Advances in Differential Equations and Control Processes

http://www.pphmj.com
http://dx.doi.org/10.17654/0974324322029
Volume 28, 2022, Pages 153-182 **P-ISSN: 0974-3243**

THE ROBUST PID CONTROLLERS FOR SPECIAL PROCESSES

Do Cao Trung

School of Mechanical Engineering (SME)
Hanoi University of Science and Technology (HUST)
Vietnam

Abstract

The paper presents the tuning method for the PID (Proportional Integral Derivative) of special processes consisting of self-balance with overshoot and self-imbalance with inverse response. It is a continued study of process identification by numerical method and PID controller design based on robust viewpoint. The self-balance process with overshoot is identified by the second order plus dead time with a negative zero (SOPDTZ), while the self-imbalance process with inverse response is modeled by integrating plus first order plus dead time (IFOPDT). To gain robust PID for processes, it requires typical designs. While the SOPDTZ needs a filter of first order lag, the IFOPDT requires to supplement integrating element for controller. For illustration, the detail identification and tuning procedures are presented via an example for each of the processes.

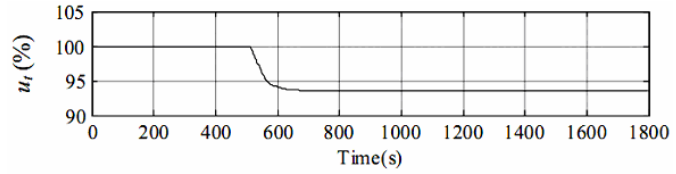
Received: April 20, 2022; Accepted: June 27, 2022
2020 Mathematics Subject Classification: 93C83.
Keywords and phrases: self-balance process with overshoot, self-imbalance process with inverse response, robust control, anti-disturbance.

How to cite this article: Do Cao Trung, The robust PID controllers for special processes, Advances in Differential Equations and Control Processes 28 (2022), 153-182.
http://dx.doi.org/10.17654/0974324322029

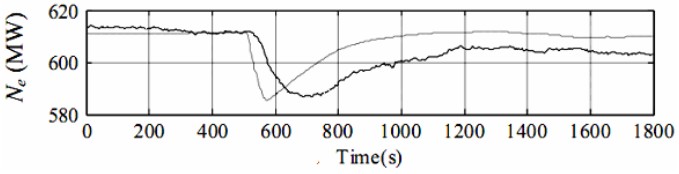
Published Online: August 22, 2022

1. Introduction

The open-loop identification is based on process response of step input. The response curve of overshoot process shows an output exceeding its final steady state value. Figure 1(a) presents the output capacity response of a 1200MW thermal power plant when the plant was excited by an input pulse of fuel. Meanwhile, Figure 1(b) expresses the response of output capacity of 300MW thermal power plant with input pulse being the open rate of steam control valve.



(a)



(b)

Figure 1(a). Unit load responses to a decreased pulse step of fuel flow-rate [3].

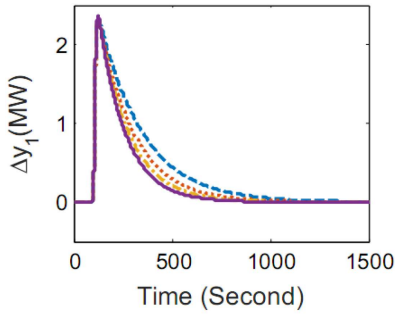


Figure 1(b). Unit load responses to an increased pulse step of control valve open rate [4].

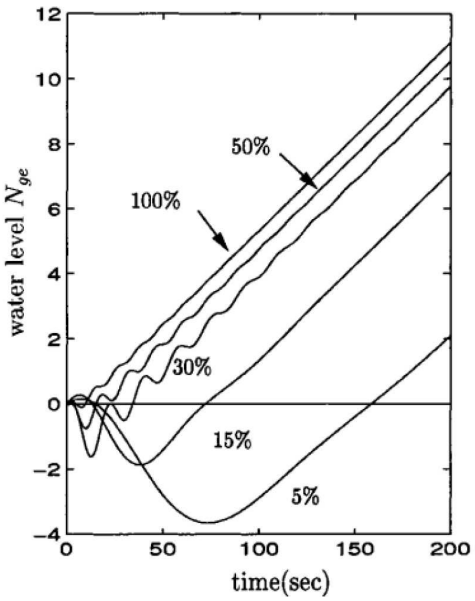


Figure 1(c). Unit load responses to an increased pulse step of control valve open rate [5].

Figure 1(c) shows the behavior of water level drum of a boiler in thermal power plant when feedwater flow increases. It is a self-imbalance process with inverse response in an initial period.

The typical step response of self-balance process with overshoot is presented in Figure 2(a), while the curve of a self-imbalance process with inverse response is drawn in Figure 2(b).

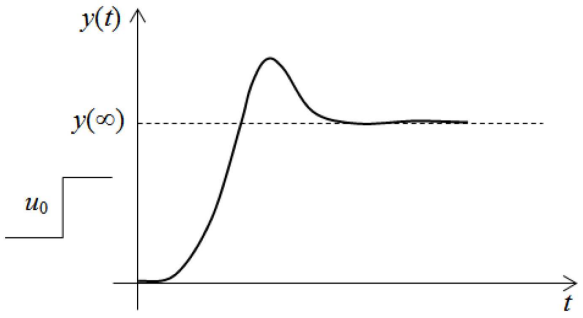


Figure 2(a). Typical step response of self-balance process with overshoot.

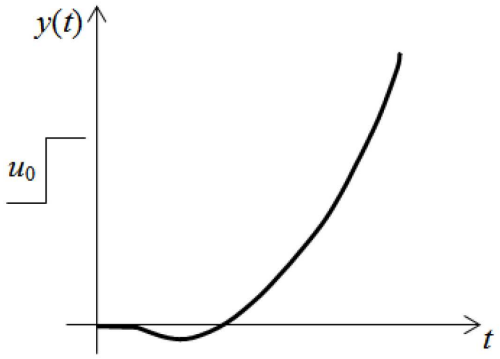


Figure 2(b). Typical step response of self-imbalance process with inverse response.

The reference [1] proposed the numerical method for process identification based on dynamic step responses. The proposal has shown the efficiency and got over the drawback of the manual method. In [1], the self-balance process with overshoot has been identified by the second order plus dead time with a negative zero (SOPDTZ), while the self-imbalance process with inverse response was modeled by integrating plus first order with dead time and a zero (IFOPDTZ). Moreover, the self-imbalance with inverse response was shown to efficiently identify by integrating plus first order with dead time (IFOPDT) for the convenience of controller synthesization [2].

