Hindawi
Complexity
Volume 2020, Article ID 4214102, 26 pages
https://doi.org/10.1155/2020/4214102

Research Article

Fractional-Order Predictive Functional Control of Industrial Processes with Partial Actuator Failures

Min-Ying Li,[1,2] Kang-Di Lu ,[3] Yu-Xing Dai,[2,4] and Guo-Qiang Zeng [2,5]

[1]*Guangdong Zhicheng Champion Group Co., Ltd., Dongguan 523718, China*
[2]*National-Local Joint Engineering Laboratory of Digitalize Electrical Design Technology, Wenzhou University, Wenzhou 325035, China*
[3]*Institute of Cyber Systems and Control, Zhejiang University, Hangzhou 310018, China*
[4]*College of Electrical and Information Engineering, Hunan University, Changsha 410082, China*
[5]*College of Cyber Security, Jinan University, Guangzhou 510632, China*

Correspondence should be addressed to Guo-Qiang Zeng; zeng.guoqiang5@gmail.com

Received 17 December 2019; Revised 13 January 2020; Accepted 21 January 2020; Published 9 February 2020

Academic Editor: Xianggui Guo

As the actuator faults in an industrial process cause damage or performance deterioration, the design issue of an optimal controller against these failures is of great importance. In this paper, a fractional-order predictive functional control method based on population extremal optimization is proposed to maintain the control performance against partial actuator failures. The proposed control strategy consists of two key ideas. The first one is the application of fractional-order calculus into the cost function of predictive functional control. Since the knowledge of analytical parameters including the prediction horizon, fractional-order parameter, and smoothing factor in fractional-order predictive functional control is not known, population extremal optimization is employed as the second key technique to search for these parameters. The effectiveness of the proposed controller is examined on two industrial processes, e.g., injection modeling batch process and process flow of coke furnace under constant faults, time-varying faults, and nonrepetitive unknown disturbance. The comprehensive simulation results demonstrate the performance of the proposed control method by comparing with a recently developed predictive functional control, genetic algorithm, and particle swarm optimization-based versions in terms of four performance indices.

1. Introduction

In industrial processes, actuators play an important role in the industrial control system because an actuator links the controller output to the physical actions and determines the quality of products [1, 2]. However, the actuator faces difficulties in executing the controller demand completely due to the physical malfunction, e.g., friction and saturation. Specially, in the control system of industrial process, an actuator fault often exists, and the control performance may be degraded caused by discrepancies between the desired actuator position and the actual position.

Generally, the actuator faults in industrial processes can be separated into three categories, i.e., the actuator outage, the actuator stuck, and the partial actuator failure [3]. As for the first two categories, it is impossible to improve any controllers' performance because the control process under these two categories is totally uncontrollable. Thus, this study focuses on the third category, i.e., the partial actuator failure, which can be tackled to a certain extent by using the adequate control scheme. As a result, many related research works have been carried out. For example, Wang et al. [4] put forward an iterative learning control (ILC) scheme for batch processes under partial actuator faults according to a particular 2D Fornasini–Marchsini model. Giridhar and El-Farra [5] proposed a robust detection, isolation, and compensation of control actuator faults based on the framework of feedback robust control. Jin et al. [6] proposed an improved ILC scheme to control the nonlinear constrained system with actuator failures. Ding et al. [7] proposed a novel ILC scheme to control the uncertain multiple-input multiple-output discrete system under actuator faults. In [8], a model predictive control method was suggested for the injection molding batch process with partial actuator failures. Due to the uncertainties of actuator fault, the control system design is often mismatched [1, 9]. To deal with this challenge problem, ILC [10–12] has been developed as one of the most popular strategies for different industrial processes. However, as discussed in [3], the ILC is largely dependent on the repetitive nature of such processes, whose performance improvement is confined by this unsuitable assumption because many real-world processes are actually time-varying and nonrepetitive and suffering from persistent disturbance. Thus, the design of the ILC is not fit for industrial processes in practice.

In the past decades, model predictive control (MPC) has shown the potential ability of control design for industrial processes [13–15]. As a recently developed MPC, state-space predictive functional control (PFC) [3, 16, 17] provides a novel insight into control design for the industrial processes against partial actuator failures because it not only has theoretical basis for the control design but also has advantages in hardware implementation, computational capability, and control accuracy. For example, in [3], Tao et al. proposed a PFC method based on a linear quadratic structure for industrial process under actuator failures and highlighted that its performance is better than traditional state-space PFC. In [16], an improved version of PFC was applied into the control design for an injection modeling batch process with a partial actuator. However, this improved PFC's weighting factors on the process state and output tracking error are determined through experience. To alleviate this deficiency, a genetic algorithm-(GA-) based PFC was proposed, where GA was used to tune its weighting factors, and six cases of partial actuator failures were used to demonstrate the performance of GA-based PFC [17]. Besides, Hu et al. [18] combined PFC with GA and linear quadratic structure for industrial processes against the partial actuator failures. Although a lot of good results have been obtained by PFC strategies, the framework of PFC design for industrial control processes still needs to be further explored for achieving high-quality control performance.

Additionally, with the deep study on mathematical fractional-order calculus, the applications of fractional-order controller have been attempted by many researchers. In [19, 20], the PFC based on fractional-order calculus was proposed to control the fractional model with model mismatches. Sanatizadeh and Bigdeli [21] designed the fractional-order predictive functional controller for unstable systems with time delay. In [22], the authors have successfully applied the fractional-order PFC into industrial heating furnace, and the experimental results on the temperature process showed the improvement of the fractional-order PFC. In all aforementioned examples, fractional-order methods have shown better performance than the corresponding traditional integer-order methods. In summary, the fractional-order calculus has a potential ability to improve the performance of traditional integer-order methods. This is one of primary motivations to incorporate fractional-order mechanism into PFC and propose a fractional-order PFC called FOPFC for an industrial process against partial actuator faults.

Unfortunately, the fractional-order calculus also involves more tuning parameters than the integer version. In other words, tuning the related parameters in FOPFC has more difficulty than PFC. In FOPFC, there are three key parameters called fractional-order parameter γ, smoothing factor λ, and prediction horizon P, which have important impacts on the performance of control system. More specifically, as discussed in [19, 20], λ plays the role in smoothing reference trajectory. γ and P have influence on the rapidity and stability of the system response. On the other hand, as tuning these parameters lacks analytical knowledge, the selection of these parameters in [19, 20] is generally based on the trial and error method. In order to alleviate this problem, in this paper, a competitive evolutionary algorithm is employed to optimize the related parameters in FOPFC for improving the closed-loop performance. As an efficient evolutionary algorithm, population-based extremal optimization (PEO) [23] is extended from extremal optimization (EO) [24] and has shown great promising ability in a variety of fields, such as numerical optimization problems including single-objective and multiobjective problems [23, 25], PID/FOPID controllers designing problems [26, 27], and weighting optimization of ensemble learning [28]. To be more precise, in [25], an improved multiobjective PEO was presented for solving multiobjective problems. In [26], multivariable PID controllers were designed using real-coded PEO. In [27], multi-non-uniform mutation-based PEO was applied to designing the FOPID controllers. Abovementioned examples have revealed that PEOs have outstanding superiority over other popular evolutionary algorithms including GA and particle swarm optimization (PSO). As discussed in [23], the authors have done extensive experiments on some benchmark single-objective optimization problems. The results in [23] demonstrate the PEO performs better than or at least competitive with many reported popular single-objective evolutionary algorithms. From the design perspective of PEO algorithm, the PEO used in this paper is relatively simper than other evolutionary algorithms including GA and PSO due to its fewer adjustable parameters and only mutation operation. Thus, the PEO is used in this work. It is worthy to be mentioned that compared with the previous studies [17, 18], the proposed control method extends the integer-order PFC to the fractional version and uses PEO algorithm to tune the main parameters in FOPFC, which is superior to the GA used in [17, 18]. In addition, the existing work [20] designed the parameter FOPFC by the trial-and-error method, while this paper uses the PEO algorithm to tune the related parameters in FOPFC and applies FOPFC to solving the industrial processes under partial actuator failures. To the best of our knowledge, there exist no reported related works focusing on PEO for tuning parameters in FOPFC. Therefore, in order to deal with this problem, the PEO-based FOPFC algorithm called PEO-FOPFC is proposed in this paper by adopting PEO to search for the adjustable parameters in FOPFC.

To the best of our knowledge, this work is the first contribution to optimize the analytical parameters including the prediction horizon, fractional-order parameter, and smoothing factor in a FOPFC controller for industrial processes with partial actuator failures by means of PEO. To be more specific, the principal contributions of this paper are summarized as follows:

(1) A fractional-order predictive functional control (FOPFC) strategy is firstly proposed for the industrial process with partial actuator failures.

(2) Encountered the difficulties in tuning-related parameters in FOPFC strategy due to the increasing adjustable parameters and lacking analytical knowledge, and the population extremal optimization is introduced into FOPFC to search for the adjustable parameters such as the prediction horizon, fractional-order parameter, and smoothing factor.

(3) The effectiveness of the PEO-FOPFC strategy is demonstrated on two industrial processes, e.g., injection modeling batch process and process flow of coke furnace under six cases including constant faults, time-varying faults, and nonrepetitive unknown disturbance. Moreover, the simulation results show that the performance of proposed PEO-FOPFC is much better than the recently developed PFC [16].

(4) The performance of fractional-order strategy is illustrated by the comparison of FOPFC with PFC. In addition, the performance of PEO algorithm is purely verified by comparison PEO-FOPFC with FOPFC and two other popular evolutionary algorithms including GA- and PSO-based FOPFC methods on an injection modeling batch process.

The remainder of this paper is given as follows. Section 2 presents preliminaries concerning fractional-order calculus, basics of EO, and problem formulation of the process. Then, the proposed PEO-FOPFC is described in Section 3. The comprehensive experimental results of two industrial processes are discussed in Sections 4 and 5, respectively. Finally, Section 6 concludes the paper and gives future works.

2. Preliminaries

In this section, a brief overview of fractional-order calculus and canonical EO algorithm is given. Then, the description of a single-input single-output industrial process is presented.

2.1. Fractional-Order Calculus. There are three common definitions of fractional-order calculus called Grünwald–Letnikov (GL) definition, Riemann–Liouville (RL) definition, and Caputo definition [29].

The RL form with order α is given as

$$ {}_{\alpha}D_t^{\alpha} f(t) = \frac{d^n}{dt^n} D_t^{-(n-\alpha)} f(t) = \frac{1}{\Gamma(n-\alpha)} \frac{d^n}{dt^n} \int_b^t (t-\tau)^{n-\alpha-1} f(\tau) d\tau, \tag{1} $$

where $f(t)$ means the function, $[b, t]$ means the interval of $f(t)$, $\alpha > 0$ is the fractional-order with $n-1 < \alpha < n$, and Γ represents the gamma function.

The Caputo form with order α is given as

$$ {}_{\alpha}D_t^{\alpha} f(t) = \frac{1}{\Gamma(n-\alpha)} \int_b^t (t-\tau)^{n-\alpha-1} f^{(n)}(\tau) d\tau, \tag{2} $$

where $f(t)$ means the function, $[b, t]$ means the interval of $f(t)$, $\alpha > 0$ is the fractional-order with $n-1 < \alpha < n$, and Γ represents the gamma function. The GL definition can be described as

$$ {}_{\beta}D_t^{\gamma} f(t) = \lim_{h \longrightarrow 0} \frac{1}{h^{\gamma}} \sum_{j=0}^{[(t-\beta)/h]} (-1)^j \begin{pmatrix} \gamma \\ j \end{pmatrix} f(t-jh), \tag{3} $$

where β is the initial time, h is the calculation step, $[x]$ means the integer part of x, γ is the fractional-order parameter, and $\omega_j^{(\gamma)} = (-1)^j \begin{pmatrix} \gamma \\ j \end{pmatrix}$ are the polynomial coefficients and can be obtained as follows:

$$ \begin{aligned} \omega_0^{(\gamma)} &= 1, \\ \omega_j^{(\gamma)} &= \left(1 - \frac{\gamma+1}{j}\right) \omega_{j-1}^{(\gamma)}, \quad j = 1, 2, \ldots. \end{aligned} \tag{4} $$

In addition, h can be substituted by the sample time T_s, when considering the practical process and the characteristics of fractional order.

As suggested in [20], this paper employs GL definition to derive the discrete form of the control system. Then, the discretized model of fractional-order integer operator can be described as follows [20, 30]:

$$ \begin{aligned} {}^{\lambda}I_a^b f(t) = \int_a^b \left[D^{1-\gamma} f(t)\right] &\cong T_s^{\gamma} \sum_{r=0}^{b/T_s} (-1)^{\gamma} \begin{pmatrix} -\gamma \\ r \end{pmatrix} f(b - rT_s) \\ &- T_s^{\gamma} \sum_{r=0}^{a/T_s} (-1)^{\gamma} \begin{pmatrix} -\gamma \\ r \end{pmatrix} f(a - rT_s) = T_s^{\gamma} \mathrm{WF}_I{}^T, \end{aligned} \tag{5} $$

where

$$ \begin{aligned} W &= [\,\omega_b - \omega_a,\ \omega_{b-1} - \omega_{a-1},\ \ldots,\ \omega_{b-a} - \omega_0,\ \omega_{b-a-1},\ \ldots,\ \omega_1,\ \omega_0\,], \\ F_I &= [\,f(0),\ f(T_s),\ \ldots,\ f(a-T_s),\ f(a),\ \ldots,\ f(b-T_s),\ f(b)\,]. \end{aligned} \tag{6} $$

2.2. Canonical Extremal Optimization. The canonical EO [24] manipulates a single configuration for the purpose of finding a satisfying solution. In this method, a local fitness value is given to each component in the initial solution. Once a suitable permutation is obtained on the basic of assigned local fitness, the mutation operator is employed on the worst component to generate the new solution. Subsequently, the undesirable component is replaced with the new one unconditionally. Canonical EO is outlined in Algorithm 1.

```
Randomly initialize configuration $S_I$. Calculate the global fitness of $S_I$ termed as $C(S_I)$
Set $S_{\text{best}} = \text{SI}$ and $\text{C}(S_{\text{best}}) = \text{C(SI)}$, where $S_{\text{best}}$ donates the best solution found so far
Repeat
    Evaluate the local fitness of each component in the current configuration $S_I$
    Obtain a permutation according to the assigned local fitness
    Employ mutation operation on worst component and a new configuration is generated
    Replace the worst component with the new configuration, unconditionally
    Update the $S_{\text{best}}$ and $\text{C}(S_{\text{best}})$
Until some criterion is satisfied;
Return $S_{\text{best}}$ and $\text{C}(S_{\text{best}})$
```

Algorithm 1: Canonical EO.

2.3. Model of Nonlinear Industrial Process. As discussed in [3, 16], the linear deviation model can be used to describe the nonlinear industrial process. For simplicity, a single-input single-output (SISO) process is used in this study. And, the corresponding process can be obtained through linearization as follows:

$$\begin{cases} x(k+1) = A_I x(k) + B_I u(k-d), \\ y(k) = C_I x(k) + \omega(k), \end{cases} \tag{7}$$

where k and d denote the current time and process time delay, respectively, and $x(k)$, $y(k)$, $u(k)$, and $\omega(k)$ represent the process state, output, input, and unknown measurement noise, respectively. A_I, B_I, and C_I are the system matrices with appropriate sizes.

Here, the term $u^F(k)$ is the failed signal from the actuator. Then, the failure model can be derived as follows:

$$u^F(k) = \text{au}(k), \quad \text{where } 0 < \alpha \leq 1. \tag{8}$$

Afterwards, the version of process under actuator failures can be described as

$$\begin{cases} x(k+1) = A_I x(k) + B_I \alpha u(k-d), \\ y(k) = C_I x(k) + \omega(k). \end{cases} \tag{9}$$

3. Proposed Control Strategy

In this section, we firstly introduce the design of fractional-order predictive functional controller (FOPFC) in Section 3.1. Then, we present the control strategy of the proposed PEO-based FOPFC (PEO-FOPFC) in Section 3.2.

3.1. Design of FOPFC. On the basis of equation (7) in Section 2.3, the state vector can be constructed as follows [17]:

$$\Delta x_m(k) = [\, \Delta x(k),\ \Delta u(k-1),\ \Delta u(k-2),\ \ldots,\ \Delta u(k-d) \,]^T, \tag{10}$$

where the Δ denotes the difference operator.

Afterwards, the new state space model can be obtained as

$$\begin{cases} \Delta x_m(k+1) = A_m \Delta x_m(k) + B_m \Delta u(k), \\ \Delta y(k+1) = C_m \Delta x_m(k+1). \end{cases} \tag{11}$$

where

$$A_m = \begin{bmatrix} A_I & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & B_I \\ \mathbf{0} & 0 & 0 & \cdots & 0 & 0 \\ \mathbf{0} & 1 & 0 & \cdots & \vdots & 0 \\ \mathbf{0} & 0 & 1 & 0 & 0 & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0} & 0 & \cdots & 0 & 1 & 0 \end{bmatrix},$$
$$B_m = [\, 0 \ 1 \ 0 \ \cdots \ 0 \,]^T,$$
$$C_m = [\, C_I \ 0 \ 0 \ \cdots \ 0 \,]. \tag{12}$$

Note that the $\mathbf{0}$ is full vector with zero elements with the appropriate dimensions.

