ISSN 1392–124X (print), ISSN 2335–884X (online) INFORMATION TECHNOLOGY AND CONTROL, 2016, T. 45, Nr. 4

A Real-coded Extremal Optimization Method with Multi-non-uniform Mutation for the Design of Fractional Order PID Controllers

Guo-Qiang Zeng, Hai-Yang Liu, Di Wu, Li-Min Li*, Lie Wu, Yu-Xing Dai

National-Local Joint Engineering Laboratory of Digitalize Electrical Design Technology & Department of Electrical and Electronic Engineering, Wenzhou University, Wenzhou, 325035, China
e-mail:158397411@qq.com, lilimin@wzu.edu.cn

Kang-Di Lu

Department of Automation, College of Information Sciences and Technology, Donghua University, Shanghai 201620, China

http://dx.doi.org/10.5755/j01.itc.45.4.13310

Abstract. Design of an effective and efficient fractional order PID (FOPID) controller for an industrial control system to obtain high-quality performances is of great theoretical and practical significance. This paper presents a novel real-coded extremal optimization algorithm with multi-non-uniform mutation called RCEO-FOPID to design FOPID controllers. The key idea behind the proposed algorithm is the population-based iterated optimization, which consists of generation of a real-coded random initial population by encoding the parameters of a FOPID controller into a set of real values, evaluation of the individual fitness by using a novel and reasonable control performance index, generation of a new population based on multi-non-uniform mutation and updating the population by accepting the new population unconditionally. The proposed RCEO algorithm for the design of FOPID controller is relatively simpler than these reported popular evolutionary algorithms, e.g., genetic algorithm (GA), particle swarm optimization (PSO), chaotic anti swarm (CAS) due to its fewer adjustable parameters and only with selection and mutation operators. Furthermore, extensive simulation results on automatic voltage regulator system and multivariable control system have shown that the proposed RCEO-based FOPID controller is superior to other reported evolutionary algorithms-based FOPID and PID controllers in terms of accuracy and robustness.

Keywords: extremal optimization; fractional order PID controller; multi-non-uniform mutation; automatic voltage regulator system; multivariable control system.

1. Introduction

A variety of advancements have been gained in control theories and practices in the past decades [1-7], but Proportional-Integral-Derivative (PID) control is still widely recognized as one of the simplest but most efficient control strategies in the control industry [8, 9]. Fractional order PID (FOPID) controller namely $PI^{\lambda}D^{\mu}$ controller [10] is a generalization of a standard PID controller based on fractional order calculus, and it has the ability to provide better control performance than standard integer order PID controller due to extra degrees of freedom introduced by an integrator of fractional order λ and a differentiator of fractional order μ. Consequently, FOPID controller has attracted increasing attention by the academic and industrial community [11-18] in the recent years. On the other hand, the introduction of extra parameters in a FOPID controller also increases the difficulty of tuning satisfied values of parameters, so how to design and tune an optimal FOPID controller to obtain high-quality performances including high stability, satisfied transient response, excellent steady performance, and good robustness, is of great theoretical and practical significance, but is still an open issue. Some researchers have made a great deal of efforts to deal with this issue by means of analytic methods [19-25] and evolutionary algorithms-based methods, e.g., genetic algorithm (GA) [13], chaotic ant swarm (CAS) [13], particle swarm optimization (PSO) [26], differential evolution (DE) [27], artificial bee colony algorithm [28], hybrid

* Corresponding author

algorithm combing electromagnetism-like algorithm and GA [29], multi-objective optimization algorithms [14, 30-32]. However, this paper focuses on an alternative novel optimization algorithm called real-coded extremal optimization (RCEO) in the attempt to obtain better performance.