The identified models of the two processes are used to synthesize the PID controller based on the robust control viewpoint presented in [2] and [9].

This study is in continuation of the works [1] and [2] which proposed the methodology including open-loop identification and PID controller tuning. The paper is aimed at:

- to study the control errors of systems of the two special processes to design the PID controllers basing on robust viewpoint;
- to apply the identification method to self-balance process with overshoot.

2. Process Identification

2.1. Identification of the self-balance process with overshoot

In [1], an identification method for self-balance processes including overshoot is presented. The overshoot process is modeled by second order plus dead time with a negative zero (SOPDTZ) by transfer function:

$$O_{SOPDTZ}(s) = \frac{K(1+bs)}{(1+T_1 s)(1+T_2 s)} e^{-\tau s}, \tag{1}$$

where K is the gain factor, T_1, T_2, b are lag constants, τ is the dead time, and s is a complex variable.

The parameters of $O_{SOPDTZ}(s)$ must satisfy the conditions:

$$K > 0, \quad \tau \geq 0, \quad T_1, T_2, b \geq 0.$$

These are defined by using "cleft-over-step" algorithm to solve the target function of the optimal problem:

$$J_1(X_1) = \sum_{i=1}^{N} [y(t_i, X_1) - y_i]^2 + p_1 \Pi_1(X_1) \rightarrow \min_{X_1} \tag{2}$$

in which X_1 is the parameter vector of $O_{SOPDTZ}(s)$ [5]. If $T_1 \neq T_2$, then

$$y(t_i, X_1) = u_0 K \left[1 + \frac{T_1 - b}{T_2 - T_1} e^{-\frac{t_i - \tau}{T_1}} - \frac{T_2 - b}{T_2 - T_1} e^{-\frac{t_i - \tau}{T_2}} \right] \tag{3}$$

and $X_1 = \{K, T_1, T_2, b, \tau\}$. The pulse is $u(t) = u_0.1(t)$.

If $T_1 = T_2 = T$, then

$$y(t_i, X_1) = u_0 K e^{-\frac{t_i - \tau}{T}} \left[\frac{b}{T^2} + (t_i - \tau)\frac{T - b}{T^3} \right] \tag{4}$$

and $X_1 = \{K, T, b, \tau\}$.

N is the number of measuring points, p and $\Pi(X)$ are penalty coefficient and penalty function, respectively, [1].

2.2. Identification of the self-imbalance inverse response process

Based on [2], the self-imbalance process with inverse response is able to be identified by IFOPDT with transfer function:

$$O_{IFOPDT}(s) = \frac{K}{s(1+Ts)} e^{-\tau s}, \tag{5}$$

where K is the gain factor, T is a lag constant, τ is the dead time, and s is a complex variable.

These values are required to satisfy:

$$K > 0, \quad \tau \geq 0, \quad T \geq 0. \tag{6}$$

The parameters of $O_{IFOPDT}(s)$ are defined by using "cleft-over-step" algorithm to solve the target function:

$$J_2(X_2) = \sum_{i=1}^{M} [y(t_i,\ X_2) - y_i]^2 + p_2\Pi_2(X_2) \to \min_{X_2} \tag{7}$$

in which $X_2 = \{K,\ T,\ \tau\}$ is the parameter vector of $O_{IFOPDT}(s)$, while

$$y(t_i,\ X_2) = u_0K[t_i - \tau - T + Te^{-\frac{t_i-\tau}{T}}] \quad (t_i > \tau),$$
$$y(t) \equiv 0 \quad (\forall t \leq \tau), \tag{8}$$

where M is the number of measuring points, p_2 and $\Pi_2(X_2)$ are penalty coefficient and penalty function, respectively.

3. PID Controller for Processes

3.1. Robust controller [2, 7-9]

Closed-loop control in principle is shown in Figure 3 which includes process $O(s)$, controller $R(s)$, input z, output y and disturbance λ.

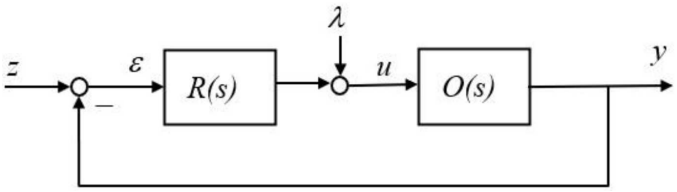


Figure 3. Typical control structure.

$O(s)$ in general type is given by $O(s) = O_{PL}(s)e^{-\tau s}$ ($O_{PL}(s)$ is a rational fraction of variable s, τ is the dead time).

The robust controller is determined based on the model of $O(s)$ as follows:

$$R(s) = \frac{1}{\theta s} O_{PL}(s)^{-1}. \tag{9}$$

in which the lag constant θ is defined as

$$\theta = \frac{\tau e^{m_s\left(\frac{\pi}{2} - \text{arctgm}_s\right)}}{\left(\frac{\pi}{2} - \text{arctgm}_s\right)\sqrt{m_s^2 + 1}}. \tag{10}$$

In equation (10), m_s is robustness index of the system shown in Figure 3. To synthesize the controller $R(s)$, firstly, the robustness index m_s is chosen in the range of $[0.132; 2.318]$ [8]. Secondly, the lag constant θ is calculated by equation (10). Finally, the controller $R(s)$ is conducted by equation (9).

The robustness of the system with $R(s)$ controller is established in [2, 7-9].

3.2. Robust controller for the self-balance process with overshoot

The overshoot process is identified by the model $O_{SOPDTZ}(s)$ (equation (1)). The robust controller for this model is given by

$$R(s) = \frac{1}{\theta s} O_{PL}(s)^{-1}$$

$$= \frac{1}{\theta s}\left(\frac{K(1+bs)}{(1+T_1 s)(1+T_2 s)}\right)^{-1}$$

$$= \frac{1}{\theta s}\left(\frac{(1+T_1 s)(1+T_2 s)}{K(1+bs)}\right)$$

$$= \frac{1}{\theta K}\left(\frac{(1+T_1 s)(1+T_2 s)}{s(1+bs)}\right). \tag{11}$$

The controller $R(s)$ given by (11) is not a PID structure. It is altered as follows:

$$R(s) = \frac{1}{(1+bs)}\left(\frac{(1+T_1 s)(1+T_2 s)}{s\theta K}\right)$$

$$= \frac{1}{(1+bs)}\left(\frac{1+(T_1+T_2)s+T_1 T_2 s^2}{\theta s K}\right)$$

$$= \frac{1}{(1+bs)}\left(\frac{T_1+T_2}{\theta K}\left[\frac{1}{(T_1+T_2)s}+1+\frac{T_1 T_2}{T_1+T_2}s\right]\right)$$

$$= \frac{1}{(1+bs)} K_P\left(\frac{1}{T_I s}+1+T_D s\right), \tag{12}$$

in which $T_I = T_1 + T_2$ is the integral constant, $T_D = \frac{T_1 T_2}{T_I}$ is the derivative constant and $K_P = \frac{T_I}{\theta K}$ is the gain coefficient.