The output tracking error is described as follows:

$$e(k) = y(k) - r(k), \tag{13}$$

where $r(k)$ denotes the reference trajectory. And, $r(k+i) = \lambda^i$ $y(k) + (1-\lambda)^i c(k)$, λ denotes the smoothing factor and $c(k)$ is the set point.

Combining equations (11) and (13), the dynamic output error can be derived as follows:

$$e(k+1) = e(k) + C_m A_m \Delta x_m(k) + C_m B_m \Delta u(k) - \Delta r(k+1), \tag{14}$$

where $\Delta r(k+1)$ means the differenced value of set-point at time $k+1$. By adding the tracking error to the state variable, the extended state vector can be obtained as follows:

$$z(k) = [\Delta x_m(k) \ \ e(k)]^T. \tag{15}$$

Then, the new model is obtained as follows:

$$z(k+1) = Az(k) + B\Delta u(k) + C\Delta r(k+1), \tag{16}$$

where $A = \begin{bmatrix} A_m & \mathbf{0} \\ C_m A_m & 1 \end{bmatrix}$; $B = \begin{bmatrix} B_m \\ C_m B_m \end{bmatrix}$; $C = \begin{bmatrix} \mathbf{0} \\ -1 \end{bmatrix}$. $\mathbf{0}$ is the zero vector with appropriate sizes.

As mentioned in [20], the cost function of integer PFC is chosen as follows:

$$J_{\text{PFC}} = \sum_{j=1}^{P} z^T(k+j) Q_j z(k+j), \tag{17}$$

where P is the prediction horizon. And, the diagonal matrix Q_j is often used as the weighting factor to give a specific value for each state variable in $z(k+j)$. As discussed in [31, 32], matrix Q_j plays an important role in control performance of the predictive controller. Therefore, in [31, 32], GA or EO was employed to tune Q_j and showed better performance than the trial and error method. In fact, equation (17) can be viewed as the continuous generalization of$J = \int_a^b [f(t)]^2 \mathrm{d}t$, where $[a, b]$ is the continuous integer interval. Because the fractional order is derived from the integer-order and it has been demonstrated to provide better control performance than integer-order in various domains, a natural idea is to replace integer-order with fractional order and test whether the performance can be improved. Thus, the cost function of FOPFC can be derived as follows [20]:

$$J_{\text{FOPFC}} = \sum_{j=1}^{P} z^T(k+j)\Lambda_j(T_s, \gamma) z(k+j), \tag{18}$$

where $\Lambda_j(T_s, \gamma) = T_s diag(m_{l-1}, m_{l-2}, \ldots, m_1, m_0)$ with $m_j = \omega_i^{(-\gamma)} - \omega_{i-(l-1)}^{(-\gamma)}, \omega_0^{(-\gamma)} = 1, \omega_i^{(-\gamma)} = (1-(1-\gamma)/j)\omega_{i-1}^{(-\gamma)}$ for $i > 0$, and otherwise $\omega_i^{(-\gamma)} = 0$, l is the number of state variables in $z(k)$, and γ is the fractional-order factor.

The PFC control action is based on the base functions [3, 16]:

$$u(k+i) = \sum_{j=1}^{N} \eta_j f_j(i), \tag{19}$$

where η_j is the coefficient, $f_j(i)$ denotes the base function, and N is the number of base functions.

Here, we denote $T_{\mathrm{i}} = [f_1(i), f_2(i), \ldots, f_{\mathrm{N}}(i)]$, $(i = 0, 1, \ldots, P-1)$, and $\Upsilon = [\eta_1, \eta_2, \ldots, \eta_N]^T$. Then, equation (19) can be rewritten as follows:

$$u(k+i) = T_i \Upsilon. \tag{20}$$

From equation (16), together with equation (20), we can obtain

$$\begin{aligned}
z(k+1) &= Az(k) + B\Delta u(k) + C\Delta r(k+1) \\
&= Az(k) + B[u(k) - u(k-1)] + C\Delta r(k+1) \\
&= Az(k) + B[T_0\Upsilon - u(k-1)] + C\Delta r(k+1) \\
&= Az(k) - Bu(k-1) + BT_0\Upsilon + C\Delta r(k+1), \\
z(k+2) &= Az(k+1) + B\Delta u(k+1) + C\Delta r(k+2) \\
&= A[Az(k) - Bu(k-1) + BT_0\Upsilon + C\Delta r(k+1)] + B[u(k+1) - u(k)] + C\Delta r(k+2) \\
&= A^2 z(k) - ABu(k-1) + ABT_0\Upsilon + AC\Delta r(k+1) + BT_1\Upsilon - BT_0\Upsilon + C\Delta r(k+2) \\
&= A^2 z(k) - ABu(k-1) + [(AB-B)T_0 + BT_1]\Upsilon + AC\Delta r(k+1) + C\Delta r(k+2), \\
z(k+3) &= Az(k+2) + B\Delta u(k+2) + C\Delta r(k+3) \\
&= A\left[A^2 z(k) - ABu(k-1) + [(AB-B)T_0 + BT_1]\Upsilon + AC\Delta r(k+1) + C\Delta r(k+2)\right] \\
&\quad + B[u(k+2) - u(k+1)] + C\Delta r(k+3) \\
&= A^3 z(k) - A^2 Bu(k-1) + \left[\left(A^2 B - AB\right)T_0 + ABT_1\right]\Upsilon + A^2 C\Delta r(k+1) + AC\Delta r(k+2) \\
&\quad + BT_2\Upsilon - BT_1\Upsilon + C\Delta r(k+3) \\
&= A^3 z(k) - A^2 Bu(k-1) + \left[\left(A^2 B - AB\right)T_0 + (AB-B)T_1 + BT_2\right]\Upsilon \\
&\quad + A^2 C\Delta r(k+1) + AC\Delta r(k+2) + C\Delta r(k+3), \\
&\ \ \vdots \\
z(k+P) &= Az(k+P-1) + B\Delta u(k+P-1) + C\Delta r(k+P) \\
&= A\begin{bmatrix} A^{P-1} z(k) - A^{P-2} Bu(k-1) + \left[\sum_{i=1}^{P-2}\left(A^i B - A^{i-1}B\right)T_{P-2-i} + BT_{P-2}\right]\Upsilon \\ + A^{P-2} C\Delta r(k+1) + A^{P-3} C\Delta r(k+2) + \cdots + C\Delta r(k+P-1) \end{bmatrix} \\
&\quad + BT_{P-1}\Upsilon - BT_{P-2}\Upsilon + C\Delta r(k+P) \\
&= A^P z(k) - A^{P-1} Bu(k-1) + \left[\sum_{i=1}^{P-1}\left(A^i B - A^{i-1}B\right)T_{P-1-i} + BT_{P-1}\right]\Upsilon \\
&\quad + A^{P-1} C\Delta r(k+1) + A^{P-2} C\Delta r(k+2) + \cdots + AC\Delta r(k+P-1) + C\Delta r(k+P).
\end{aligned} \tag{21}$$

Then, the following equation can be obtained:

$$\begin{bmatrix} z(k+1) \\ z(k+2) \\ z(k+3) \\ \vdots \\ z(k+P) \end{bmatrix} = \begin{bmatrix} A \\ A^2 \\ A^3 \\ \vdots \\ A^P \end{bmatrix} z(k) - \begin{bmatrix} B \\ AB \\ A^2B \\ \vdots \\ A^{P-1}B \end{bmatrix} u(k-1)$$

$$+ \begin{bmatrix} BT_0 \\ (AB-B)T_0 + BT_1 \\ (A^2B - AB)T_0 + (AB-B)T_1 + BT_2 \\ \vdots \\ \sum_{i=1}^{P-1} (A^iB - A^{i-1}B)T_{P-1-i} + BT_{P-1} \end{bmatrix} \Upsilon$$

$$+ \begin{bmatrix} C & 0 & 0 & 0 & 0 \\ AC & C & 0 & 0 & 0 \\ A^2C & AC & C & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A^{P-2}C & A^{P-3}C & A^{P-4}C & \cdots & C \end{bmatrix} \cdot \begin{bmatrix} \Delta r(k+1) \\ \Delta r(k+2) \\ \Delta r(k+3) \\ \vdots \\ \Delta r(k+P) \end{bmatrix}. \tag{22}$$

In addition, denote the $Z = [z(k+1), z(k+2), \ldots, z(k+P)]^T$ and $\Delta R = [\Delta r(k+1), \Delta r(k+2), \ldots, \Delta r(k+P)]^T$.

We obtain

$$Z = Fz(k) - Gu(k-1) + \Phi\Upsilon + S\Delta R, \tag{23}$$

where

$$F = \begin{bmatrix} A \\ A^2 \\ A^3 \\ \vdots \\ A^P \end{bmatrix},$$

$$G = \begin{bmatrix} B \\ AB \\ A^2B \\ \vdots \\ A^{P-1}B \end{bmatrix},$$

$$S = \begin{bmatrix} C & 0 & 0 & 0 & 0 \\ AC & C & 0 & 0 & 0 \\ A^2C & AC & C & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A^{P-1}C & A^{P-2}C & A^{P-3}C & \cdots & C \end{bmatrix},$$

$$\Phi = \begin{bmatrix} BT_0 \\ (AB-B)T_0 + BT_1 \\ (A^2B - AB)T_0 + (AB-B)T_1 + BT_2 \\ \vdots \\ \sum_{k=1}^{P-1} (A^kB - A^{k-1}B)T_{P-1-k} + BT_{P-1} \end{bmatrix}. \tag{24}$$

Then, equation (18) can be rewritten as

$$J_{\mathrm{FOPFC}} = Z^T \Lambda Z, \tag{25}$$

where $\Lambda = \text{block diag}\{\Lambda_1, \Lambda_2, \ldots, \Lambda_P\}$.

Afterwards, the optimal control law can be obtained by finding the minimum value of equation (25):

$$\Upsilon = -\left(\Phi^T \Lambda \Phi\right)^{-1} \Phi^T \Lambda (Fz(k) - Gu(k-1) + S\Delta R). \tag{26}$$

At last, the control signal $u(k)$ is then derived as follows:

$$u(k) = \sum_{j=1}^{N} \eta_j f_j(0). \tag{27}$$

In the realistic industrial process, there exist system uncertainties. Thus, a robust stability condition is needed for the closed-loop control system to ensure a stable system. In [16] and [33], the authors have given the robust stability condition of state space predictive controller. Here, extended from [16], we give a robust stability condition for the proposed FOPFC, which is described below.

Theorem 1. *For the industrial process considering unknown partial actuator failures, i.e., description in equation (9), if the FOPFC is designed based on the model equation (7) such that the following condition holds:*

$$\sigma_{\max}(\Delta A) < -\sigma_{\max}(A - BK_s) + \sqrt{\sigma_{\max}{}^2(A - BK_s) + \frac{\lambda_{\min}(M_P)}{\lambda_{\max}(W_P)}}, \tag{28}$$

where $\sigma_{\max}(\kappa)$ is the maximum singular value ofκ, $\lambda_{\min}(\kappa)$ and $\lambda_{\max}(\kappa)$ are the minimum and maximum eigenvalues ofκ, respectively, and M_P and W_P represent the symmetric positive matrices subject to the following equation:

$$(A - BC_S)^T W_P (A - BC_S) - W_P = -M_P, \tag{29}$$

where

$$\Delta A = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \alpha B_I - B_I & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \vdots & \vdots & \vdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \cdots & \mathbf{0} & \alpha C_I B_I - C_I B_I & \mathbf{0} \end{bmatrix},$$

$$\Omega = \begin{bmatrix} B & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ AB & B & \mathbf{0} & \cdots & \mathbf{0} \\ A^2B & AB & B & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A^{P-1}B & A^{P-2}B & A^{P-3}B & \cdots & B \end{bmatrix},$$
$$C_S = (1,\ \mathbf{0}\ \ldots,\ \mathbf{0})\left(\Omega^T \Phi \Phi^T \Lambda \Omega\right)^{-1} \Phi \Phi^T \Lambda F. \tag{30}$$

Then, the proposed controller is the robust stability for the considered system.

The proof is presented in Proof of Theorem 1 in Appendix.

The error tracking and constant disturbances rejecting the performance of PFC have been given in [34]. And, we can examine error estimates and sensitivity to disturbances of the proposed controller extended from [34] as follows.

Proposition 1. *If the process is treated in the form of equation (16) and the subsequent FOPFC law is designed as equation (20), then the proposed FOPFC control law tracks the constant set-point without steady error and for the constant input disturbances and output disturbances and the FOPFC can reject with no steady error.*

The proof is presented in Proof of Proposition 1 in Appendix.

Remark 1. As suggested in [1, 3, 16], to facilitate the controller design, the process model is based on nominal state space model and the noise is not considered. In the simulation part, the noise is not ignored. To consider the $\omega(k)$ in design controller, one can use system identification technique [10].

3.2. PEO-Based FOPFC Control Strategy.

These are two key strategies in the PEO-FOPFC method. One is that the fractional -order mechanism is applied into the cost function of PFC technique, which is presented in Section 3.1. The other is that the PEO algorithm is used to search the adjustable parameters in FOPFC. In the evolutionary algorithm, the fitness function plays an important role in searching the optimal parameters. Thus, we firstly define the fitness function used in the process of evolution and then describe the specific steps of the PEO-FOPFC.

3.2.1. Fitness Function.

In [17], the combination of overshoot and rise time was used as the fitness function to tune the weighting factors in PFC, while the integral of time weighted absolute error (*ITAE*) was adopted as fitness definition in [31]. As discussed in [26], a more reasonable performance index has been proposed, which considers not only integral of absolute error (*IAE*) but also overshoot, steady-state error, rise time, settling time, square of the input signal, and output signal. Also, its superiority to *IAE* and *ITAE* has been demonstrated on multivariable PID controllers. Thus, in this paper, we use the following equation (31) as fitness function for the industrial process under partial actuator faults. As seen from equation (31), this fitness function not only considers *IAE*, rise time, and settling time but also the square of input signal to avoid exporting a large control value and Δy to avoid getting a large overshoot:

$$F(S) = \begin{cases} w_1\left(t_r + t_s\right) + \int_0^\infty \left(w_2|e(t)| + w_3 u^2(t)\right)\mathrm{d}t, & \text{if } \Delta y(t) \geq 0, \\ w_1\left(t_r + t_s\right) + \int_0^\infty \left(w_2|e(t)| + w_3 u^2(t) + w_4|\Delta y(t)|\right)\mathrm{d}t, & \text{otherwise,} \end{cases} \tag{31}$$

where t_r, t_s, $e(t)$, $u(t)$, and $y(t)$ are rise time, settling time, system error, input signal, and output signal at the time t. w_1, w_2, and w_3 are weight coefficients. As suggested in [26], here w_1, w_2, w_3, and w_4 are set as $w_1 = 50$, $w_2 = 0.999$, $w_3 = 0.001$, and $w_4 = 100$.

Remark 2. w_1, w_2, w_3, and w_4 are the weight coefficients, which have a large influence on the control performance. From equation (31), one can see that w_1 controls the rise time and settling time of system response. The parameters w_2 and w_3 have effect on the system error and input signal, and w_4 has impact on the overshoot of system response. In real-life engineering, these weight coefficients are often determined via the experiential rules and trial-and-error method according to priority of performance indices. In general, w_2 and w_3 are subject to the equation, i.e., $w_2 + w_3 = 1$ and w_2 is often set much larger than w_3. Thus, w_2 and w_3 are set as 0.999 and 0.001, respectively. In addition, w_4 is generally set as $50 \leq w_4 \leq 100$ or a larger value; here, $w_4 = 100$ is used. After determining w_2, w_3, and w_4, the parameter w_1 is determined by the trial-and-error method and set as 50. The weight coefficients, i.e., w_1, w_2, w_3, and w_4 are not the optimal values in this paper. In fact, how to obtain more appropriate weight coefficients is still worth studying.

3.2.2. Main Description of PEO-FOPFC.

The main parameters P, γ, and λ in FOPFC are optimized by the PEO algorithm. The flowchart of PEO-FOPFC strategy is shown in Figure 1. The detailed steps of optimizing three parameters in FOPFC by the PEO algorithm are described as follows.

> Input: the system model, PEO's adjustable parameters including the population size N_P, the maximum number of iterations $I_{\max}$, and mutation parameter b, number of base functions N, and upper and lower values of parameters to be optimized, i.e., P, γ, and λ

Output: the best solution S_{best} (i.e., the optimal parameters P, γ, and λ used in FOPFC) and the corresponding fitness value C_{best}

Step 1. The parameters to be optimized are encoded into a solution S in the PEO algorithm given in Figure 2. More specifically, one initial population $\mathbf{P_I} = \{S_1, S_2, \ldots, S_{\text{NP}}\}$ contains generated solutions with NP size, where each solution $S_i = [P_i, \gamma_i, \lambda_i]$ denotes one group parameters used in FOPFC. Then, set $\mathbf{P} = \mathbf{P_I}$, $S_{best} = \text{PI}_{be}$ (PI_{be} is the best solutions in $\mathbf{P_I}$ based on the fitness value defined in equation (27)), and $C_{\text{best}} = F(S_{best})$.