Extremal optimization (EO) [33, 34] is a novel meta-heuristics optimization algorithm originally inspired by far-from-equilibrium dynamics of self-organized criticality (SOC) [35, 36]. Unlike traditional evolutionary algorithms, it merely selects against the bad instead of favoring the good randomly or according to a power-law probability distribution, and the mechanism of EO can be characterized from the perspectives of statistical physics, biological co-evolution and ecosystem [37]. The original EO algorithm and its modified versions have been successfully applied to a variety of benchmark and real-world engineering optimization problems, such as graph partitioning [38], graph coloring [39], travelling salesman problem [40, 41], maximum satisfiability (MAX-SAT) problem [42, 43], numerical optimization problems and multi-objective optimization problems [44, 45], community detection in complex network [46], steel production scheduling [47], design of heat pipe [48], and unit commitment problem for power systems [49]. The more comprehensive introduction concerning EO is referred to the surveys [50, 51].

It should be noted that the original EO algorithm and most of modified versions are with individual-based evolutionary mechanism and binary-based mutation operator for combinatorial optimization problems. Nevertheless, there are few reported modified EO algorithms with population-based evolutionary mechanism for continuous optimization problems [44, 59, 60], and these algorithms have never been extended to design FOPID controllers. Mutation operator plays a key role in population-based EO search that generates new solutions [44]. The existing population-based EO algorithms are with random mutation or hybrid Gaussian and Cauchy mutation or polynomial mutation operators. A natural idea is to introduce other mutation operators in real-coded population-based EO algorithms and test whether the performance of the modified algorithms with other mutation operators can be improved. In fact, the effects of different mutation operators on the performance of other reported evolutionary algorithms, e.g., GA, have been studied in a recently reported work [61]. Extensive experimental results have shown that multi-non-uniform mutation (MNUM) performs better than random mutation (RM), non-uniform mutation (NUM), polynomial mutation (PLM), and power mutation (PM) in real-coded GA algorithm for continuous optimization problems. Motivated by these above mentioned research results, we introduce this effective mutation operator called MNUM in population-based EO in this paper. To the best of the authors' knowledge, MNUM is adopted in population-based EO firstly, although it was originally developed for GAs.

On the other hand, the applications of EO to the design of PID controllers are relatively rare [52, 53]. To the best of our knowledge, there is only few reported research work concerning the optimum design of FOPID controllers based on EO. In our recent work [54], a multi-objective individual-based EO algorithm is proposed to design a FOPID controller for an automatic voltage regulator (AVR) system, which is used to maintain the terminal voltage of a synchronous generator at a desired level. This paper presents a novel real-coded population-based EO algorithm with multi-non-uniform mutation called RCEO for the design of FOPID controllers. The basic idea behind the proposed algorithm is the population-based iterated optimization, which consists of generation of a real-coded random initial population by encoding the parameters of FOPID controller into a set of real values, evaluation of the individual fitness by using a more reasonable control performance index, generation of new population based on multi-non-uniform mutation (MNUM) [55], and updating the population by accepting the new population unconditionally. The proposed RCEO algorithm for the design of FOPID controller is relatively simpler than these reported evolutionary algorithms, e.g., GA [13, 56], PSO [13, 56], CAS [13], due to its fewer adjustable parameters and only with selection and mutation operations. Furthermore, a large number of experimental results on some typical benchmark control systems, e.g., AVR system and multivariable control system will demonstrate the superiority of the proposed RCEO-FOPID method to other reported evolutionary algorithms.

The rest of this paper is organized as follows. In Section 2, we give preliminaries concerning FOPID controller, AVR and multivariable fractional-order control system. Section 3 presents the proposed RCEO algorithm for the design of FOPID controller in AVR system. The simulation results on AVR system and multivariable control system are given and discussed in Sections 4 and 5, respectively. Finally, we give the conclusion and open problems in Section 6.