The controller $R(s)$ in equation (12) consists of a PID controller and a first order lag. This first order is designed as a filter and the controller $R(s)$ becomes a PID and a first order filter in series.

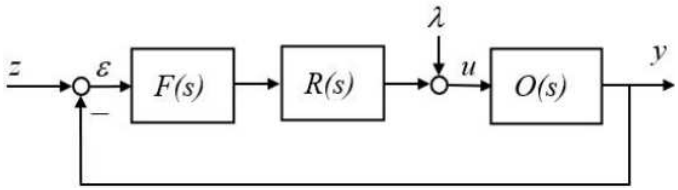


Figure 4. PID controller and filter.

The new control structure is redesigned as shown in Figure 4. In it $F(s)$ is the filter function of a first order lag:

$$F(s) = \frac{1}{(1+bs)}. \tag{13}$$

The filter function $F(s)$ is set unchanged according to the identified function $O_{SOPDTZ}(s)$ in equation (1). Meanwhile, the robust controller $R(s)$ is tuned by tuning the parameters of the PID controller via varying the robustness index m_s of the system.

3.3. Robust controller for the self-imbalance process with inverse response

With identified model IFOPDT in equation (5), the robust controller is as follows:

$$\begin{aligned} R(s) &= \frac{1}{\theta K}(1+Ts) = K_P(1+T_D) \\ &= \frac{1}{\theta K} + \frac{T}{\theta K}s = c_1 + c_2 s, \end{aligned} \tag{14}$$

where

$$K_P = \frac{1}{\theta K}, \quad T_D = T, \quad c_1 = \frac{1}{\theta K}, \quad c_2 = \frac{T}{\theta K}.$$

4. Control Errors of Control System

4.1. Methods

In control structure of Figure 3, suppose $Z(s)$, $\Lambda(s)$, $Y(s)$ and $E(s)$ are Laplace functions of z, λ, y and ε, respectively. The output is created by

both set point (z) and disturbance (λ) signals, calculated as below:

$$Y(s) = \frac{R(s)O(s)}{1 + R(s)O(s)} Z(s) + \frac{O(s)}{1 + R(s)O(s)} \Lambda(s). \tag{15}$$

The error is given by

$$\begin{aligned} E(s) &= Z(s) - Y(s) \\ &= Z(s) - \frac{R(s)O(s)}{1 + R(s)O(s)} Z(s) - \frac{O(s)}{1 + R(s)O(s)} \Lambda(s) \\ &= \frac{1}{1 + R(s)O(s)} Z(s) - \frac{O(s)}{1 + R(s)O(s)} \Lambda(s) \\ &= W_{\varepsilon z}(s) Z(s) + W_{\varepsilon\lambda} \Lambda(s), \end{aligned}$$

where

$$\begin{aligned} W_{\varepsilon z}(s) &= \frac{1}{1 + R(s)O(s)}, \\ W_{\varepsilon\lambda}(s) &= -\frac{O(s)}{1 + R(s)O(s)}. \end{aligned} \tag{16}$$

Set

$$E_z(s) = W_{\varepsilon z}(s) Z(s), \quad E_\lambda(s) = W_{\varepsilon\lambda} \Lambda(s). \tag{17}$$

From this, it follows that

$$E(s) = E_z(s) + E_\lambda(s), \tag{18}$$

where $E_z(s)$ and $E_\lambda(s)$ are errors created by $Z(s)$ and $\Lambda(s)$, respectively, $W_{\varepsilon z}(s)$ is the transfer function from set point z to ε (z channel: $z \to \varepsilon$); $W_{\varepsilon\lambda}(s)$ is the transfer function from disturbance λ to ε (λ channel: $\lambda \to \varepsilon$).

Analyzing $W_{\varepsilon z}(s)$ and $W_{\varepsilon\lambda}(s)$ by Taylor chain in $s = 0$, we have

$$\begin{aligned} W_{\varepsilon z}(s) &= c_0 + c_1 s + c_2 s^2 + \cdots, \\ W_{\varepsilon\lambda}(s) &= d_0 + d_1 s + d_2 s^2 + \cdots \end{aligned} \tag{19}$$

with

$$c_0 = W_{\varepsilon z}(0); \quad c_1 = \frac{dW_{\varepsilon z}(s)}{ds}\bigg|_{s=0}, \ldots$$

and

$$d_0 = W_{\varepsilon\lambda}(0); \quad d_1 = \frac{dW_{\varepsilon\lambda}(s)}{ds}\bigg|_{s=0}, \ldots.$$

Put equation (14) and equation (15) into equation (12) of $E_z(s)$ and $E_\lambda(s)$, and then take the inverse Laplace under the condition that $z(t)$ and $\lambda(t)$ are continuously differentiable in the range $t \in (0; \infty)$:

$$\begin{aligned}\varepsilon_z(t) &= c_0 z(t) + c_1 z'(t) + c_2 z''(t) + \cdots,\\ \varepsilon_\lambda(t) &= d_0\lambda(t) + d_1\lambda'(t) + d_2\lambda''(t) + \cdots.\end{aligned} \tag{20}$$

The functions $\varepsilon_z(t)$ and $\varepsilon_\lambda(t)$ of time variable t are errors of control processes of z and λ channels, respectively. The values of c_0 and d_0 are coefficients of static errors, c_1 and d_1 are coefficients of velocity errors, c_2 and d_2 are coefficients of acceleration errors, … Based on the dynamic characteristics of the system, some of coefficients are equal to zero.

Consider the system in Figure 3 with the controller $R(s)$ of PID type including control signal $z(t)$ and disturbance $\lambda(t)$. Suppose that the inputs are step pulses meaning $z(t) = z_0.1(t)$, $\lambda(t) = \lambda_0.1(t)$, where z_0 and λ_0 are constants.