Step 2 (for each solution S_i in $\mathbf{P}$).

(a) Obtain the D mutated solutions $\{S_{ik,}\ (k = 1, 2, 3, \ldots, D)\}$ by application of multi-non-uniform mutation (MNUM) operation [35] shown in equations (32) and (33). More specially, the j-th component is mutated by MNUM operation and the other components remain unchanged, and then fitness value for each mutated solution is obtained by calculating the fitness function defined in equation (31). For example, for solution $S_1 = [P_1, \gamma_1, \lambda_1]$ in $\mathbf{P}$, three mutated solutions $S_{11} = [P_1' , \gamma_1, \lambda_1]$, $S_{12} = [P_1, \gamma_1' , \lambda_1]$, and $S_{13} = [P_1, \gamma_1, \lambda_1']$ can be obtained by using MNUM operation. And, the process of mutation operation in PEO is shown in Figure 3:

$$S_i(x_j) = \begin{cases} S_i(x_j) + (U_j - S_i(x_j)) \times A(t), & \text{if } r < 0.5, \\ S_i(x_j) + (S_i(x_j) - L_j) \times A(t), & \text{if } r \geq 0.5, \end{cases} \tag{32}$$

$$A(t) = \left[r_1 \left(1 - \frac{t}{I_{\max}} \right) \right]^b, \tag{33}$$

where the subscript j is the j-th decision variable, x is the decision variable, U and L are the upper and lower values of decision variable, t is the current number of iteration, r and r_1 are uniformly distributed random values between 0 and 1, and b is the mutation parameter.

(b) Access each mutated solutions based on the fitness function $F(S_{ik})$. Then, rank the D solutions $\{S_{ik,}\ (k = 1, 2, 3, \ldots, D)\}$.

(c) Select the best solution among S_{ik} according to the rank index and term it as S_{bi}. Additionally, the corresponding best fitness value is termed as C_{bi}. Then save S_{bi} and C_{bi} in $\mathbf{P_b}$ and $\mathbf{C_b}$ for the purpose of updating, respectively.

Step 3. Update the best solution and the corresponding fitness. More specially, find the best solution C_{nb} in $\mathbf{C_b}$ and corresponding solution S_{nb}, if C_{nb} is better than C_{best}, and then set $C_{best} = C_{nb}$ and $S_{best} = S_{nb}$.

Step 4. Accept $\mathbf{P} = \mathbf{P_b}$ unconditionally.

Step 5. Obtain the optimal parameters of FOPFC (i.e., the optimal parameters P, γ, and λ used in FOPFC) and corresponding fitness value C_{best} when the predefined $I_{\max}$ is satisfied; otherwise, go to Step 2 with the $\mathbf{P}$.

Remark 3. To improve integer-order PFC, we have used two key operators in the PEO-FOPFC. One is the application of fractional-order calculus into PFC. This operator makes the PFC have more tuning parameters. The other operator is employing an effective PEO algorithm to optimize the related parameters in FOPFC by minimizing the constructed fitness function, i.e., equation (31). From equation (31), one can see that a comprehensive performance is considered. Additionally, a specific improvement performance can be obtained by adjusting the weighting factors. Compared with the integer-order PFC, the fractional-order PFC is more likely to obtain a better comprehensive performance by using PEO.

Remark 4. The fractional calculus introduced into the cost function of PFC is expected to enhance the performance of PFC because of more tuning parameters. However, how to determine the adjustable parameters of FOPFC is not easy due to lacking analytical knowledge. Thus, this paper proposed PEO to tune the related parameters of FOPFC. In other words, the proposed PEO-FOPFC is one kind of FOPFC in essence, where the adjustable parameters of FOPFC are determined by PEO.

Remark 5. From the description of PEO-FOPFC, we can see that PEO-FOPFC mainly performs the following operations: (1) evaluating population fitness, where the time complexity of FOPFC needs to be considered; (2) sorting population fitness; (3) population mutations, which include $\mathbf{O}(I_{\max}N_PDPT_t/T_s)$, $\mathbf{O}(I_{\max}N_PD\lg N_PD)$, and $\mathbf{O}(I_{\max}N_PD)$. Thus, the time complexity of PEO-FOPFC is $\mathbf{O}(I_{\max}N_PDPT_t/T_s)$ or $\mathbf{O}(I_{\max}N_PD\lg N_PD)$, whichever is larger. In practice, the PEO can be used offline to optimize the adjustable parameters of FOPFC. In other words, the time complexity of the application of PEO-FOPFC into real-world engineering problem is same with FOPFC i.e., $\mathbf{O}(PT_t/T_s)$, where T_t is the time range of simulation.

Remark 6. In [22], the authors have successfully applied the fractional-order predictive functional control into industrial heating furnace. According to this reference, we can implement the proposed PEO-FOPFC for real-world engineering problem described as follows: after using the PEO algorithm, we can obtain the optimized FOPFC. Then, Grünwald–Letnikov approximation is used to transform the fractional-order process into an integer formulation. Finally, the implementation of fractional-order mechanism can be transformed into the implementation of integer model-based PFC.

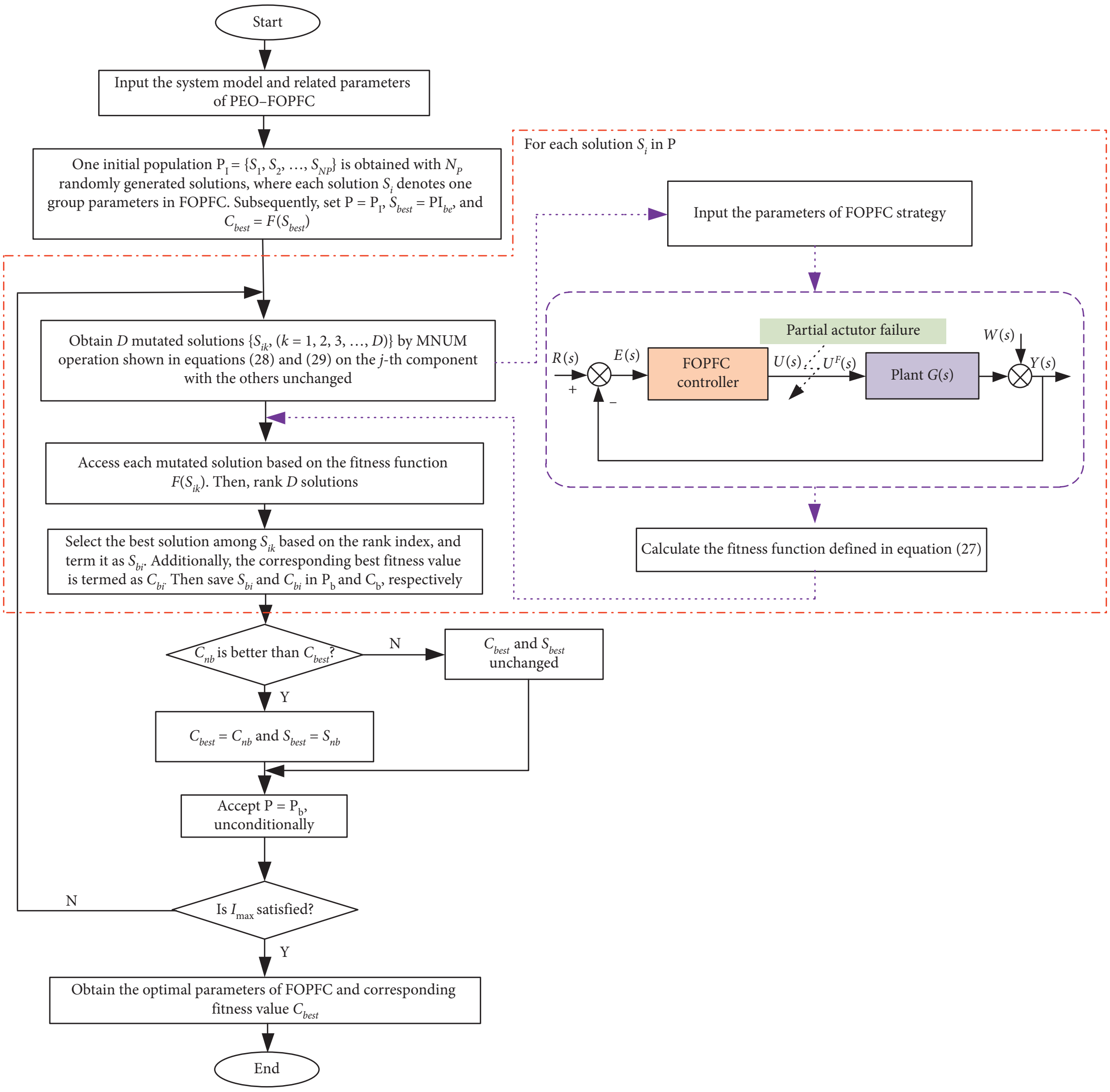

Start
Input the system model and related parameters of PEO–FOPFC
One initial population $P_I = \{S_1, S_2, \ldots, S_{NP}\}$ is obtained with N_p randomly generated solutions, where each solution S_i denotes one group parameters in FOPFC. Subsequently, set $P = P_I$, $S_{best} = PI_{be}$, and $C_{best} = F(S_{best})$
For each solution S_i in P
Obtain D mutated solutions $\{S_{ik}, (k = 1, 2, 3, \ldots, D)\}$ by MNUM operation shown in equations (28) and (29) on the j-th component with the others unchanged
Access each mutated solution based on the fitness function $F(S_{ik})$. Then, rank D solutions
Select the best solution among S_{ik} based on the rank index, and term it as S_{bi}. Additionally, the corresponding best fitness value is termed as C_{bi}. Then save S_{bi} and C_{bi} in P_b and C_b, respectively
Input the parameters of FOPFC strategy
Partial actutor failure
$W(s)$
$R(s)$
$E(s)$
FOPFC controller
$U(s)$
$U^F(s)$
Plant $G(s)$
$Y(s)$
+
–
Calculate the fitness function defined in equation (27)
C_{nb} is better than C_{best}?
N
C_{best} and S_{best} unchanged
Y
$C_{best} = C_{nb}$ and $S_{best} = S_{nb}$
Accept $P = P_b$, unconditionally
N
Is I_{max} satisfied?
Y
Obtain the optimal parameters of FOPFC and corresponding fitness value C_{best}
End

Figure 1: The flowchart of the proposed PEO-FOPFC strategy.

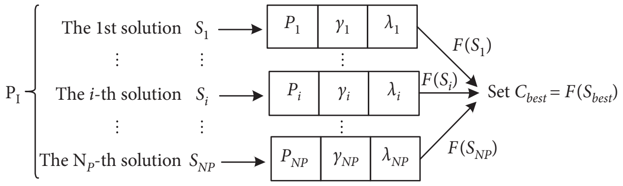


Figure 2: Encoding of related parameters in FOPFC.

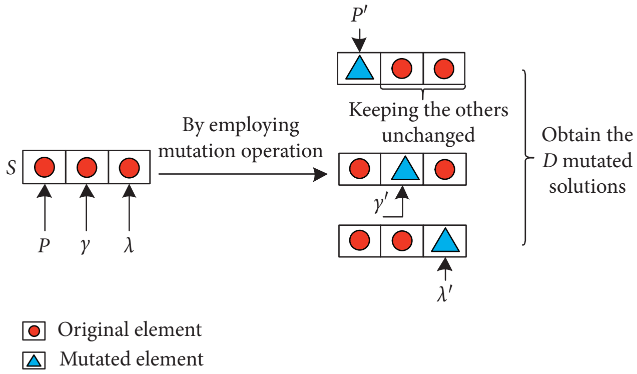

S
P
γ
λ
By employing mutation operation
P'
Keeping the others unchanged
γ'
λ'
Obtain the D mutated solutions
Original element
Mutated element

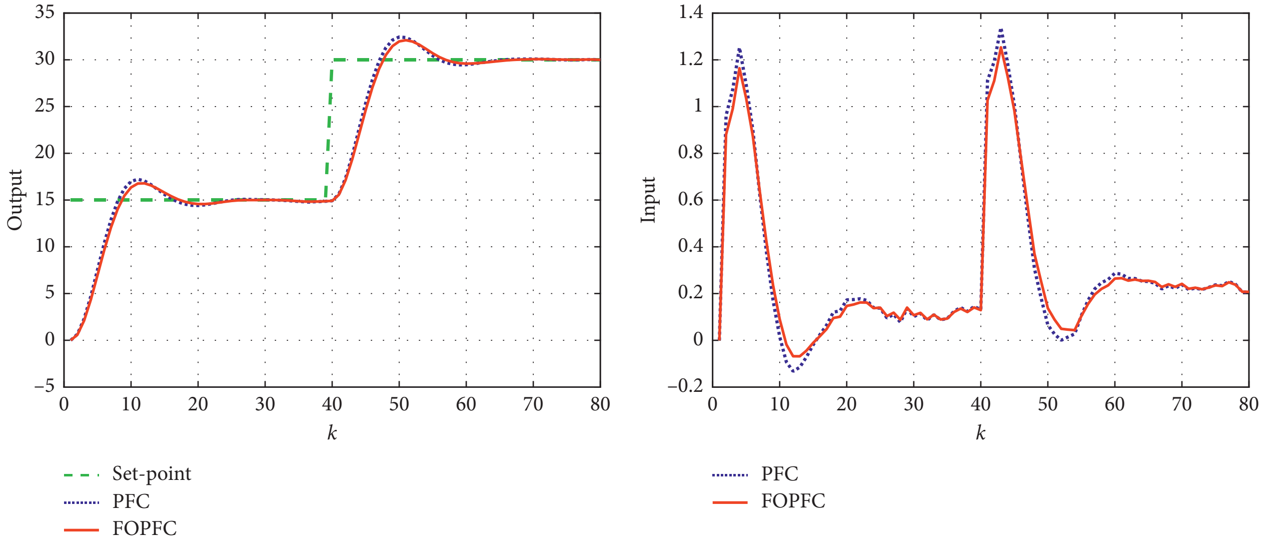
Figure 3: The process of mutation operation in PEO.

4. Simulation Results on Injection Molding Process

In this section, we firstly introduce a typical industrial process, i.e., injection molding process in Section 4.1. Then, in order to demonstrate the effectiveness of the proposed PEO-FOPFC, we do some experiments on control of the injection velocity under different partial faults and unknown disturbance. In this section, our experiments are divided into three parts for different purposes. The first part is to investigate the ability of fractional order. Thus, we compare the FOPFC with PFC in this part. The second one is to compare the proposed control method with the recently published PFC [16]. Note that the PFC in [16] is

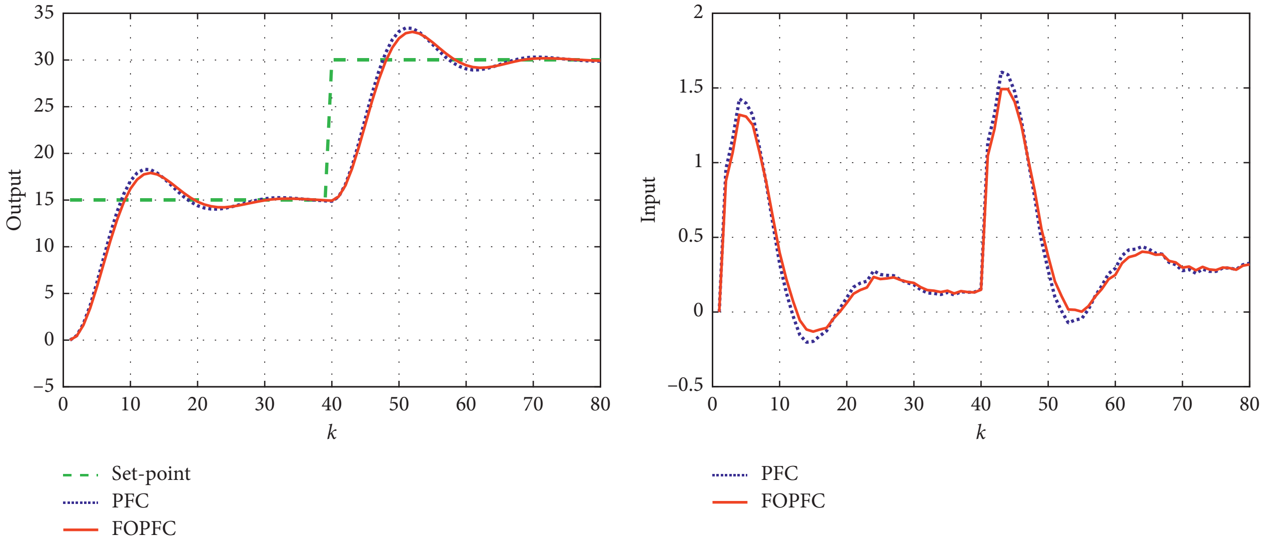
Table 1: The conditions of six actuator faults.