2. Preliminaries

2.1. Fractional order PID controller

As one the most commonly used definitions for fractional differ-integral, Riemann-Liouville (RL) definition is given as the following form [57]:

$$ {}_aD_t^r f(t) = \frac{1}{\Gamma(n-r)} \frac{d^n}{dt^n} \int_a^t \frac{f(\tau)}{(t-\tau)^{r-n+1}} d\tau,\ n-1 < r < n \quad (1) $$

where $\Gamma(.)$ is the Gamma function. The Laplace transform of RL fractional derivative (1) is expressed as follows:

$$\int_0^{\infty} e^{-st}\,{}_0D_t^r f(t)dt = s^r F(s) - \sum_{k=0}^{n-1} s^k\,{}_0D_t^{r-k-1} f(t)\Big|_{t=0} \quad (2)$$

Fig. 1 shows block diagram of a control system with a FOPID controller, which is also called $PI^{\lambda}D^{\mu}$ controller. Its definition in terms of transfer function is given as follows:

Definition 1. The transfer function $G_c(s)$ of a FOPID controller is defined as the following equation [10]:

$$G_c(s) = \frac{U(s)}{E(s)} = K_P + K_I s^{-\lambda} + K_D s^{\mu} \quad (3)$$

where K_P, K_I, and K_D are proportional, integral, and derivative gain, respectively, λ and μ are the fractional order parameter of integrator and differentiator, respectively, and $\lambda>0$, $\mu>0$.

Note that the standard integer order PID controller is one of the special FOPID controller with λ=1 and μ=1.

From the perspective of time domains, the $PI^{\lambda}D^{\mu}$ controller is also expressed in the following form:

$$u(t) = K_P e(t) + K_I D^{-\lambda} e(t) + K_D D^{\mu} e(t)\,. \quad (4)$$

2.2. AVR system

An AVR system [56] consists of four main components including amplifier, exciter, generator, and sensor. More details concerning the transfer functions with the range of parameters modeling these components are shown in Table 1. Here, K_A, K_E, K_G, and K_R are the gains of amplifier, exciter, generator, and sensors, respectively, and τ_A, τ_E, τ_G, and τ_R are inertia time constants of amplifier, exciter, generator, and sensors, respectively. The block diagram of an AVR system with a FOPID controller is given in Fig. 2, where $V_{ref}(s)$ and $V_t(s)$ are the reference voltage and terminal voltage, respectively.

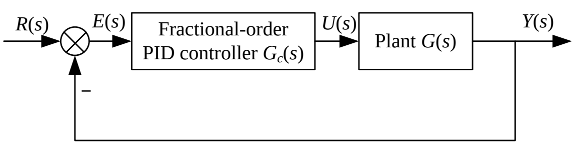


Figure 1. Block diagram of a control system with a FOPID controller

Table 1. Models of the components in an AVR system

Component	Transfer function	Parameters range
Amplifier	$K_A/(1+\tau_A s)$	$10<K_A<400$, $0.02<\tau_A<0.1$
Exciter	$K_E/(1+\tau_E s)$	$1<K_E<400$, $0.5<\tau_E<1$
Generator	$K_G/(1+\tau_G s)$	$0.7<K_G<1$, $1<\tau_G<2$
Sensor	$K_R/(1+\tau_R s)$	$0.001<\tau_R<0.06$

2.3. Multivariable fractional order control system

Fig. 3 shows the block diagram of a multivariable fractional-order control system with multivariable FOPID controller $D(s)$ and a multivariable plant $G(s)$= $\begin{bmatrix} g_{11}(s) & \cdots & g_{1n}(s) \\ \vdots & \ddots & \vdots \\ g_{n1}(s) & \cdots & g_{nn}(s) \end{bmatrix}$.