Based on equations (14) and (15), it is concluded that the error of the control channel z in steady state is as below:

$$\begin{aligned}E_z(\infty) &= \lim_{t\to\infty} E_z(t) = \lim_{s\to 0}[sE_z(s)]\\ &= \lim_{s\to 0}[sW_{\varepsilon z}(s)Z(s)] = \lim_{s\to 0}\left[sW_{\varepsilon z}(s)\frac{Z_0}{s}\right]\\ &= z_0 \lim_{s\to 0}[W_{\varepsilon z}(s)].\end{aligned}$$

Therefore,

$$E_z(\infty) = z_0 \lim_{s\to 0} W_{\varepsilon z}(s) \tag{21}$$

and the error of disturbance channel λ in steady state is given by

$$\begin{aligned} E_\lambda(\infty) &= \lim_{t\to\infty} E_\lambda(t) = \lim_{s\to 0} [sE_\lambda(s)] \\ &= \lim_{s\to 0} [sW_{\varepsilon\lambda}(s)\Lambda(s)] = \lim_{s\to 0}\left[sW_{\varepsilon\lambda}(s)\frac{\lambda_0}{s}\right] \\ &= \lambda_0 \lim_{s\to 0} [W_{\varepsilon\lambda}(s)]. \end{aligned}$$

Therefore,

$$E_\lambda(\infty) = \lambda_0 \lim_{s\to 0} W_{\varepsilon\lambda}(s). \tag{22}$$

4.2. Control errors of the overshoot and inverse systems

This sub-section calculates the control errors in steady state of the overshoot system of SOPDTZ model with robust controller in equation (12) and inverse system of IFOPDT model with robust controller equation (14).

4.2.1. Errors of overshoot system of SOPDTZ and PID controller

From equations (21) and (22), the error of z channel is derived as

$$\begin{aligned} E_z(\infty) &= z_0 \lim_{s\to 0} W_{\varepsilon z}(s) = z_0 \lim_{s\to 0} \frac{1}{1 + R(s)O(s)} \\ &= z_0 \frac{1}{1 + \lim\limits_{s\to 0} R(s)O(s)} \\ &= z_0 \frac{1}{1 + \lim\limits_{s\to 0} R(s) \cdot \lim\limits_{s\to 0} O(s)}, \end{aligned} \tag{23}$$

in which

$$\lim_{s\to 0} R(s) = \lim_{s\to 0} \frac{1}{(1+bs)} K_P\left(\frac{1}{T_I s} + 1 + T_D s\right)$$

$$= \lim_{s\to 0} \frac{1}{(1+bs)} \cdot \lim_{s\to 0} K_P\left(\frac{1}{T_I s} + 1 + T_D s\right)$$

$$= \lim_{s\to 0} \frac{1}{(1+bs)} \cdot K_P \cdot \lim_{s\to 0}\left(\frac{1}{T_I s} + 1 + T_D s\right)$$

$$= 1 \cdot K_P \cdot (\infty) = \infty \tag{24}$$

and

$$\lim_{s\to 0} O(s) = \lim_{s\to 0} O_{\text{SOPDTZ}}(s) = \lim_{s\to 0} \frac{K(1+bs)}{(1+T_1 s)(1+T_2 s)} e^{-\tau s}$$

$$= \frac{\lim\limits_{s\to 0} K(1+bs)}{\lim\limits_{s\to 0} (1+T_1 s)(1+T_2 s)} \cdot \lim_{s\to 0} e^{-\tau s}$$

$$= \frac{K}{1} \cdot 1 = K. \tag{25}$$

Replacing equations (24) and (25) into equation (23), we have

$$E_z(\infty) = z_0 \cdot \frac{1}{1+(\infty)K} = 0. \tag{26}$$

The error of λ channel is as follows:

$$E_\lambda(\infty) = \lambda_0 \lim_{s\to 0} W_{\varepsilon\lambda}(s) = \lambda_0 \lim_{s\to 0} \frac{O(s)}{1+R(s)O(s)}$$

$$= \lambda_0 \frac{\lim\limits_{s\to 0} O(s)}{1+\lim\limits_{s\to 0} R(s) \cdot \lim\limits_{s\to 0} O(s)} = \lambda_0 \frac{K}{1+(\infty)K} = 0. \tag{27}$$

Finally, the error of control system which is created from equation (18) is given by

$$E(s) = E_z(s) + E_\lambda(s) = 0 + 0 = 0. \tag{28}$$

Equations (26)-(28) show that the overshoot control system consisting of SOPDTZ model and PID robust controller (equation (12)) give the errors in

control channel z and disturbance channel λ by zero in steady state leading to the error of control system annulled. In other words, this control structure always guarantees $y(\infty) = z_0$.

4.2.2. Errors of inverse system of IFOPDT and PD controller

From equations (21) and (22), the error of λ channel is derived as follows:

$$E_z(\infty) = z_0 \lim_{s\to 0} W_{\varepsilon z}(s) = z_0 \lim_{s\to 0} \frac{1}{1 + R(s)O(s)}$$

$$= z_0 \frac{1}{1 + \lim\limits_{s\to 0} R(s) \cdot \lim\limits_{s\to 0} O(s)},$$

where

$$\lim_{s\to 0} R(s) = \lim_{s\to 0} K_P(1 + T_d s)$$

$$= K_P\left(1 + \lim_{s\to 0} T_d s\right) = K_P(1 + T_d 0) = K_P \tag{29}$$

and

$$\lim_{s\to 0} O(s) = \lim_{s\to 0} O_{\text{IFOPDT}}(s) = \lim_{s\to 0} \frac{K}{s(1 + Ts)} e^{-\tau s}$$

$$= \lim_{s\to 0} \frac{K}{s(1 + Ts)} \cdot \lim_{s\to 0} e^{-\tau s}$$

$$= \frac{K}{0} \cdot 1 = \infty. \tag{30}$$

From equations (29) and (30), we obtain

$$E_z(\infty) = \frac{z_0}{1 + K_P \cdot (\infty)} = 0. \tag{31}$$

The error of λ channel is as follows:

$$E_\lambda(\infty) = \lambda_0 \lim_{s\to 0} W_{\varepsilon\lambda}(s) = \lambda_0 \lim_{s\to 0}\left(-\frac{O(s)}{1 + R(s)O(s)}\right)$$

$$= \lambda_0 \lim_{s\to 0}\left(-\frac{1}{\frac{1}{O(s)} + R(s)} \right)$$

$$= \lambda_0 \left(-\frac{1}{\frac{1}{\lim\limits_{s\to 0} O(s)} + \lim\limits_{s\to 0} R(s)} \right).$$

From equations (29) and (30), we have

$$E_\lambda(\infty) = -\frac{\lambda_0}{\frac{1}{\infty} + K_P} = -\frac{\lambda_0}{K_P}. \tag{32}$$

Finally, the error of control system as given by equation (18) is

$$E(s) = E_z(s) + E_\lambda(s) = 0 + \left(-\frac{\lambda_0}{K_P} \right) = -\frac{\lambda_0}{K_P}. \tag{33}$$

Equations (31)-(33) show that the integrating inverse control system consisting of IFOPDT model and PD robust controller (equation (12)) gives the error as zero of z control channel in steady state. However, it provides the error of the λ disturbance channel leading to the error of control system not annulled. In other words, this control structure does not make $y(\infty)$ equal to z_0 in the final. This means that the structure needs to be modified for rejection of the error of disturbance channel. The requirement is fulfilled by adding the anti-disturbance element to the robust controller PD.