Three constant faults and a random white noise of standard deviation 0.2	Case 1	$\alpha = 0.4$
	Case 2	$\alpha = 0.3$
	Case 3	$\alpha = 0.25$
Three time-varying faults and a random white noise of standard deviation 0.2	Case 4	$\alpha = 0.4 + 0.2\sin(k)$
	Case 5	$\alpha = 0.3 + 0.2\sin(k)$
	Case 6	$\alpha = 0.2 + 0.1\sin(k)$



(a) (b)

Figure 4: The results obtained by FOPFC and PFC under Case 1 in simulation 1. (a) Output responses; (b) input signals.



(a) (b)

Figure 5: The results obtained by FOPFC and PFC under Case 2 in simulation 1. (a) Output responses; (b) input signals.

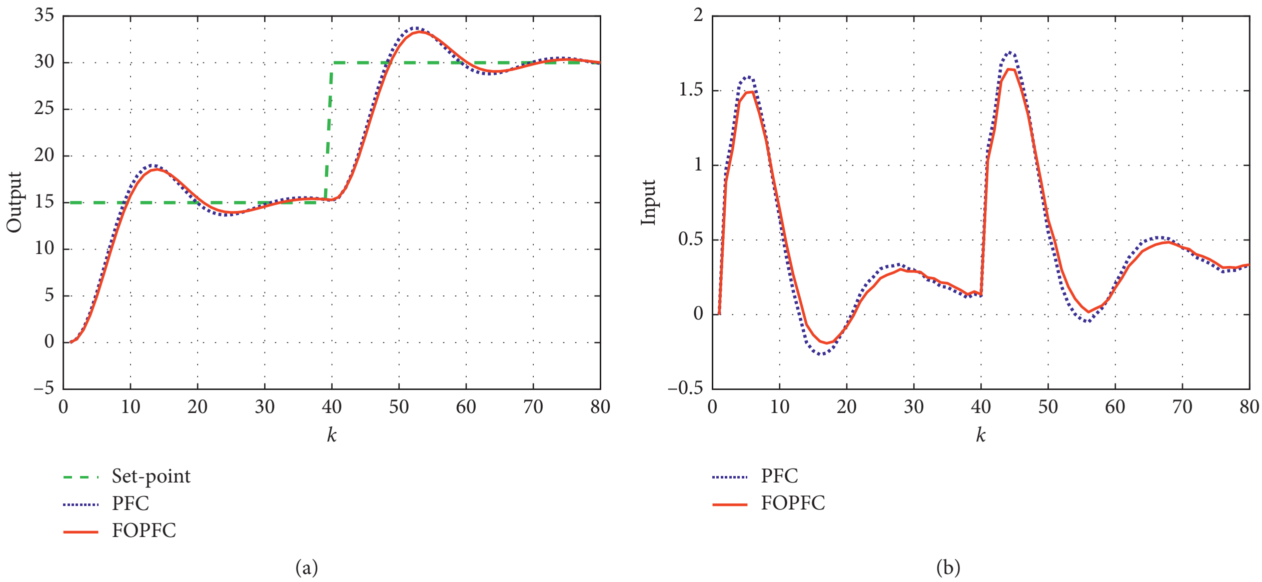


FIGURE 6: The results obtained by FOPFC and PFC under Case 3 in simulation 1. (a) Output responses; (b) input signals.

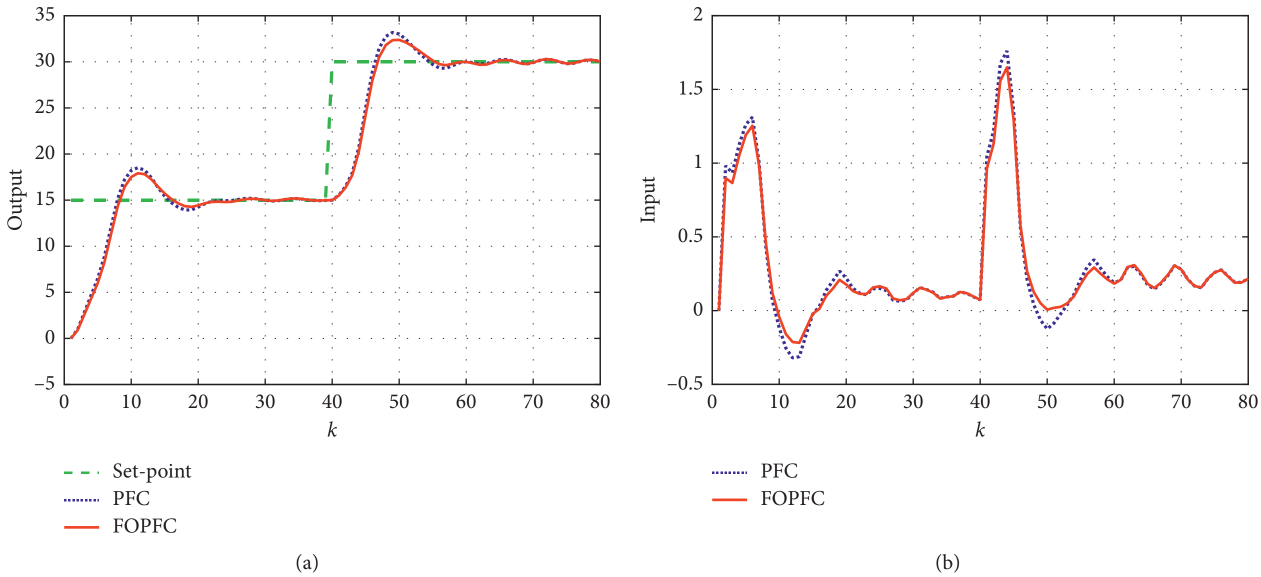
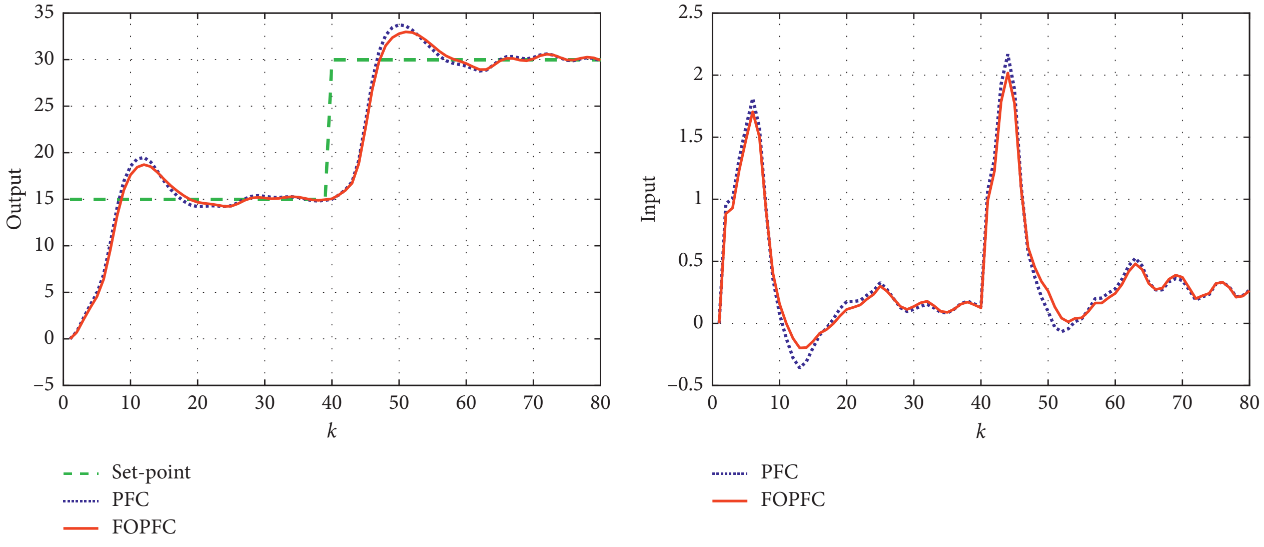


FIGURE 7: The results obtained by FOPFC and PFC under Case 4 in simulation 1. (a) Output responses; (b) input signals.

the improved version of PFC, and the performance of PFC in [16] has been proved to be better than the traditional PFC. This part is designed to show the PEO-FOPFC's superiority to the reported PFC [16]. The third one is to validate the influence of PEO on tuning the related parameters (i.e., P, λ, and γ) in PEO-FOPFC. Here, the FOPFC and two other evolutionary algorithms (i.e., GA and PSO) based FOPFC methods are considered as competitors. It should be noted that all the computer simulations are conducted in MATLAB 2016a software on a 2.5 GHz and 8 GB RAM computer under the Windows 7 operating system.

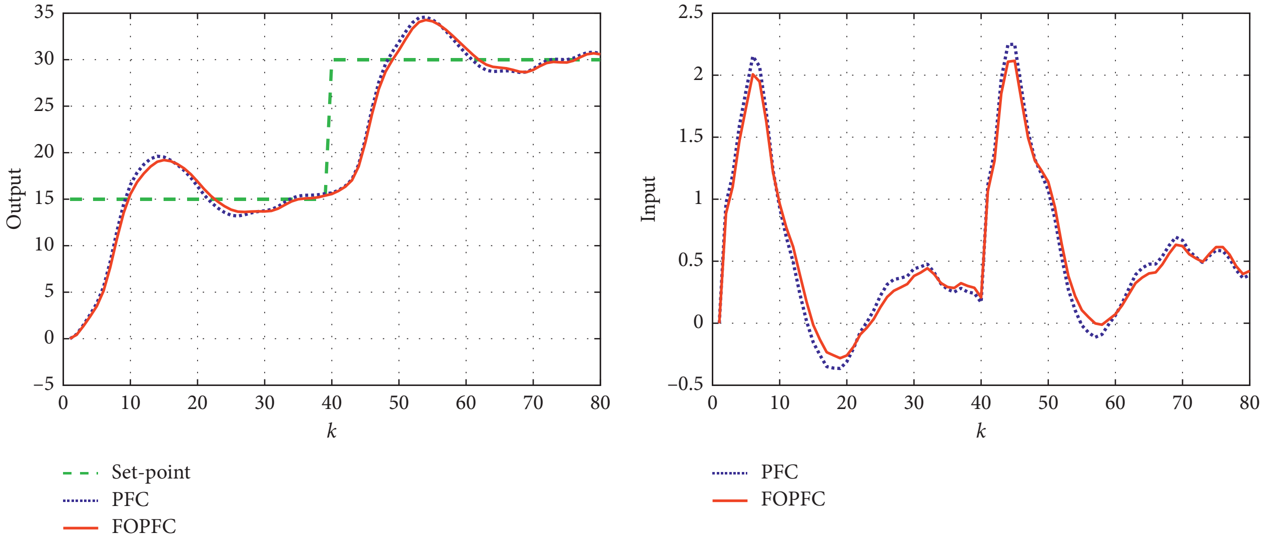
4.1. System Description. As reported in [16], the process of injection molding consists of three phases: filling, packing, and cooling. Among above three phases, the packing stage is very important because it plays key role in maintaining product quality, mechanical strength, deformation, and accuracy. Therefore, the injection velocity should be controlled with high-precision to get the high-quality products. Generally, it is operated through the opening of the proportional valve by regulating the flow of hydraulic oil. Thus, in this process, the output is injection velocity termed as $y(k)$, and the input variable is the proportional valve opening termed as $u(k)$, respectively. In this paper, the typical



(a)

(b)

Figure 8: The results obtained by FOPFC and PFC under Case 5 in simulation 1. (a) Output responses; (b) input signals.



(a)

(b)

Figure 9: The results obtained by FOPFC and PFC under Case 6 in simulation 1. (a) Output responses; (b) input signals.

Table 2: The results of control performance obtained by FOPFC and PFC in simulation 2.

	Case	Algorithm	*IAE*	*ITAE*	M_d	*Ess*
Three constant faults	Case 1	PFC [16]	**165.572**	**51.773**	2.443	0.0511
		FOPFC	167.798	52.217	**2.098**	**0.0252**
	Case 2	PFC [16]	205.998	68.744	3.435	0.1290
		FOPFC	**204.480**	**67.094**	**3.018**	**0.0708**
	Case 3	PFC [16]	232.875	80.469	3.698	0.0706
		FOPFC	**231.065**	**78.484**	**3.312**	**0.0162**
Three time-varying faults	Case 4	PFC [16]	183.022	57.771	3.189	0.0317
		FOPFC	**180.799**	**56.268**	**2.381**	**0.0266**
	Case 5	PFC [16]	218.925	72.425	3.739	0.1219
		FOPFC	**213.976**	**69.688**	**2.981**	**0.0675**
	Case 6	PFC [16]	271.063	96.301	4.576	0.6553
		FOPFC	**267.114**	**94.254**	**4.270**	**0.5764**

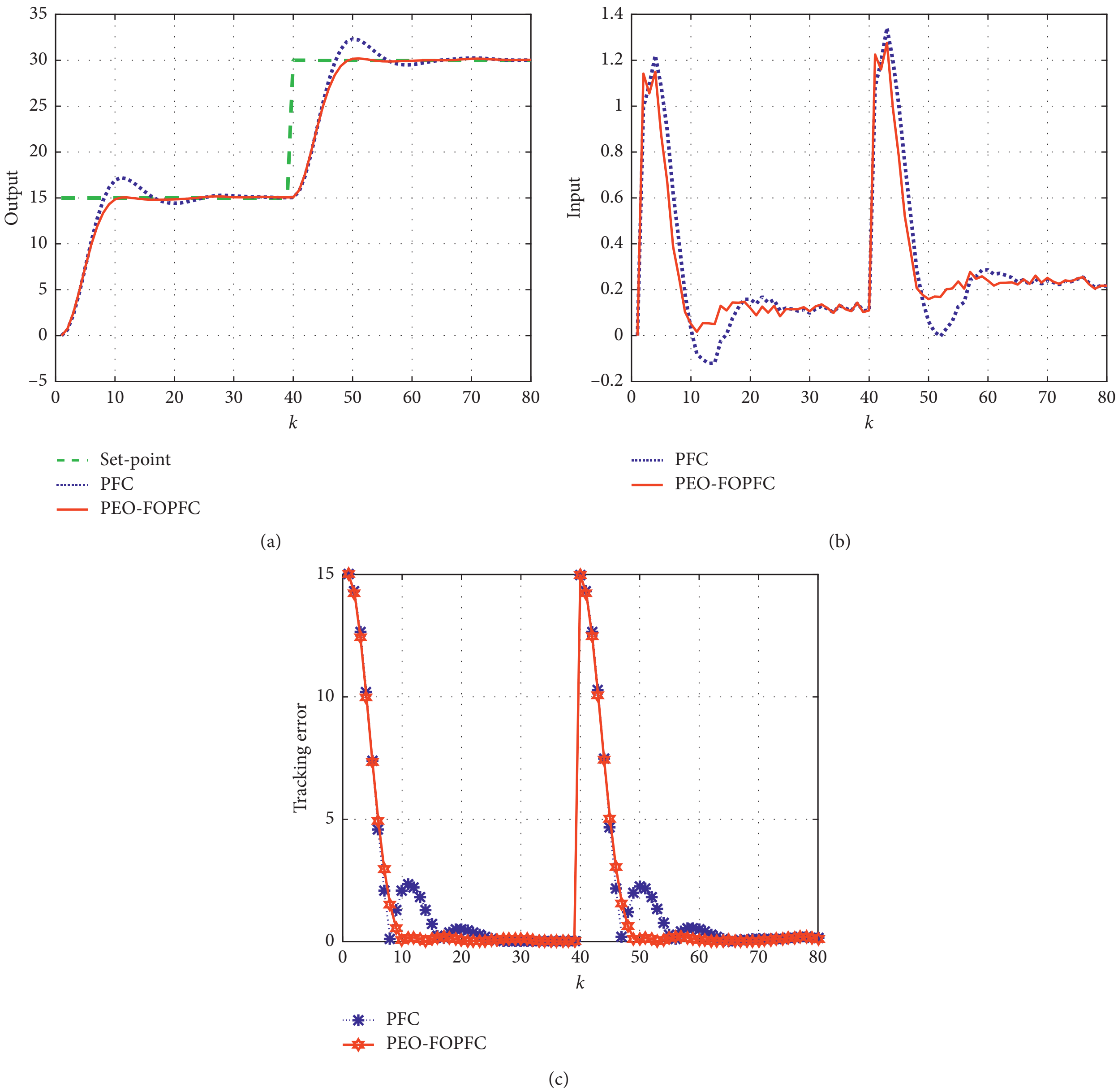


Figure 10: The results obtained by PEO-FOPFC and PFC under Case 1 in simulation 2. (a) Output responses; (b) input signals; (c) tracking error signals.

injection velocity control model is adopted from [16] shown as follows:

