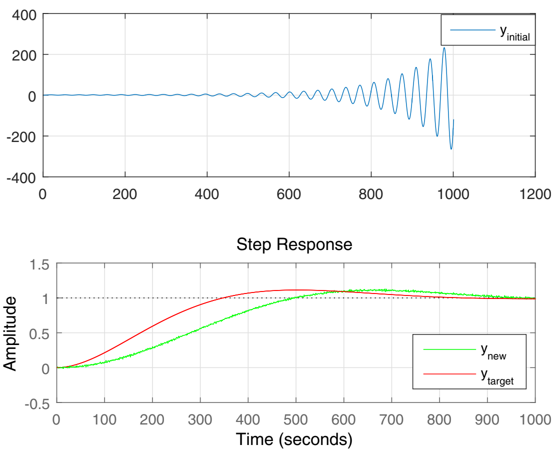

Fig. 7. The step response of the closed loop system with initial and updated controller parameters.

works very well although the initial PI parameters give unstable step response. Moreover, it is worth pointing out that the target system is nonlinear, but the proposed algorithm is still able to catch the system characteristics. Compared with the model-based tuning methods, the proposed tuning method can avoid the risk of model mismatch that usually happens at the identification step.

5. CONCLUSION

This paper has briefly reviewed the model-free tuning of feedback controllers. The practical issues were addressed and we proposed a new model-free tuning approach to compute PID parameters with recursive implementation procedures. The side effect of measurement noise was analyzed. A recursive RIV algorithm was provided. The simulation examples have shown that the proposed method can be applied to nonlinear systems and the system with time delay. Especially, when initial PID parameters are badly tuned, the proposed algorithm still works fine.

The research on the model-free tuning problem is still at the developing stage and there are many issues that have not fully explored. We would like to provide two interesting ones that are valuable to be further studied. One is to incorporate delay estimation in the recursive tuning procedures. In process industries, most processes contain delays and some of them contain very large time delay. Although the proposed algorithm works well for systems with small time delay, it becomes inaccurate for systems with large latency. Therefore, combining delay estimation procedure is of great importance. Second, more attention should be paid to the disturbance attenuation. As well known, the robustness of the control systems is of top priority for end-users. When disturbance exists, it is necessary to take some additional treatment to extract the disturbance information and hence improve the control performance.

REFERENCES

H. Hjalmarsson, M. Gevers, S. Gunnarsson and O. Lequin. Iterative Feedback Tuning : Theory and Application. *IEEE Control System Magazines*, 18: 26–41, 1998.

M. C. Campi, A. Lecchini and S. M. Savaresi. Virtual Reference Feedback Turning: A Direct Method for the Design of Feedback Controllers. *Automatica*, 38: 1337–1346, 2002.

G. O. Guardabassi and S. M. Savaresi. Virtual reference direct design method: an off-line approach to data-based control system design. *IEEE Transactions on Automatic Control*, 45(5): 954C-959, 2000.

K. Hayashi and T. Yamamoto. Design of a One-Shoot Tuning PID Controller. *IEEE International Conference on Control Applications*, 38: 712–717, 2011.

X. Gao, F. Yang, C. Shang and D. Huang. A Novel Data-Driven Method for Simultaneous Performance Assessment and Retuning of PID Controllers. *Industrial and Engineering Chemistry Research*, 56(8): 2127–2139, 2017.

P. Young and A. Jakeman. Refined instrumental variable methods of recursive time-series analysis Part III. Extensions. *International Journal of Control*, 31(4): 741–764, 1980.