5. Anti-disturbance Supplement for Robust PD Controller

Based on the analysis in Section 4, the PD controller needs to complete the integral element (I). Besides, the proportional (P) is also to be supplemented to higher anti-disturbance of the system. It means that ΔR_{PI} is to be added to original controller PD $R(s)$. New PID robust controller now

becomes:

$$R^*(s) = R(s) + \Delta R_{PI} = \frac{c_0^*}{s} + c_1^* + c_2 s, \tag{34}$$

in which

$$c_1^* = c_1(1+B), \quad c_0^* = c_1^*(1+B)A,$$

where parameters c_1, c_2 of the PD controller given by

$$c_1 = \frac{1}{\theta K}, \quad c_2 = \frac{T}{\theta K}.$$

are coefficients, and A, B are defined as

$$B = \frac{0{,}2\omega_c c_2}{c_1 \tau}, \quad A = \frac{0{,}35\omega_c c_1}{2c_1\tau + c_2}$$

with

$$\omega_c = \frac{1}{\tau}\left(\frac{\pi}{2} - \arctan m_s\right).$$

6. Example

6.1. Steps of implementation

In this part, a self-balance process with overshoot and a self-imbalance with inverse response of complicated functions are considered for illustration. The methodology is simulated including identification technique and robust PID controller tuning process.

The procedure is implemented via three steps as below:

- 1st step_Process identification (model deduction):

The open-loop unit step responses of the processes are drawn to extract parameters for building the target functions. We solve the optimal functions (equation (2) and (7)) to define parameters of models (equations (1) and (5)). Finally, the unit step and frequency responses of original and identified models are drawn for comparison.

• 2nd step_Robust controller tuning:

By using the identified models and chosen robustness indexes, the robust controller for each process is calculated. After that, the qualitative control indexes including steady time, overshoot and decay are withdrawn from the closed-loop step response.

• 3rd step_PID controller tuning:

Analyzing qualitative indexes, perhaps some factors need to be improved. This will be fulfilled by resetting the m_s robustness index of the system leading to tune the parameters of PID controllers.

6.2. Self-balance process with overshoot

6.2.1. Process identification

Examine the overshoot process with complex transfer function as below:

$$O_1(s) = \frac{(1+25s)(1+4.5s)(1+5.5s)}{(1+4s)(1+5s)(1+7s)(1+8s)(1+10s)} e^{-3s}. \quad (35)$$

Its unit step response is drawn in Figure 5.

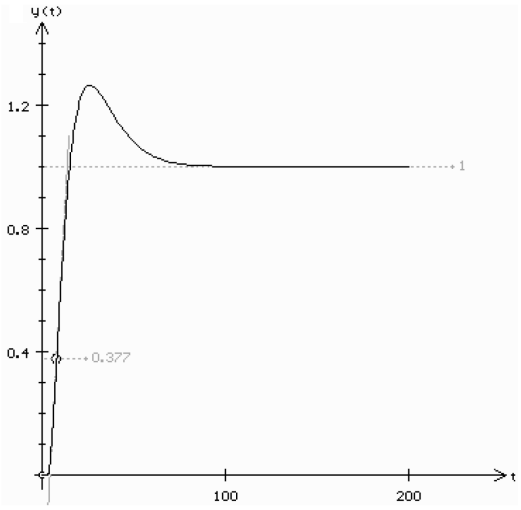


Figure 5. Unit step response of $O_1(s)$.

The identification process is based on Section 2 whose detail is available in [1].

Sub step 1. Target function

The curve has inflection point $U(t_f, y_f)$ with $y_f = 0.377$. Since $g = y_f/(y(\infty)u_0) = 0.377/1 = 0.377 > 0.264$, in model (1), choose $T_1 = T_2$. The unconstraint optimization function equation (2) uses the value of $y(t_i, X_1)$ in equation (4).

Take 30 measuring points of the curve in Figure 5 shown in Table 1.

Table 1. Parameters extracted from the curve

t_1	t_2	t_3	t_4	t_5	t_6
0	2.5	5	10	15	20
y_1	y_2	y_3	y_4	y_5	y_6
0	0	0.087	0.587	0.995	1.201
t_7	t_8	t_9	t_{10}	t_{11}	t_{12}
25	30	35	40	45	50
y_7	y_8	y_9	y_{10}	y_{11}	y_{12}
1.262	1.249	1.206	1.157	1.115	1.080
t_{13}	t_{14}	t_{15}	t_{16}	t_{17}	t_{18}
55	60	65	70	75	80
y_{13}	y_{14}	y_{15}	y_{16}	y_{17}	y_{18}
1.055	1.037	1.025	1.016	1.010	1.007
t_{19}	t_{20}	t_{21}	t_{22}	t_{23}	t_{24}
85	90	95	100	105	110
y_{19}	y_{20}	y_{21}	y_{22}	y_{23}	y_{24}
1.004	1.003	1.002	1.001	1.001	1.001
t_{25}	t_{26}	t_{27}	t_{28}	t_{29}	t_{30}
115	120	125	130	135	140
y_{25}	y_{26}	y_{27}	y_{28}	y_{29}	y_{30}
1.001	1.001	1.001	1.001	1	1

Replace parameters in Table 1 into target function equation (2) to get the optimization function $J(X)$. The roots are the parameters of the model described by equation (1).

Sub step 2. Solving the target function

The optimal function equation (2) is solved by cleft-over algorithm [1]. It needs the start vector of model parameters $X_0 = \{K_0, T_0, b_0, \tau_0\}$. This vector is set based on the unit step response in Figure 5.

The start vector $X_0 = \{K_0, T_0, b_0, \tau_0\}$ is

$$K_0 = 1; \quad T_0 = 9; \quad b_0 = 18; \quad \tau_0 = 3. \tag{36}$$

The gain factor K is set unchangingly, means K is not an optimizing variable.