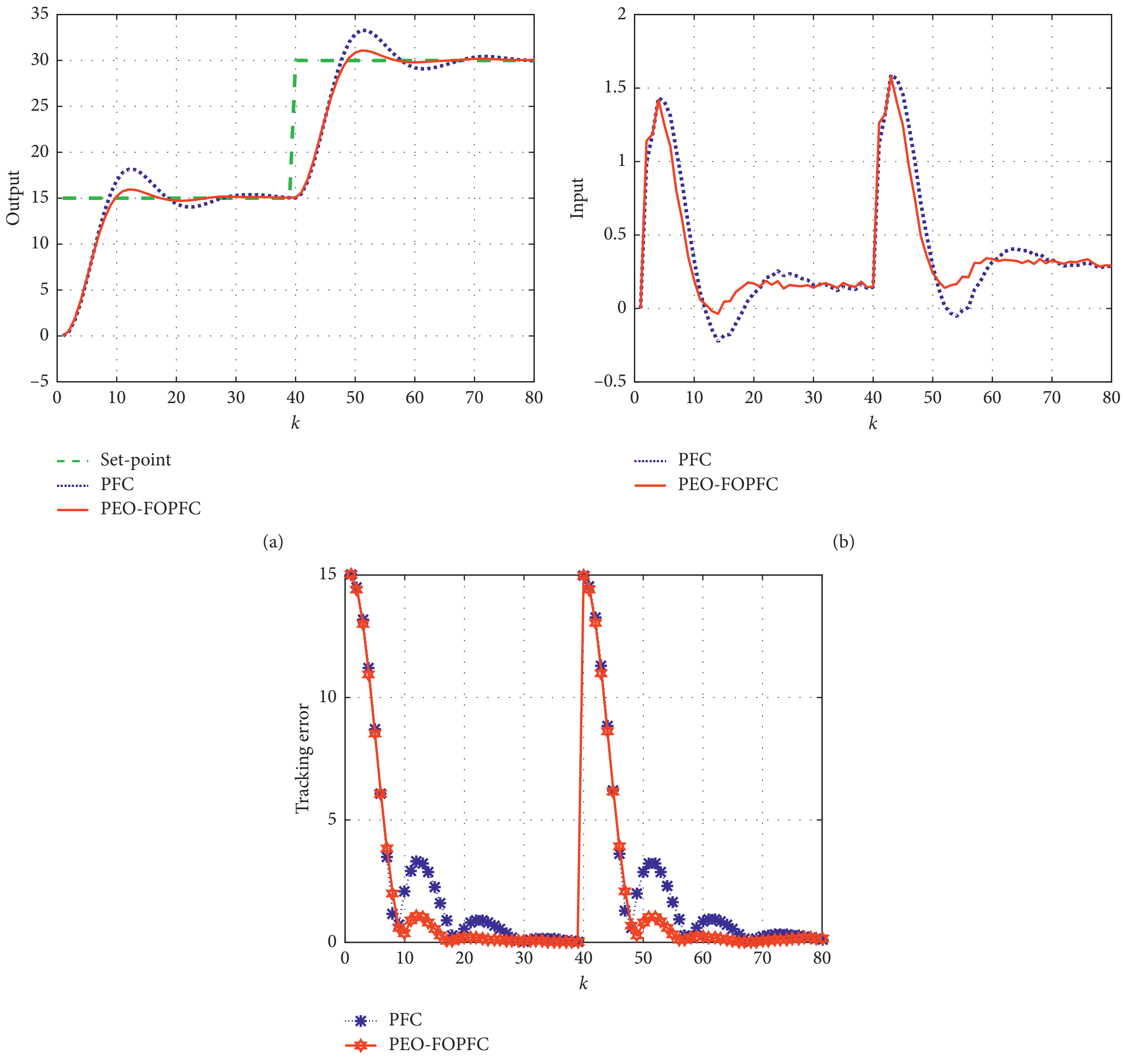
$$P(z) = \frac{y(z)}{u(z)} = \frac{1.69z + 1.419}{z^2 - 1.582z + 0.5916}, \tag{34}$$

where $y(z)$ and $u(z)$ are the z-transforms of $y(k)$ and $u(k)$, respectively. In addition, there exists unknown actuator failure α in the valve opening. And, the set-point $r(k)$ form is shown in the following equation:

$$\begin{cases} r(k) = 15 \,(\text{for } 1 \le k < 40), \\ r(k) = 30 \,(\text{for } 40 \le k \le L), \end{cases} \tag{35}$$

where L is set as 80. The goal of controller design is to make the process output track the set point as closely as possible under actuator failures and unknown disturbances. In this study, six faults are considered under nonrepetitive unknown disturbance. More specially, three cases of constant faults and three cases of time-varying faults with a random white noise of standard deviation 0.2 added to the process output are used. Table 1 gives the above six conditions.

Additionally, in order to quantitative analyze the control performance of different methods, we use the following performance indices: *IAE*, *ITAE*, maximum deviation M_d, and steady-state error *Ess*. Here, *IAE*, *ITAE*, M_d, and *Ess* are



(a)

(b)

(c)

Figure 11: The results obtained by PEO-FOPFC and PFC under Case 2 in simulation 2. (a) Output responses; (b) input signals; (c) tracking error signals.

defined for injection velocity control system in this study as follows:

$$\begin{aligned} \text{IAE} &= \sum_{k=1}^{L} |e(k)|, \\ \text{ITAE} &= \sum_{k=1}^{L} k/L|e(k)|, \qquad (36) \\ M_d &= \max(y(k)) - r(L), \\ Ess &= y(L) - r(L). \end{aligned}$$

Smaller performance implies a better control performance.

Remark 7. In the all simulations, the fault is from beginning to end, which includes the moment the set-point is changing.

4.2. Simulation 1: FOPFC and Its Comparison with PFC. As suggested in [16], the related parameters in PFC are set as follows: prediction horizon $P = 4$, the weighting factors $Q_j = [1, 0, 1, 0]$, and the number of base function $N = 1$. To the best of the authors' knowledge, the FOPFC strategy is firstly proposed for the industrial process with partial actuator failures. Thus, this subsection is devoted to investigating the control performance of FOPFC against the partial actuator failures in the industrial process and the effects of fractional-order mechanism on the control performance. Here, the version of PFC equipped with fractional-order mechanism, i.e., FOPFC, is considered to be compared with PFC [16]. Note that the main parameters in FOPFC are set as the same as those of PFC, except the specific parameter, i.e., fractional-order

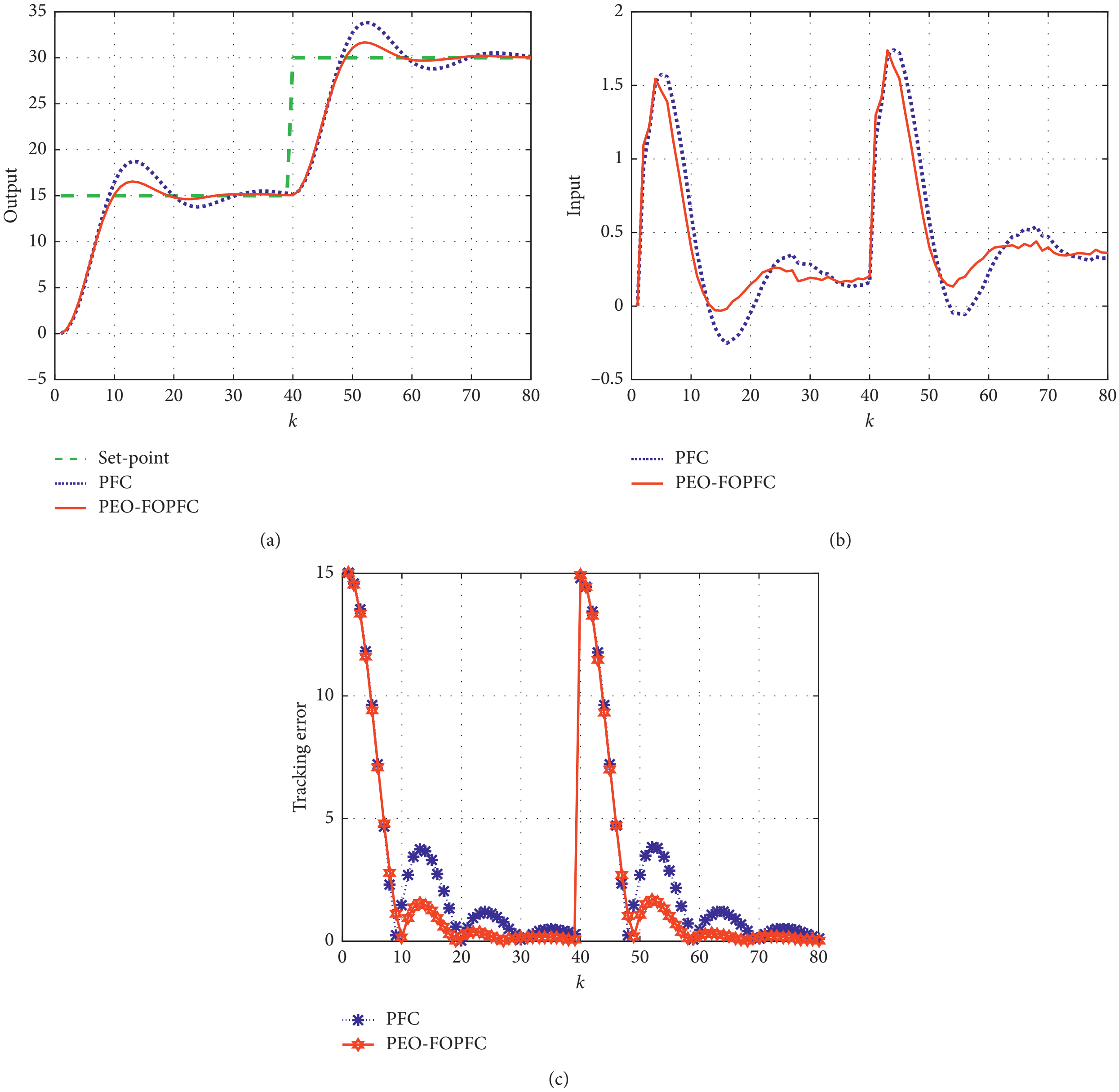


Figure 12: The results obtained by PEO-FOPFC and PFC under Case 3 in simulation 2. (a) Output responses; (b) input signals; (c) tracking error signals.

parameter γ used in FOPFC. Here, γ is set to 0.5 by the trail-and-error tuning approach. Figures 4–9 show the comparative results of output responses and input signals obtained by PFC and FOPFC under different six cases. In addition, Table 2 tabulates four performance indices. Clearly, the closed-loop system responses and input signals obtained by FOPFC are slightly better than those of PFC under Case 2 and Case 3. For Case 1, FOPFC has a similar performance to PFC. As another three server faults (i.e., Case 4, Case 5, and Case 6) with time-varying faults and a random white noise of standard deviation 0.2, FOPFC shows superiority to PFC, which indicates that the fractional-order mechanism have potential ability in solving more server faults.

4.3. Simulation 2: PEO-FOPFC and Its Comparison with PFC. In PEO-FOPFC, T_s is abstracted as 1 during the process of evolution because it has same influence on weighting factors, and $I_{\max}$, N_P, and b are set as 10, 50, and 3, respectively. The parameters including P, γ, and λ of FOPFC optimized by PEO are $P = 2$, $\gamma = 0.5659$, and $\lambda = 0.8231$. And, the faults are the same with six cases used in simulation 1. The comparison results are presented in Figures 10–15, where Figures 10(a)–15(a) show the output responses, Figures 10(b)–15(b) show the input control signals, and Figures 10(c)–15(c) show the tacking error under the six cases, respectively. Additionally, Table 3 presents the comparative results of control performance indices obtained by PEO-FOPFC and PFC [16] under six cases, where the best performance is highlighted in bold.

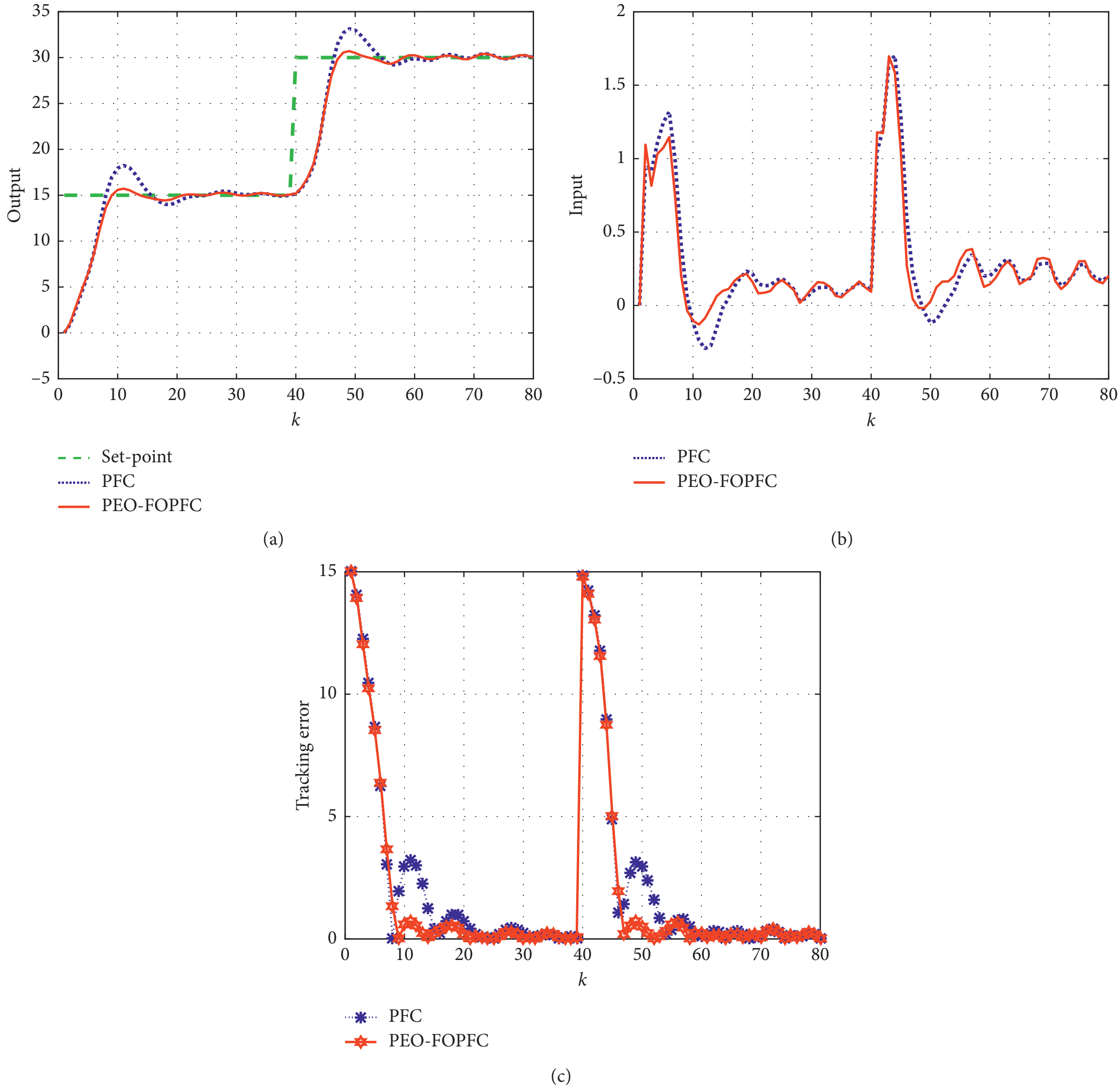
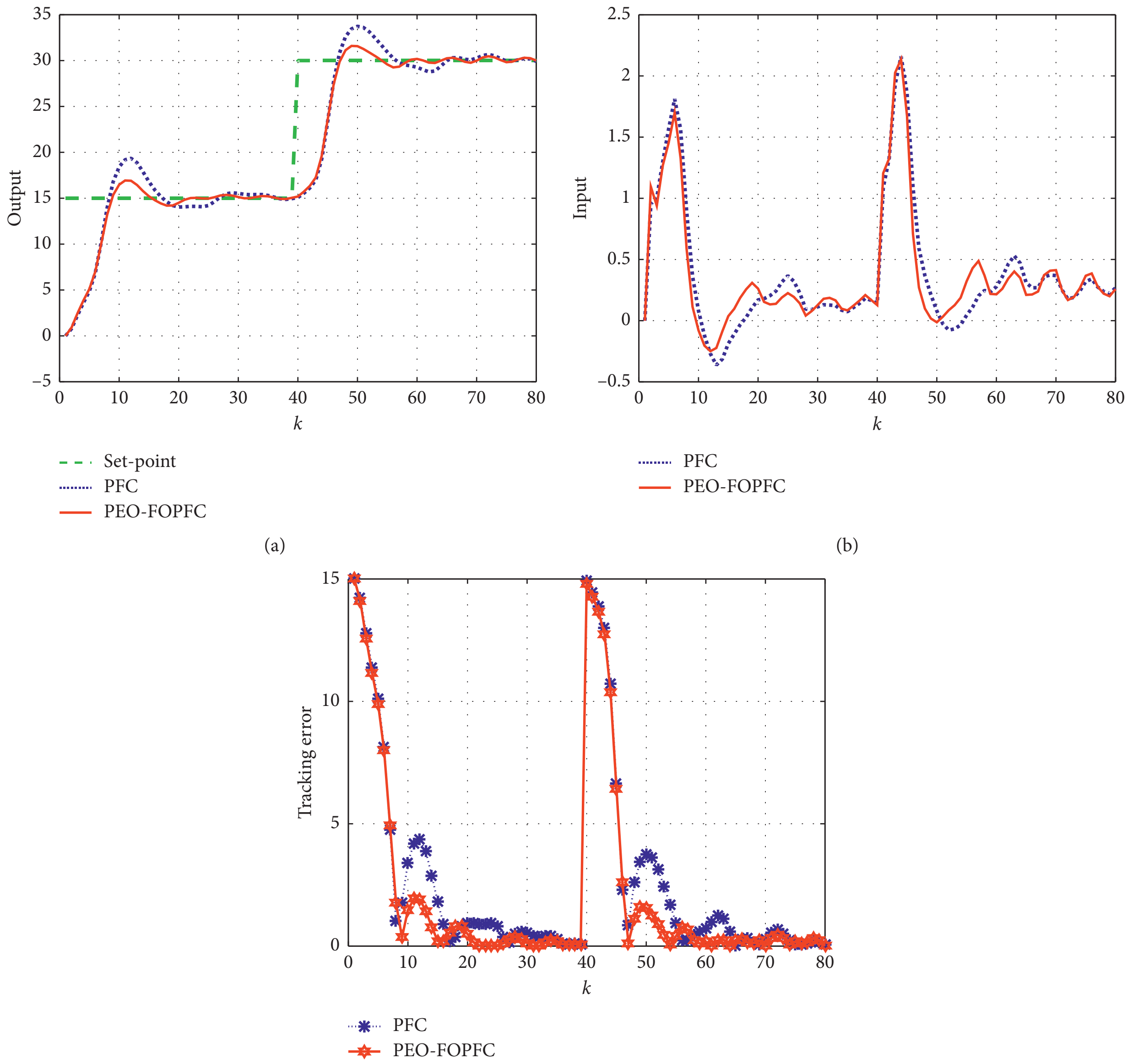


FIGURE 13: The results obtained by PEO-FOPFC and PFC under Case 4 in simulation 2. (a) Output responses; (b) input signals; (c) tracking error signals.