After 1034 iterative steps of optimal algorithm, the root is achieved:

$$K = 1; \quad T = 12.458; \quad b = 30.475; \quad \tau = 5.291. \tag{37}$$

Replace above values into (1) gaining the identified model as

$$O_1'(s) = \frac{1 + 30.475s}{(1 + 12.458s)^2} e^{-5.291s}. \tag{38}$$

The result means that the step response curve in Figure 5 is approximated by model in equation (38) or the original complex model in equation (35) is deducted to simple model in equation (38).

Sub step 3. Comparison

The quality of identified model $O_1'(s)$ is tested by comparing its time response and frequency response with those of original model $O_1(s)$ (equation (35)).

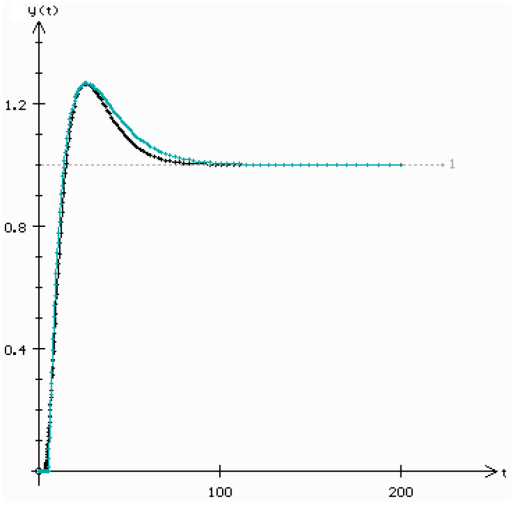


Figure 6. Unit step responses of $O_1(s)$ and $O_1'(s)$: $O_1(s)$*_Black*, $O_1'(s)$*_Blue*.

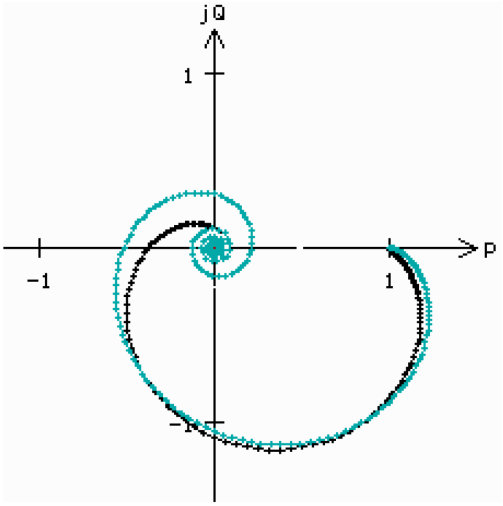


Figure 7. Frequency characteristics of $O_1(s)$ and $O_1'(s)$: $O_1(s)$*_Black*, $O_1'(s)$*_Blue*.

The unit step responses of $O_1(s)$ and $O_1'(s)$ are expressed in Figure 6. The frequency characteristics of $O_1(s)$ and $O_1'(s)$ are shown in Figure 7, in which the curve of $O_1(s)$ is in black and that of $O_1'(s)$ is in blue.

The curves show that approximated model $O_1(s)$ is acceptable to design controller.

6.2.2. Robust controller design

In this step, the robust controller $R(s)$ equation (9) is synthesized for $O_1(s)$ and robust PID controller equation (12) is solved for identified model (38). In both the cases, the robustness index is set the same.

The robustness index m_s of the control system varies in the range from 0.132 to 2.318 [8]. Based on this value, the lag coefficient θ is determined by equation (10).

Sub step 4. Robust controller for $O_1(s)$

Choose $m_s = 0.71$, and replace into equation (10) (with $\tau = 3$) to obtain the lag coefficient $\theta = 5.049$. From this value, the controller $R_1(s)$ by (9) becomes

$$R_1(s) = 0.198\frac{(1+4s)(1+5s)(1+7s)(1+8s)(1+10s)}{s(1+25s)(1+4.5s)(1+5.5s)}. \tag{39}$$

Sub step 5. Robust controller for $O_1'(s)$

Choose $m_s = 0.71$, and replace into (10) (with $\tau = 5.291$) to obtain the lag coefficient $\theta = 8.905$. From this value, the controller $R_1'(s)$ by (12) is given as follows:

$$R_1'(s) = \frac{2.791}{1+30.475s}\left(\frac{1}{24.916s} + 1 + 6.229s\right). \tag{40}$$

In (40), the PID controller is given by

$$2.791\left(\frac{1}{24.916s} + 1 + 6.229s\right) \tag{41}$$

and the filter function by

$$F(s) = \frac{1}{1+30.475s}. \tag{42}$$

Sub step 6. Closed-loop control qualities

To test the control qualities of $R_1'(s)$ controller equation (14), both controller equations (13) and (14) are used for closed control loop of $O_1(s)$ process.

The unit step responses of $O_1(s)$ closed-loop for each $R_1(s)$ and $R_1'(s)$ controllers are shown in Figure 8.

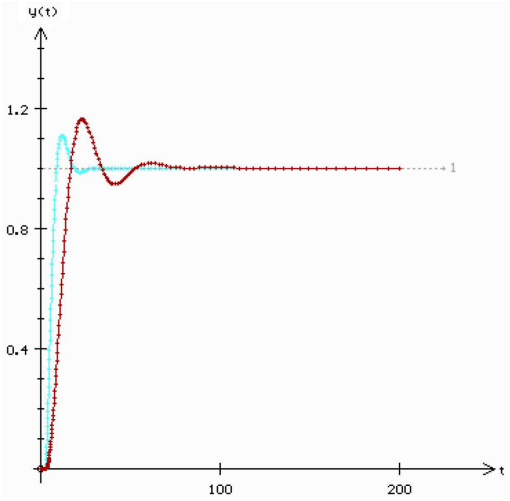


Figure 8. Closed-loop step responses of $R_1(s)$ and $R_1'(s)$: $R_1(s)\_Blue$, $R_1'(s)\_Red$.

Three qualitative factors of $R_1'(s)$ are as follows:

+ Steady time (time from active moment to process value reaching $(100 \pm 5)\%$ of set point: $T_q = 41.67$.

+ Overshoot:

$$\delta = \frac{y_{\max} - y(\infty)}{y(\infty)} = 16.4\%.$$

+ Decay (ratio of second and first peaks): $D = 90\%$.

In comparison, the $R_1'(s)$ controller gives better qualitative factors but the factor expressed by $R_1'(s)$ is fairly good. It suggests that $R_1'(s)$ is acceptable for initial setting of the PID controller for process.

Sub step 7. Retuning controller for better qualities

For detail requirements of each process, it may be needed to improve some factors. This is fulfilled by modifying the robustness index m_s of the system. For example, to reduce the overshoot δ factor, higher m_s from 0.71 to 0.972, the θ lag coefficient now is 10.317. It gives the new controller $R_1''(s)$ as follows:

$$R_1''(s) = \frac{2.417}{1 + 30.475s}\left(\frac{1}{24.916s} + 1 + 6.229s\right). \tag{43}$$

The PID controller with smaller gain becomes

$$2.417\left(\frac{1}{24.916s} + 1 + 6.229s\right), \tag{44}$$

while the filter still remains unchanged in equation (42).

Now, the control qualities of the three controllers are compared.

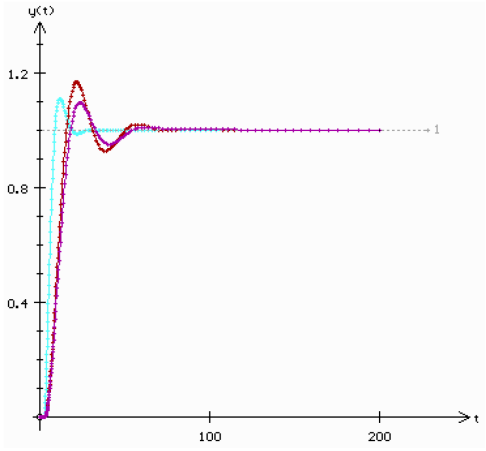


Figure 9. Closed-loop step responses of $R_1(s)$, $R_1'(s)$, $R_1''(s)$: $R_1(s)$ *_Blue*, $R_1'(s)$ *_Red*, $R_1''(s)$ *_Violet*.

Figure 9 shows $O_1(s)$ closed-loop response to unit step input of $R_1(s)$, $R_1'(s)$ and $R_1''(s)$.

The result shows that $R_1''(s)$ gives lower δ about 10% in comparison with those of $R'(s)$. In practice, parameters of controller are able to tune for a better control qualitative factor according to the real requirement.

6.3. Self-imbalance process with inverse

After the detailed presentation above of the self-balance process with overshoot, a brief procedure is expressed for self-imbalance process with inverse.

6.3.1. Process identification

The integrating process with inverse response is presented in equation (45). This model is similar to the one studied by Kaya in [10] but added the inverse response. In [2], the model of Pai et al. [6] was also studied giving good result:

$$O_2(s) = \frac{(1-1.5s)}{s(1+0.25s)(1+0.5s)(1+s)(1+10s)} e^{-3.5s}. \quad (45)$$

The step response of $O_2(s)$ is shown in Figure 10.

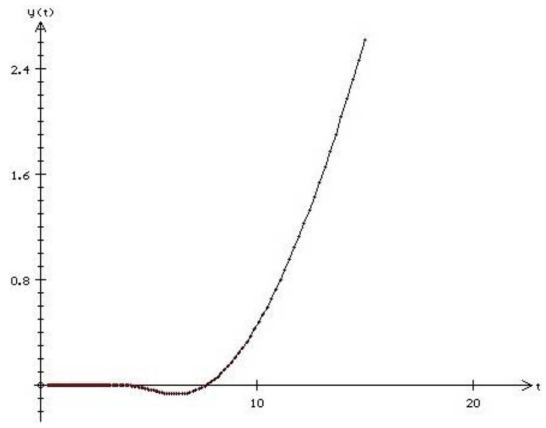


Figure 10. Unit step pulse response of $O_2(s)$.

This process is identified by IFOPDT model by using data from the response curve in Figure 10. The identified model is as follows:

$$O_2'(s) = \frac{0.759}{s(1+6.202s)} e^{-7.093s}. \tag{46}$$

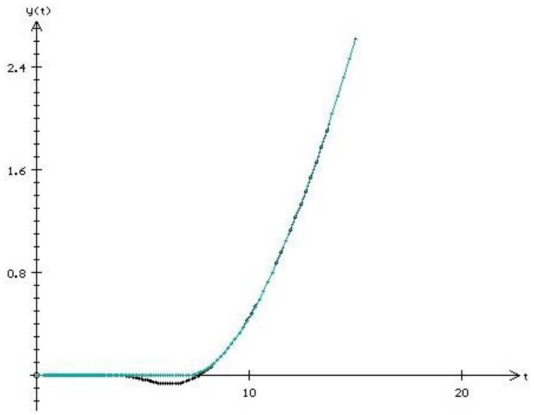


Figure 11. Time responses of $O_2(s)$ and $O_2'(s)$.

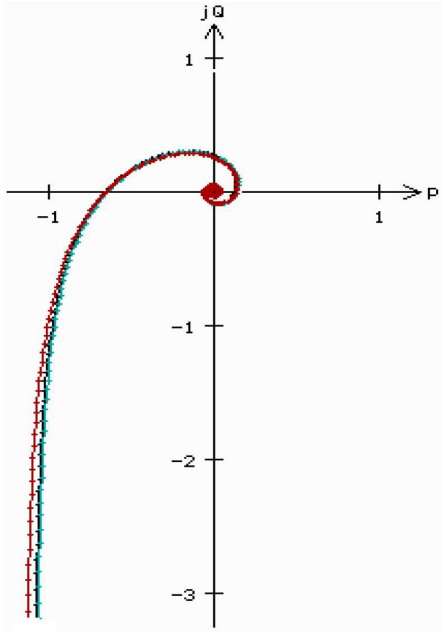


Figure 12. Frequency responses of $O_2(s)$ and $O_2'(s)$.

The time and frequency responses of $O_2(s)$ and $O_2'(s)$ to unit step pulse are shown in Figures 11 and 12. It can be seen that the identification gets high quality.

6.3.2. Original robust controller

The PD robust controller (14) with $m_s = 0.71$ $(\theta = 2.838)$ for $O_2'(s)$ is given by

$$R_2(s) = 0.069\,(1 + 5.445s). \tag{47}$$

The system of robust controller $R_2(s)$ in equation (47) and process $O_2(s)$ (equation (45)) give closed-loop responses presented in Figure 13 including z (control) and λ (disturbance) channels.