From Figures 10(a)–15(a), the proposed PEO-FOPFC presents better output tacking performance than PFC because it shows less oscillations and smaller overshoots. Observed from Figures 10(b)–15(b), we can see that the PFC [16] presents more drastic input control signals than PEO-FOPFC, which indicates the product quality obtained by PFC [16] is worse than PEO-FOPFC. Besides, the tacking error performance in Figures 10(c)–15(c) also illustrates the superiority of the proposed PEO-FOPFC.

4.4. Simulation 3: PEO-FOPFC and Its Comparison with FOPFC, GA-FOPFC, and PSO-FOPFC. In this section, we devote to investigating the influence of PEO algorithm on control performance of PEO-FOPFC. Thus, we compare PEO-FOPFC with FOPFC and two other popular evolutionary algorithms (i.e., GA [35] and PSO [36]) based FOPFC methods (termed as GA-FOPFC and PSO-FOPFC, respectively) under above six cases of faults. In order to clearly compare the three evolutionary algorithms-based FOPFC methods, we tabulate their related parameters and optimal parameters of FOPFC in Table 4.

From Table 4, we can see the number of adjustable parameters used in PEO is less than that in GA and PSO, which indicates the PEO algorithm is simpler than GA and PSO from the perspective of algorithm design. Figures 16(a)–16(f) show the results of four compared methods, and the corresponding performance indices are presented in Table 5. From Figure 16 and Table 5, the main observations can be summarized as follows: (1) it can be found that the performance of



(a)

(b)

(c)

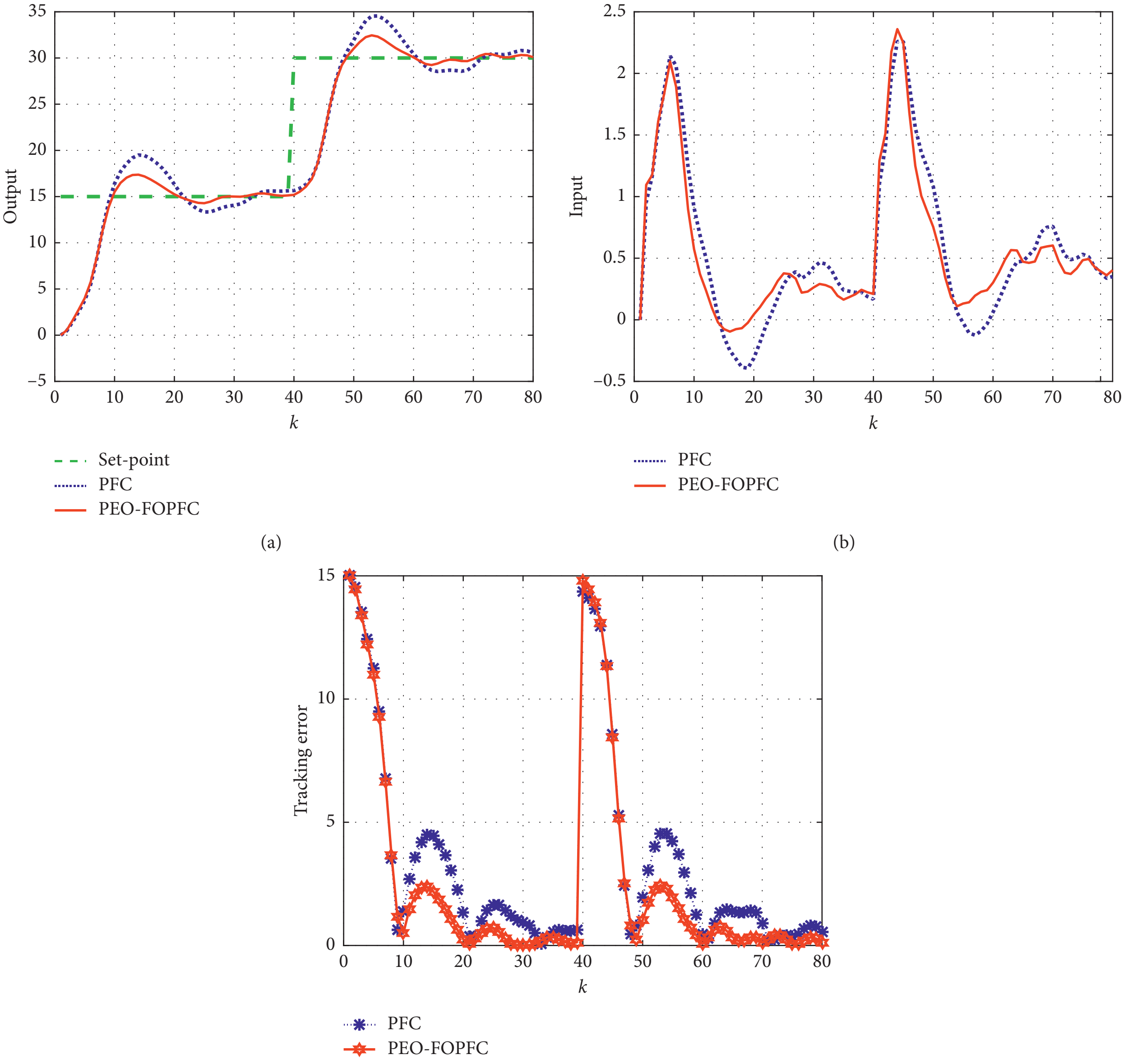
Figure 14: The results obtained by PEO-FOPFC and PFC under Case 5 in simulation 2. (a) Output responses; (b) input signals; (c) tracking error signals.

FOPFC for the industrial process with partial actuator failures can be improved by using evolutionary algorithms; (2) the PEO algorithm is better than GA and PSO algorithm to find the better related parameters of FOPFC in term of four indices, although M_d obtained by PEO-FOPFC is worse than that by GA-FOPFC and PSO-FOPFC in Cases 4–6; (3) the proposed PEO-FOPFC can improve ensemble control performance against partial actuator failures under both constant faults and time-varying faults.

Remark 8. In Table 1, the conditions of six actuator faults are listed and a different α is used. Smaller α value means more severe fault. From above simulations 1–3, we can find that worse control performance is obtained with smaller α values in terms of four control performance indices and output responses.

5. Simulation Results on Process Flow of Coke Furnace

In this section, we investigate the application of PEO-FOPFC into control of process flow of coke furnace under different partial faults and unknown disturbance. Figure 17 gives the sketch of process flow of coking furnace. The main job of this unit is coking residues oil. First, the residual oil is divided into two flows, i.e., FRC8103 and FRC8104, and the convection room of the process furnace (101/3) is preheated. Then, the separated streams will be merged into the fractionating tower, i.e., T102, where the preheated oil exchanges



(a)

(b)

(c)

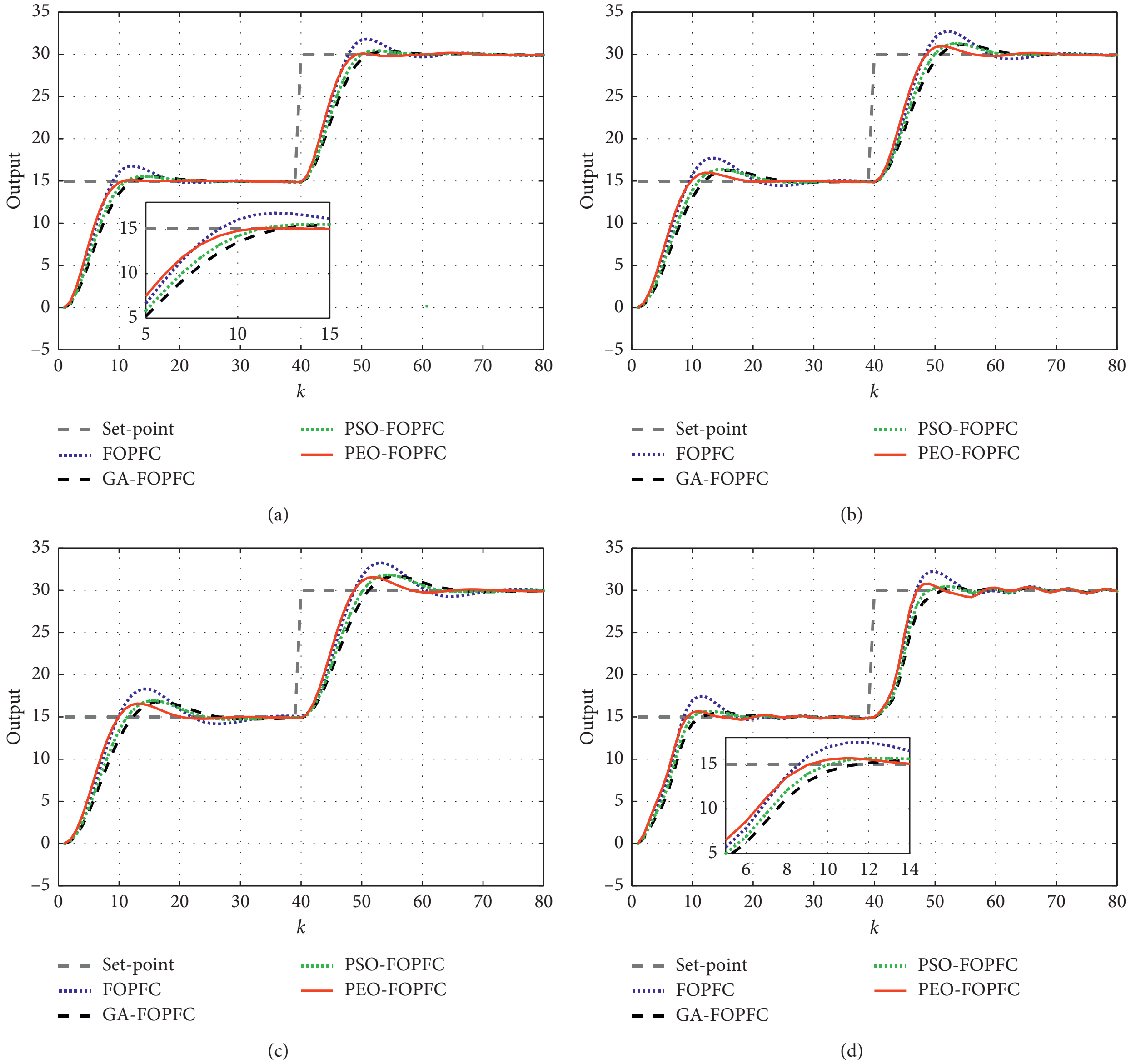
Figure 15: The results obtained by PEO-FOPFC and PFC under Case 6 in simulation 2. (a) Output responses; (b) input signals; (c) tracking error signals.

Table 3: The results of the control performance obtained by PEO-FOPFC and PFC in simulation 1.

Fault	Case	Algorithm	*IAE*	*ITAE*	M_d	*Ess*
Three constant faults	Case 1	PFC [16]	165.572	51.773	2.443	0.0511
		PEO-FOPFC	**143.502**	**42.159**	**0.1296**	**0.0152**
	Case 2	PFC [16]	205.998	68.744	3.435	0.1290
		PEO-FOPFC	**163.051**	**49.112**	**0.9378**	**0.0207**
	Case 3	PFC [16]	232.875	80.469	3.698	0.0706
		PEO-FOPFC	**181.602**	**56.049**	**1.511**	**0.0180**
Three time-varying faults	Case 4	PFC [16]	183.022	57.771	3.189	0.0317
		PEO-FOPFC	**154.261**	**46.632**	**0.691**	**0.0252**
	Case 5	PFC [16]	218.925	72.425	3.739	0.1219
		PEO-FOPFC	**178.013**	**55.044**	**1.587**	**0.0220**
	Case 6	PFC [16]	271.063	96.301	4.576	0.6553
		PEO-FOPFC	**213.876**	**69.374**	**2.377**	**0.0429**

Table 4: Related adjustable parameters used in three different evolutionary algorithms and corresponding optimal parameters in FOPFC.

Algorithm	Related adjustable parameters	Number of adjustable parameters	Optimal parameters in FOPFC
GA-FOPFC	Population size $NP = 30$, maximum number of iterations $I_{\max} = 50$, crossover probability $p_c = 0.8$, mutation parameter $b = 3$	4	$P = 4$, $\gamma = 2.3549$, $\lambda = 0.8650$
PSO-FOPFC	Swarm size $NP = 30$, maximum number of iterations $I_{\max} = 50$, acceleration factors $c_1 = 1.49$, $c_2 = 1.49$, limits of velocity $V_{\max} = 2$, $V_{\min} = -2$	6	$P = 3$, $\gamma = 0.8009$, $\lambda = 1.000$
PEO-FOPFC	Population size $NP = 10$, maximum number of iterations $I_{\max} = 50$, parameter $b = 3$ used in MNUM operation	3	$P = 2$, $\gamma = 0.5659$, $\lambda = 0.831$



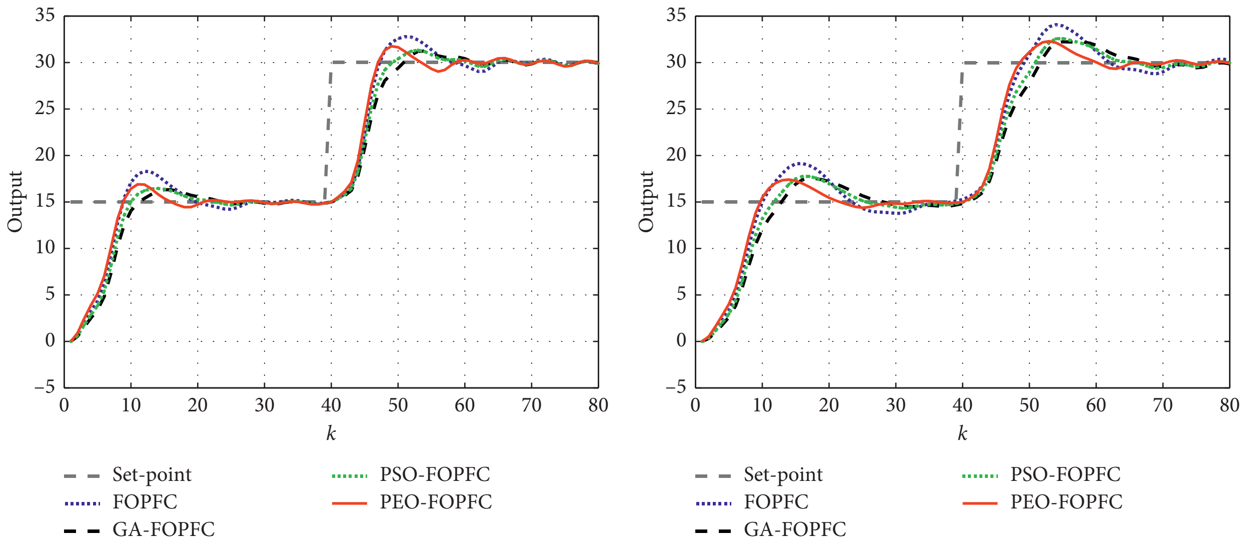
(a)

(b)

(c)

(d)

Figure 16: Continued.