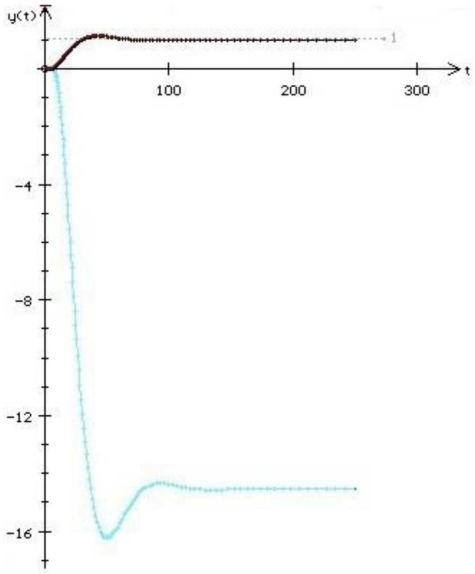


Figure 13. Closed-loop responses of PD controller $R_2(s)$ in z and λ channels.

In z channel, the qualities are:

+ Steady time (time from active moment to process value reaching $(100 \pm 5)\%$ of set point: $T_q = 60.433$.

+ Overshoot:

$$\delta = \frac{y_{\max} - y(\infty)}{y(\infty)} = 13.61\%.$$

+ Decay (ratio of second and first peaks): $D = 100\%$.

The result shows that the error in disturbance channel is not nulled complying with the demonstration above.

6.3.3. Anti-disturbance robust controller

The PID robust controller (34) with $m_s = 0.71$ $(\theta = 2.838)$ for $O_2'(s)$ is as follows:

$$R_2'(s) = 0.098\left(1 + \frac{1}{42.532s} + 5.445s\right). \quad (48)$$

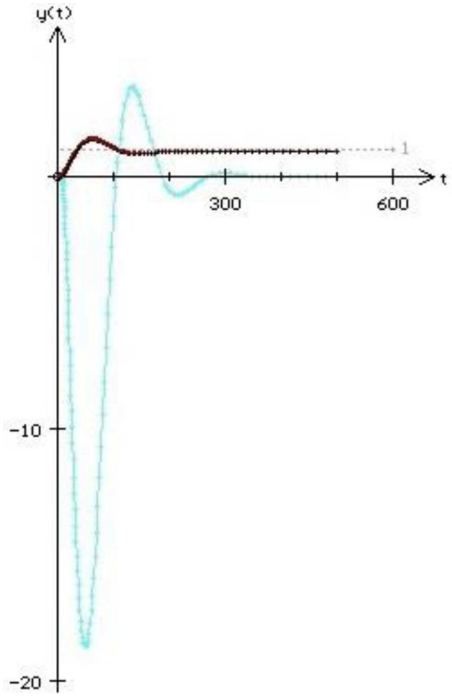


Figure 14. Responses of $R_2(s)$ in z and λ channels: z *channel_black*, λ *channel_blue*.

The system of robust controller $R_2(s)$ in equation (48) and process $O_2(s)$ (equation (45)) give closed-loop responses presented in Figure 14 including z (control) and λ (disturbance) channels.

In z channel, the qualities are:

+ Steady time (time from active moment to process value reaching $(100 \pm 5)\%$ of set point: $T_q = 167.25$.

+ Overshoot:

$$\delta = \frac{y_{\max} - y(\infty)}{y(\infty)} = 48.89\%.$$

+ Decay (ratio of second and first peaks): $D = 100\%$.

In the λ (disturbance) channel, the error is nulled due to ΔR_{PI} (in equation (34)) element added to $R_2(s)$. However, the element also makes the steady time in z (control) channel of the system to be longer and the overshoot value to be higher.

7. Conclusions

The paper is in continuation of the work on process identification and PID tuning based on robust viewpoint. For identification, it is based on numerical technique of cleft-over optimal algorithm presented in [1] which applied the method for self-balance process without overshoot. In this study, the method continuously applied for overshoot process with identified model of second order plus dead time with a negative zero (SOPDTZ), and self-imbalance process with inverse response by integrating first order plus dead time (IFOPDT). The results show that the technique is fairly good for the identification of overshoot and inverse processes.

For tuning work, the robust-based controller synthesized from SOPDTZ model needs to be modified to get PID controller. This is carried out by

analyzing the controller into a PID controller and a first order lag filter connected in series. Meanwhile, the PD controller from IFOPDT model is supplemented for the P and I elements for anti-disturbance.

The closed-loop control quality of the PID controllers in both the cases is acceptable and the qualitative control factors are able to improve with the modification of the robustness index of the system.

The methodology of identification and tuning works for overshoot process is comprehensive, reliable and promising for application.

References

[1] D. C. Trung, A method for process identification and model reduction to design PID controller for thermal power plant, 11th Asian Control Conference (ASCC), December 17-20, 2017, pp. 2714-2719.

[2] D. C. Trung, An identification and tuning method for integrating processes with deadtime and inverse response, GMSARN International Journal 14 (2020), 202-211.

[3] Ji-Zhen Liu, Shu Yan, De-Liang Zeng, Yong Hu and You Lv, A dynamic model used for controller design of a coal fired once-through boiler-turbine unit, Energy 93 (2015), 2069-2078.

[4] L. Sun, D. Li, K. Y. Lee and Y. Xue, Control-oriented modeling and analysis of direct energy balance in coal-fired boiler-turbine unit, Control Engineering Practice 55 (2016), 38-55.

[5] Mayuresh V. Kothare, Bernard Mettler, Manfred Morari, Pascale Bendotti and Clément-Marc Falinower, Level control in the steam generator of a nuclear power plant, IEEE Trans. on Control Systems Technology 8(1) (2000), 55-69.

[6] Neng-Sheng Pai, Shih-Chi Chang and Chi-Tsung Huang, Tuning PI/PID controllers for integrating processes with deadtime and inverse response by simple calculations, Journal of Process Control 20(6) (2010), 726-733.

[7] N. V. Manh, Optimizing methods of uncertainty control systems, Diss. Doc. Sc., MPEI, Moscow, 1999.

[8] D. C. Trung, Research tuning method of thermal process control loops for power plant wide-range operation, Doctoral dissertation, Hanoi University of Science and Technology (HUST), 2019.

[9] D. C. Trung and N. V. Manh, A tuning method for uncertain processes of thermal power plant based on the worst soft characteristic, 11th Asian Control Conference (ASCC), December 17-20, 2017, pp. 594-599.

[10] Ibrahim Kaya, I-PD controller design for integrating time delay processes based on optimum analytical formulas, Preprints of the 3rd IFAC Conference on Advances in Proportional Integral-Derivative Control, Ghent, Belgium, May 9-11, 2018, pp. 575-580.