(e)

(f)

Figure 16: The output responses obtained by PEO-FOPFC and FOPFC under six cases in simulation 3. (a) Case 1, (b) Case 2, (c) Case 3, (d) Case 4, (e) Case 5, and (f) Case 6.

Table 5: The results of control performance obtained by FOPFC and PFC in Simulation 3.

	Case	Algorithm	*IAE*	*ITAE*	M_d	*Ess*
Three constant faults	Case 1	FOPFC	167.798	52.217	2.098	0.0252
		GA-FOPFC	178.770	53.465	0.326	0.0436
		PSO-FOPFC	167.457	49.645	0.438	0.0248
		PEO-FOPFC	**143.502**	**42.159**	**0.129**	**0.0152**
	Case 2	FOPFC	204.480	67.094	3.018	0.0708
		GA-FOPFC	205.666	64.234	1.201	0.0360
		PSO-FOPFC	193.774	60.013	1.004	0.0261
		PEO-FOPFC	**163.051**	**49.112**	**0.938**	**0.0207**
	Case 3	FOPFC	231.065	78.484	3.312	0.0162
		GA-FOPFC	227.807	73.855	1.719	**0.0155**
		PSO-FOPFC	215.677	69.338	1.857	0.0221
		PEO-FOPFC	**181.602**	**56.049**	**1.511**	0.0180
Three time-varying faults	Case 4	FOPFC	180.799	56.268	2.381	0.0266
		GA-FOPFC	179.445	55.383	**0.601**	0.0360
		PSO-FOPFC	171.842	51.918	0.622	0.0360
		PEO-FOPFC	**154.261**	**46.632**	0.691	**0.0252**
	Case 5	FOPFC	213.976	69.688	2.981	0.0675
		GA-FOPFC	207.156	65.122	**1.309**	0.0884
		PSO-FOPFC	196.743	61.116	1.415	0.0316
		PEO-FOPFC	**178.013**	**55.044**	1.587	**0.0220**
	Case 6	FOPFC	267.114	94.254	4.270	0.5764
		GA-FOPFC	264.112	90.721	**2.291**	0.1552
		PSO-FOPFC	249.106	84.295	2.591	**0.0306**
		PEO-FOPFC	**213.876**	**69.374**	2.377	0.0429

heat with the gas oil from the top of the coke towers. After that, the circulating oil, i.e., the heavy part of mixed oil, is pumped into two flows and sent back to the radiation room for the purpose of heating to about 495 °C. Finally, the two branches join together and will be sent to the coke towers to remove coke. Each time the output temperature in the chamber has a large impact on this process. Thus, the output temperature needs to be accurately controlled in the coke furnace under different partial faults and unknown disturbance. As suggested in [37], a first-order flus dead time (FOPDT) can be used to test and can be described as follows:

$$G(s) = \frac{1.1}{300s + 1} e^{-150s}. \tag{37}$$

Under sampling time $T_s = 30$ s, the discrete-time model can be obtained as follows [37]:

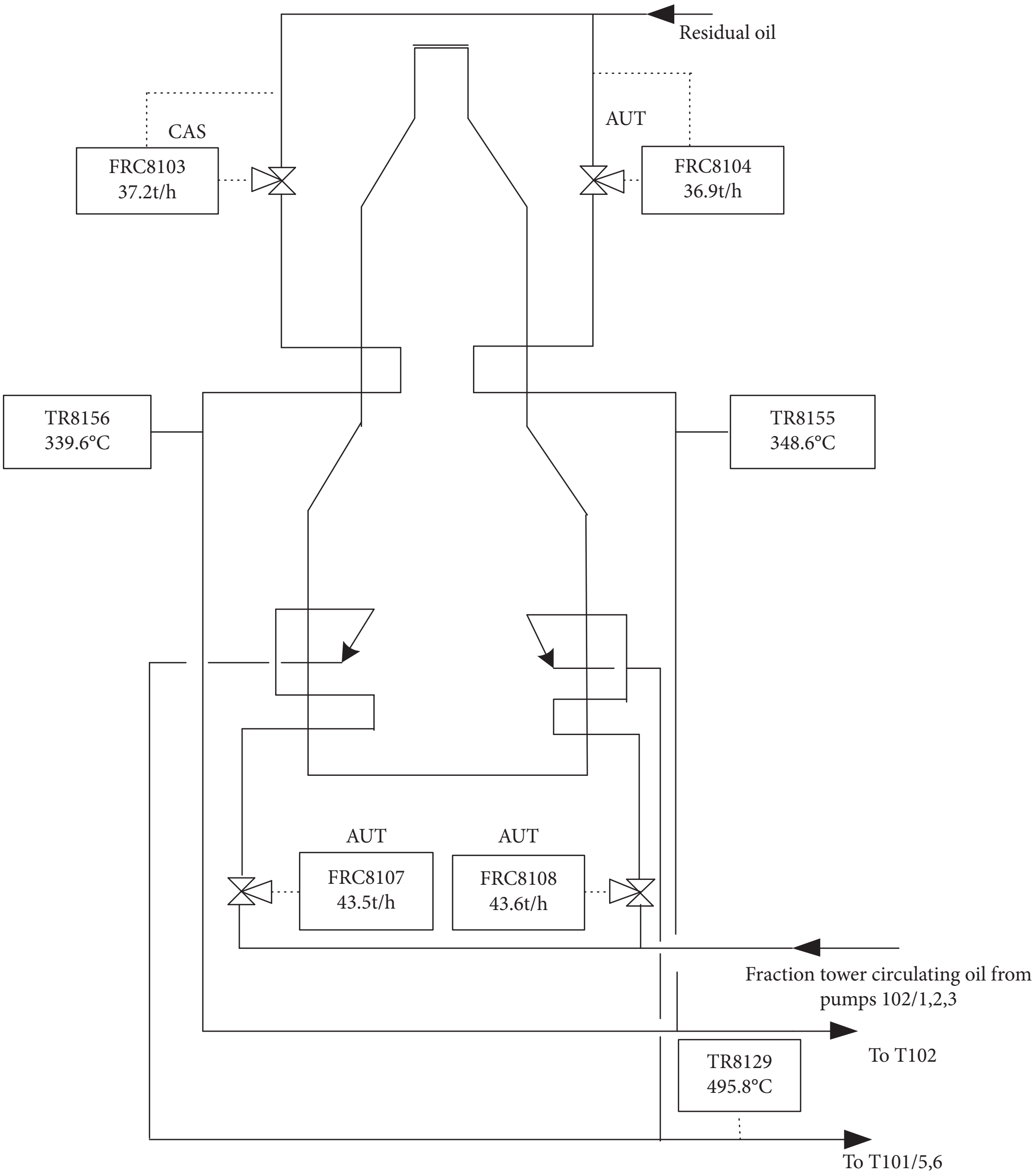
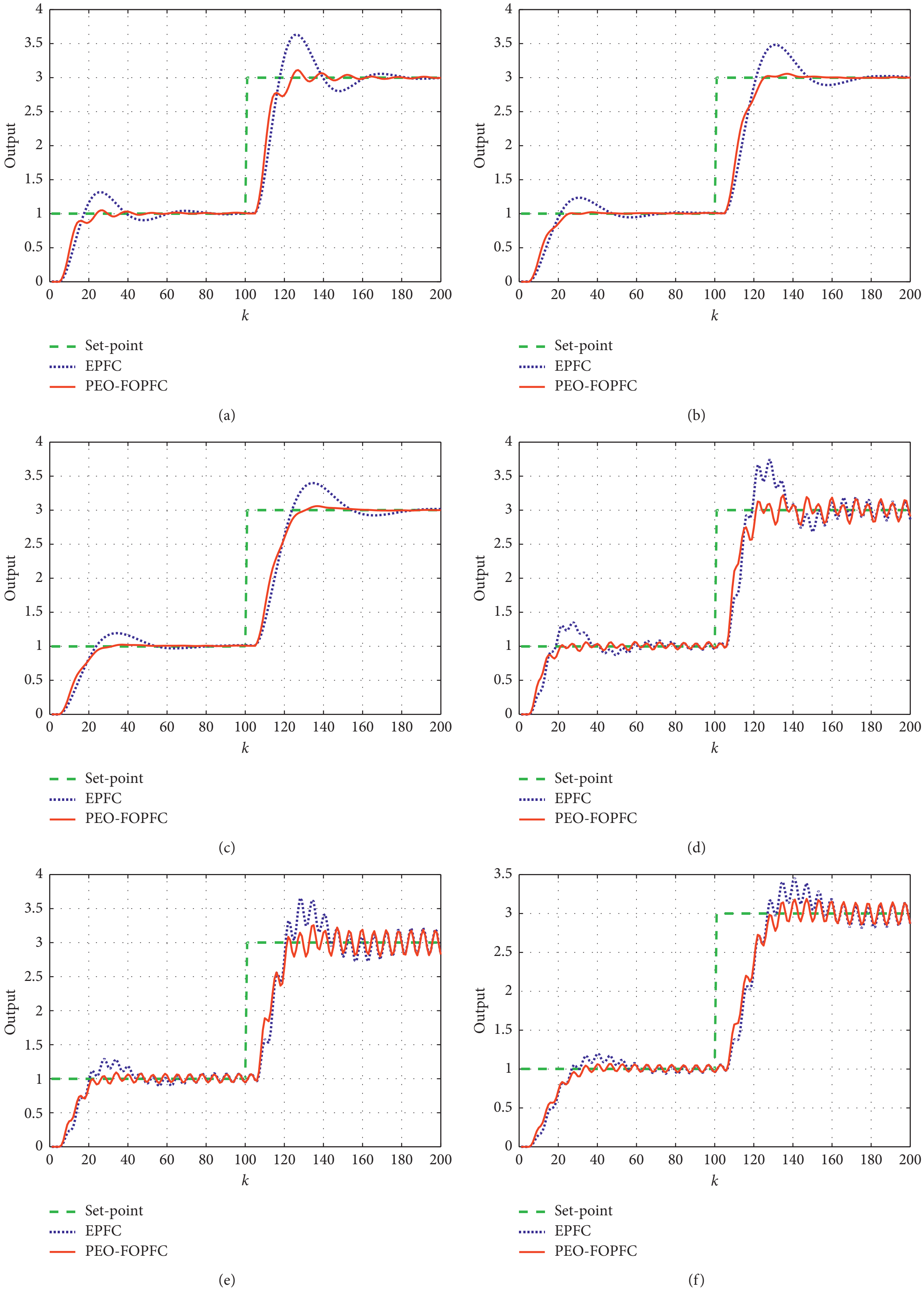


Figure 17: Process flow of coking furnace (101/3) [37].

Table 6: The results of control performance obtained by PEO-FOPFC and PFC on the process flow of coke furnace.

	Case	Algorithm	*IAE*	*ITAE*	M_d	*Ess*
Three constant faults	Case 1	EPFC [37]	52.995	54.386	0.265	0.203
		PEO-FOPFC	**35.358**	**43.013**	**0.177**	**0.185**
	Case 2	EPFC [37]	54.759	58.230	0.274	0.217
		PEO-FOPFC	**38.795**	**48.190**	**0.194**	**0.204**
	Case 3	EPFC [37]	56.523	61.266	0.283	0.228
		PEO-FOPFC	**42.903**	**52.310**	**0.215**	**0.217**
Three time-varying faults	Case 4	EPFC [37]	56.182	54.715	0.281	0.196
		PEO-FOPFC	**42.433**	**43.666**	**0.212**	**0.174**
	Case 5	EPFC [37]	61.111	59.624	0.306	0.206
		PEO-FOPFC	**50.230**	**49.832**	**0.251**	**0.187**
	Case 6	EPFC [37]	63.899	67.314	0.319	0.236
		PEO-FOPFC	**55.613**	**59.116**	**0.278**	**0.219**



(a)

(b)

(c)

(d)

(e)

(f)

Figure 18: The output responses obtained by PEO-FOPFC and FOPFC under six cases on process flow of coke furnace. (a) Case 1, (b) Case 2, (c) Case 3, (d) Case 4, (e) Case 5, and (f) Case 6.

$$G(z)) = \frac{0.1047}{z^{6} - 0.9048 z^{5}}. \qquad (38)$$

Remark 9. The set-point is abstracted into 1 to 3, and the six faults considered here are the same with simulations 1–3 given in Table 1. The used adjustable parameters of PEO are also the same with simulations 1–3. The competitor is taken from [37], called predictive functional control, based on the extended state space model (denoted as EPFC here).

By employing PEO, we can obtain $P = 5$, $\gamma = 5.6773$, and $\lambda = 0.8335$ for this simulation. Figures 18(a)–18(f) compare the output responses of the set-point tracking obtained by PEO-FOPFC and EPFC. Table 6 presents the results of control performance obtained by PEO-FOPFC and PFC on the process flow of coke furnace. From Figure 18 and Table 6, one can see that PEO-FOPFC shows an improvement over EPFC under partial faults and unknown disturbance considered in the paper, which implies the PEO algorithm and fractional-order theory contribute a lot to obtain a better comprehensive control performance.

6. Conclusion

In this paper, we have proposed the PEO-FOPFC strategy for industrial processes under partial actuator failures. There are two key operators in the PEO-FOPFC. More specifically, one is the application of fractional-order calculus into the PFC technique to improve its performance against partial actuator failures, and the other one is employing an effective PEO algorithm to search the optimal-related parameters in FOPFC. To demonstrate the strong competitiveness, the PEO-FOPFC strategy has been evaluated on two industrial processes under six cases considering constant faults, time-varying faults, and nonrepetitive unknown disturbance, compared with the recently developed PFC [16, 37]. In addition, in order to purely verify the effectiveness of the fractional-order mechanism, the experiment is designed for comparing FOPFC with PFC. And, another experiment is used to purely evaluate the influence of PEO algorithm by comparing PEO-FOPFC with FOPFC, GA-FOPFC, and PSO-FOPFC. The three experimental comparison results imply that the fractional-order mechanism and PEO algorithm are largely improving the performance of PFC to maintain the control performance and against the partial actuator faults. As a consequence, the proposed PEO-FOPFC strategy can be considered as a strong competitive version of PFC strategy for industrial processes against partial actuator failures.

Overall, by introducing the fractional-order mechanism into the cost function, the FOPFC can be designed to improve the performance of PFC due to more tuning parameters. The application of PEO to tuning the adjustable of FOPFC can handle the difficulty in lacking the knowledge of analytical these parameters. However, the parameter tuning issue of FOPFC is based on the designed fitness function, which may change for different control systems. Also, the PEO algorithm costs more computation time.

In future, it will be a significant subject to extend the proposed PFO-FOPFC into more complex industrial processes and design more advanced controller for handling actuator faults [38].

Appendix

Proof of Theorem 1. This proof is extended from [16].

From equation (26), we can obtain that

$$\Phi^{T} \Lambda [\Phi \Upsilon - G u(k-1)] = -\Phi^{T} \Lambda (F z(k) + S \Delta R). \qquad (A.1)$$

From equations (20), (23), and (A.1), we can derive that

$$\begin{aligned}
\Phi \Upsilon - G u(k-1) &= \begin{bmatrix} B T_{0} \\ (AB - B) T_{0} + B T_{1} \\ (A^{2}B - AB) T_{0} + (AB - B) T_{1} + B T_{2} \\ \vdots \\ \sum_{i=1}^{P} (A^{i}B - A^{i-1}B) T_{P-1-i} + B T_{P-1} \end{bmatrix} \Upsilon - \begin{bmatrix} B \\ AB \\ A^{2}B \\ \vdots \\ A^{P-1}B \end{bmatrix} u(k-1) \\
&= \begin{bmatrix} B T_{0} \Upsilon - B u(k-1) \\ (AB - B) T_{0} \Upsilon + B T_{1} \Upsilon - AB u(k-1) \\ (A^{2}B - AB) T_{0} \Upsilon + (AB - B) T_{1} \Upsilon + B T_{2} \Upsilon - A^{2} B u(k-1) \\ \vdots \\ \sum_{i=1}^{P} (A^{i}B - A^{i-1}B) T_{P-1-i} \Upsilon + B T_{P-1} \Upsilon - A^{P-1} B u(k-1) \end{bmatrix}
\end{aligned}$$

$$= \begin{bmatrix} Bu(k) - Bu(k-1) \\ (AB - B)u(k) + Bu(k+1) - ABu(k-1) \\ (A^2B - AB)u(k) + (AB - B)u(k+1) + Bu(k+2) - A^2Bu(k-1) \\ \vdots \\ \sum_{i=1}^{P} (A^iB - A^{i-1}B)u(k+P-1-i) + Bu(k+P-1) - A^{P-1}Bu(k-1) \end{bmatrix}$$

$$= \begin{bmatrix} B & 0 & 0 & \cdots & 0 \\ AB & B & 0 & \cdots & 0 \\ A^2B & AB & B & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A^{P-1}B & A^{P-2}B & A^{P-3}B & \cdots & B \end{bmatrix} \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \Delta u(k+2) \\ \vdots \\ \Delta u(k+P-1) \end{bmatrix} = \Omega\Delta U. \tag{A.2}$$

Based on equations (A.2) and (A.1),

$$\Phi^T \Lambda\Omega\Delta U = -\Phi^T \Lambda (Fz(k) + S\Delta R). \tag{A.3}$$

Then, by premultiplying both sides of equation (A.3) with $\Omega^T\Lambda$, we can obtain

$$\Delta U = -\left(\Omega^T\Phi\Phi^T\Lambda\Omega\right)^{-1}\Omega^T\Phi\Phi^T\Lambda (Fz(k) + S\Delta R). \tag{A.4}$$

Here, let C_S and C_R be subject to the following equations:

$$\begin{aligned} C_S &= (1, 0, \ldots, 0)\left(\Omega^T\Phi\Phi^T\Lambda\Omega\right)^{-1}\Omega^T\Phi\Phi^T\Lambda F, \\ C_R &= (1, 0, \ldots, 0)\left(\Omega^T\Phi\Phi^T\Lambda\Omega\right)^{-1}\Omega^T\Phi\Phi^T\Lambda S. \end{aligned} \tag{A.5}$$

At time instant k, the control increment can be described as

$$\Delta u(k) = -C_S z(k) - C_R\Delta R. \tag{A.6}$$

Without loss of generality, when considering closed-loop stability, we can chose the set-point and let $\Delta R = 0$. Then, the proposed control law can be rewritten as follows:

$$\Delta u(k) = -C_S z(k). \tag{A.7}$$

According to the derivation of equations (7)–(15), we can obtain following $z(k+1)$ form of uncertain system:

$$z(k+1) = (A + \Delta A)z(k) + B\Delta u(k) + C\Delta r(k+1). \tag{A.8}$$

Combining equations (A.7) and (A.8), $z(k+1)$ can be rewritten as

$$z(k+1) = \left(A - BC_S\right)z(k) + \Delta Az(k). \tag{A.9}$$

Consider a Lyapunov functional candidate between the time instants k and $k+1$ as

$$\begin{aligned} \Delta V(z(k)) &= V(z(k+1)) - V(z(k)), \\ &= z^T(k+1)W_P z(k+1) - z^T(k)W_P z(k). \end{aligned} \tag{A.10}$$

By substituting equations (A.9) into (A.10), $\Delta V(z(k))$can be rewritten as

$$\begin{aligned} \Delta V(z(k)) = {} & z^T(k)\left(A - BC_S\right)^T W_P \left(A - BC_S\right)z(k) \\ & + z^T(k)\left(A - BC_S\right)^T W_P\Delta Az(k) \\ & + z^T(k)\Delta A^T W_P\left(A - BC_S\right)z(k) + z^T(k)\Delta A^T \\ & \cdot W_P\Delta Az(k) - z^T(k)W_P z(k). \end{aligned} \tag{A.11}$$

Considering the following inequations,

$$\begin{aligned} & z^T(k)\left[\left(A - BC_S\right)^T W_P\left(A - BC_S\right)\right]z(k) - z^T(k)W_P z(k) \\ & \quad \le -\lambda_{\min}(M_P)\|z(k)\|^2 \\ & z^T(k)\left(A - BC_S\right)^T W_P\Delta Az(k) + z^T(k)\Delta A^T W_P\left(A - BC_S\right)z(k) \\ & \quad \le -2\sigma_{\max}\left(A - BC_S\right)\lambda_{\max}(W_P)\|\Delta A\|\|z(k)\|^2, \end{aligned} \tag{A.12}$$

we can get

$$\begin{aligned} \Delta V(z(k)) \le {} & \|z(k)\|^2\left(-\lambda_{\min}(M_P) + 2\sigma_{\max}\left(A - BK_s\right)\lambda_{\max}\right. \\ & \left. \cdot (W_P)\|\Delta A\| + \lambda_{\max}(W_P)\|\Delta A\|^2\right). \end{aligned} \tag{A.13}$$

If the following inequations holds,

$$\begin{aligned} & -\sigma_{\max}\left(A - BC_S\right) - \sqrt{\sigma_{\max}{}^2\left(A - BC_S\right) + \frac{\lambda_{\min}(M_P)}{\lambda_{\max}(W_P)}} \\ & \le \|\Delta A\| = \sigma_{\max}(\Delta A) \\ & \le -\sigma_{\max}\left(A - BC_S\right) + \sqrt{\sigma_{\max}{}^2\left(A - BC_S\right) + \frac{\lambda_{\min}(M_P)}{\lambda_{\max}(W_P)}}, \end{aligned} \tag{A.14}$$

then, $\Delta V(z(k)) < 0$ is guaranteed, which implies the robust stability of the control system.

This completes the proof.

Proof of Proposition 1. This proof is extended from [34].

Proof. Suppose that the transfer function is $A_d(z)/B_d(z)$, where model order is d. Here, we write the gain vector C_S and C_R in equation (A.6) as $C_S = [c_1, c_2, \ldots, c_d, c_{d+1}, c_{d+2}, \ldots, c_{2d}]$ and $C_R = [c_{r1}, c_{r2}, \ldots, c_{rP}]$ and define two polynomial functions $M(z)$, $N(z)$ as follows:

$$
\begin{aligned}
M(z) &= c_1 + c_2 z^{-1} + c_3 z^{-2} + \cdots + c_d z^{-(d-1)},\\
N(z) &= 1 + c_{d+1} z^{-1} + c_{d+2} z^{-2} + \cdots + c_{2d-1} z^{-(d-1)}.
\end{aligned}
\tag{A.15}
$$

Additionally, ΔR can be rewritten as

$$
\begin{aligned}
\Delta R &= \begin{bmatrix} \Delta r(k+1) \\ \Delta r(k+2) \\ \vdots \\ \Delta r(k+P) \end{bmatrix} = \begin{bmatrix} r(k+1) - r(k) \\ r(k+2) - r(k+1) \\ \vdots \\ r(k+P) - r(k+P-1) \end{bmatrix}\\
&= \begin{bmatrix} (1-\lambda)(c(k) - y(k)) \\ \lambda(1-\lambda)(c(k) - y(k)) \\ \vdots \\ \lambda^{P-1}(1-\lambda)(c(k) - y(k)) \end{bmatrix}.
\end{aligned}
\tag{A.16}
$$

According to the control law equations (A.6), (15), and (A.16), we define

$$
C_R' = c_{r1}(1-\lambda) + c_{r2}\lambda(1-\lambda) + \cdots + c_{rP}\lambda^{P-1}(1-\lambda).
\tag{A.17}
$$

Then, we can derive the control low in the following polynomial form:

$$
\left(1 - z^{-1}\right)N(z)U(z) = -\left(1 - z^{-1}\right)M(z)Y(z) + C_R'(C(z) - Y(z)).
\tag{A.18}
$$

Afterwards, we can get the transfer function $T(z)$ described from the set-point to the output as

$$
T(z) = \frac{C_R' B_d(z)}{\left(1 - z^{-1}\right)\left(M(z)A_d(z) + N(z)B_d(z)\right) + C_R' B_d(z)},
\tag{A.19}
$$

which can obtain

$$
\lim_{z \longrightarrow 1} T(z) = 1.
\tag{A.20}
$$

If the equation (A.20) satisfied, then we can conclude that the closed-loop system tracks the set-point without steady error.

Additionally, the transfer function from the output disturbance and input disturbance to the output response can be obtained, respectively, as follows:

$$
\begin{aligned}
S_O(z) &= \frac{\left(1 - z^{-1}\right)N(z)A_d(z)}{\left(1 - z^{-1}\right)\left(N(z)A_d(z) + M(z)B_d(z)\right) + C_R' B_d(z)},\\
S_I(z) &= \frac{\left(1 - z^{-1}\right)N(z)B_d(z)}{\left(1 - z^{-1}\right)\left(N(z)A_d(z) + M(z)B_d(z)\right) + C_R' B_d(z)}.
\end{aligned}
\tag{A.21}
$$

One can see that

$$
\begin{aligned}
\lim_{z \longrightarrow 1} S_O(z) &= 0,\\
\lim_{z \longrightarrow 1} S_I(z) &= 0,
\end{aligned}
\tag{A.22}
$$

which indicates that the constant output disturbances and input disturbances can be rejected without steady error.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (Grant no. 61972288) and Zhejiang Provincial of Natural Science Foundation (Grant no. LZ16E050002).

References

[1] R. Zhang, R. Lu, A. Xue, and F. Gao, “New minmax linear quadratic fault-tolerant tracking control for batch processes,” *IEEE Transactions on Automatic Control*, vol. 61, no. 10, pp. 3045–3051, 2016.

[2] X. Yu and J. Jiang, “Hybrid fault-tolerant flight control system design against partial actuator failures,” *IEEE Transactions on Control Systems Technology*, vol. 20, no. 4, pp. 871–886, 2012.

[3] J. Tao, Z. Yu, Y. Zhu, and L. Ma, “A linear quadratic structure based predictive functional control design for industrial processes against partial actuator failures,” *Chemometrics and Intelligent Laboratory Systems*, vol. 146, pp. 263–269, 2015.

[4] Y. Wang, J. Shi, D. Zhou, and F. Gao, “Iterative learning fault-tolerant control for batch processes,” *Industrial & Engineering Chemistry Research*, vol. 45, no. 26, pp. 9050–9060, 2006.

[5] A. Giridhar and N. H. El-Farra, “A unified framework for detection, isolation and compensation of actuator faults in uncertain particulate processes,” *Chemical Engineering Science*, vol. 64, no. 12, pp. 2963–2977, 2009.

[6] X. Jin, “Iterative learning control for output-constrained nonlinear systems with input quantization and actuator faults,” *International Journal of Robust and Nonlinear Control*, vol. 28, no. 2, pp. 729–741, 2018.

[7] J. Ding, B. Cichy, K. Galkowski, E. Rogers, and H. Yang, “Parameter-dependent Lyapunov function-based robust iterative learning control for discrete systems with actuator faults,” *International Journal of Adaptive Control and Signal Processing*, vol. 30, no. 12, pp. 1714–1732, 2016.

[8] T. Zou, S. Wu, and R. Zhang, “Improved state space model predictive fault-tolerant control for injection molding batch processes with partial actuator faults using GA optimization,” *ISA Transactions*, vol. 73, pp. 147–153, 2018.

[9] C. A. Harrison and S. J. Qin, “Discriminating between disturbance and process model mismatch in model predictive control,” *Journal of Process Control*, vol. 19, no. 10, pp. 1610–1616, 2009.

[10] J. H. Lee and K. S. Lee, “Iterative learning control applied to batch processes: an overview,” *Control Engineering Practice*, vol. 15, no. 10, pp. 1306–1318, 2007.

[11] B. Chu, D. H. Owens, and C. T. Freeman, “Iterative learning control with predictive trial information: convergence, robustness, and experimental verification,” *IEEE Transactions on Control Systems Technology*, vol. 24, no. 3, pp. 1101–1108, 2016.

[12] L. Wang, F. Liu, J. Yu, P. Li, R. Zhang, and F. Gao, “Iterative learning fault-tolerant control for injection molding processes

against actuator faults," *Journal of Process Control*, vol. 59, pp. 59–72, 2017.

[13] R. Zhang, S. Wu, and F. Gao, "State space model predictive control for advanced process operation: a review of recent development, New Results, and Insight," *Industrial & Engineering Chemistry Research*, vol. 56, no. 18, pp. 5360–5394, 2017.

[14] M. Elsisi, M. Soliman, M. A. S. Aboelela, and W. Mansour, "Bat inspired algorithm based optimal design of model predictive load frequency control," *International Journal of Electrical Power & Energy Systems*, vol. 83, pp. 426–433, 2016.

[15] A. Zakharov, E. Zattoni, M. Yu, and S.-L. Jämsä-Jounela, "A performance optimization algorithm for controller reconfiguration in fault tolerant distributed model predictive control," *Journal of Process Control*, vol. 34, pp. 56–69, 2015.

[16] R. Zhang, R. Lu, A. Xue, and F. Gao, "Predictive functional control for linear systems under partial actuator faults and application on an injection molding batch process," *Industrial & Engineering Chemistry Research*, vol. 53, no. 2, pp. 723–731, 2014.

[17] R. Zhang, H. Zou, A. Xue, and F. Gao, "GA based predictive functional control for batch processes under actuator faults," *Chemometrics and Intelligent Laboratory Systems*, vol. 137, pp. 67–73, 2014.

[18] X. Hu, H. Zou, and L. Wang, "Design of the linear quadratic structure based predictive functional control for industrial processes against partial actuator failures using GA optimization," *International Journal of Control, Automation and Systems*, vol. 17, no. 3, pp. 597–605, 2019.

[19] N. Bigdeli, "The design of a non-minimal state space fractional-order predictive functional controller for fractional systems of arbitrary order," *Journal of Process Control*, vol. 29, pp. 45–56, 2015.

[20] Q. Zou, Q. Jin, and R. Zhang, "Design of fractional order predictive functional control for fractional industrial processes," *Chemometrics and Intelligent Laboratory Systems*, vol. 152, pp. 34–41, 2016.

[21] M. Sanatizadeh and N. Bigdeli, "The design of NMSS fractional-order predictive functional controller for unstable systems with time delay," *ISA Transactions*, vol. 92, pp. 49–64, 2019.

[22] X. Hu, Q. Zou, and H. Zou, "Design and application of fractional order predictive functional control for industrial heating furnace," *IEEE Access*, vol. 6, pp. 66565–66575, 2018.

[23] L.-M. Li, K.-D. Lu, G.-Q. Zeng, L. Wu, and M.-R. Chen, "A novel real-coded population-based extremal optimization algorithm with polynomial mutation: a non-parametric statistical study on continuous optimization problems," *Neurocomputing*, vol. 174, pp. 577–587, 2016.

[24] S. Boettcher and A. Percus, "Nature's way of optimizing," *Artificial Intelligence*, vol. 119, no. 1-2, pp. 275–286, 2000.

[25] G.-Q. Zeng, J. Chen, L.-M. Li et al., "An improved multi-objective population-based extremal optimization algorithm with polynomial mutation," *Information Sciences*, vol. 330, pp. 49–73, 2016.

[26] G.-Q. Zeng, J. Chen, M.-R. Chen et al., "Design of multivariable PID controllers using real-coded population-based extremal optimization," *Neurocomputing*, vol. 151, pp. 1343–1353, 2015.

[27] G. Zeng, H. Liu, D. Wu et al., "A real-coded extremal optimization method with multi-non-uniform mutation for the design of fractional order PID controllers," *Information Technology and Control*, vol. 45, no. 4, pp. 358–375, 2016.

[28] F. Zhao, G.-Q. Zeng, and K.-D. Lu, "EnLSTM-WPEO: Short-term traffic flow prediction by ensemble LSTM, NNCT weight integration and population extremal optimization," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 101–113, 2020.

[29] C. A. Monje, Y. Chen, B. M. Vinagre, D. Xue, and V. Feliu, *Fractional-Order Systems and Controls: Fundamentals and Applications*, Springer-Verlag, London, UK, 2010.

[30] M. Romero, A. P. de Madrid, and M. B. Vinagre, "Arbitrary real-order cost functions for signals and systems," *Signal Processing*, vol. 91, no. 3, pp. 372–378, 2011.

[31] K. Lu, W. Zhou, G. Zeng, and W. Du, "Design of PID controller based on a self-adaptive state-space predictive functional control using extremal optimization method," *Journal of the Franklin Institute*, vol. 355, no. 5, pp. 2197–2220, 2018.

[32] R. Zhang, Q. Jin, and F. Gao, "Design of state space linear quadratic tracking control using GA optimization for batch processes with partial actuator failure," *Journal of Process Control*, vol. 26, pp. 102–114, 2015.

[33] P. Hu and P. Yuan, "The robust stability conditions of the state feedback model predictive control systems," *Control Decision*, vol. 16, pp. 123–127, 2001.

[34] R. Zhang, A. Xue, S. Wang, and J. Zhang, "An improved state-space model structure and a corresponding predictive functional control design with improved control performance," *International Journal of Control*, vol. 85, no. 8, pp. 1146–1161, 2012.

[35] P.-H. Tang and M.-H. Tseng, "Adaptive directed mutation for real-coded genetic algorithms," *Applied Soft Computing*, vol. 13, no. 1, pp. 600–614, 2013.

[36] N. Lynn and P. N. Suganthan, "Ensemble particle swarm optimizer," *Applied Soft Computing*, vol. 55, pp. 533–548, 2017.

[37] R. Zhang, A. Xue, and F. Gao, *Model Predictive Control*, Springer, Berlin, Germany, 2019.

[38] M. Zhang, C. Shen, P. Shi, and Z. Wu, "Static output feedback control of switched nonlinear systems with actuator faults," *IEEE Transactions on Fuzzy Systems*, 2019.