The corresponding form of $n\times n$ multivariable FOPID controller $D(s)$ is presented as the following equation (5):

$$D(s) = \begin{bmatrix} d_{11}(s) & \cdots & d_{1n}(s) \\ \vdots & \ddots & \vdots \\ d_{n1}(s) & \cdots & d_{nn}(s) \end{bmatrix} \quad (5)$$

where the transfer function of a FOPID sub-controller $d_{ij}(s)$ is characterized as the following equation (6):

$$d_{ij}(s) = K_{Pij} + K_{Iij} s^{-\lambda_{ij}} + K_{Dij} s^{\mu_{ij}}, \forall i, j \in \{1, 2, \cdots, n\} \quad (6)$$

where K_{Pij}, K_{Iij} and K_{Dij} are proportional gain, integral gain, and derivative gain, respectively, and λ_{ij} and μ_{ij} are fractional order parameter of integrator and differentiator, respectively.

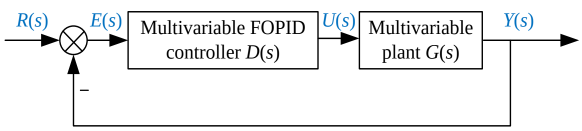


Figure 3. A multivariable fractional-order control system with a multivariable FOPID controller

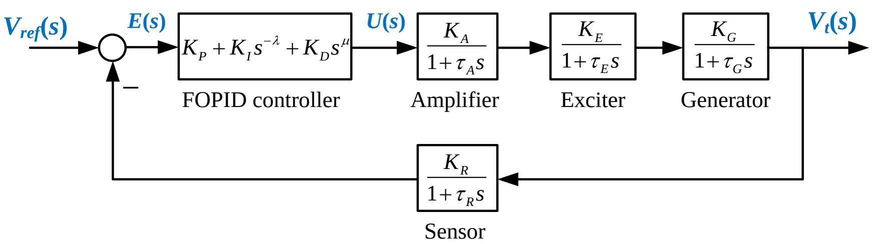


Figure 2. Block diagram of an AVR system with a FOPID controller

3. The proposed algorithms

3.1. Performance criterion for a FOPID controller

In most of the previous research works, the integral absolute error (IAE), or the integral of squared-error (ISE), or the integral of time-squared-error (ITSE) are often used to evaluate the control performance of a control system with a PID or FOPID controller. However, IAE and ISE result in a response with relatively small overshoot but a long settling time, and the derivation process of the analytical ITSE formula is generally complex and time consuming [56, 62]. In this paper, a novel performance criterion is proposed to evaluate a FOPID controller by considering not only IAE, but also the following factors. For example, these indices in the time domain including overshoot M_p, steady-state error E_{ss}, rise time t_r, and settling time t_s should be considered, and the square of the controllers' output, i.e., $\int_0^\infty w_5 u^2(t)dt$ is introduced in order to avoid exporting a large control value. Additionally, $\int_0^\infty w_6 |\Delta y(t)| dt$ is added to avoid a large overshoot value. More accurately, the definition of the proposed performance criterion is presented as follows:

Definition 2. For a real-coded solution *S=[K_P , K_I, K_D, λ, μ]*, which represents a FOPID controller, the corresponding performance criterion *F(S)* in the time domain is defined as equation (7):

$$F(S) = \begin{cases} w_1 M_p + w_2(t_r + t_s) + w_3 E_{ss} + \int_0^\infty (w_4 |e(t)| + w_5 u^2(t))dt, & \text{if } \Delta y(t) \ge 0 \\ w_1 M_p + w_2(t_r + t_s) + w_3 E_{ss} + \int_0^\infty (w_4 |e(t)| + w_5 u^2(t) + w_6 |\Delta y(t)|)dt, & \text{if } \Delta y(t) < 0 \end{cases} \quad (7)$$

where M_p, E_{ss}, t_r, t_s are overshoot, steady-state error, rise time, and settling time, respectively, $e(t)$ is the system error, $\Delta y(t) = y(t) - y(t - T_s)$, T_s is sample time, $u(t)$ is the control output at the time t, w_1, w_2, w_3, w_4, w_5, w_6 are weight coefficients, and $w_6 >> w_4$. For a multivariable fractional-order control system with multivariable FOPID controller, the fitness $F(S)$ is the sum of the fitness of all sub-FOPID controllers.

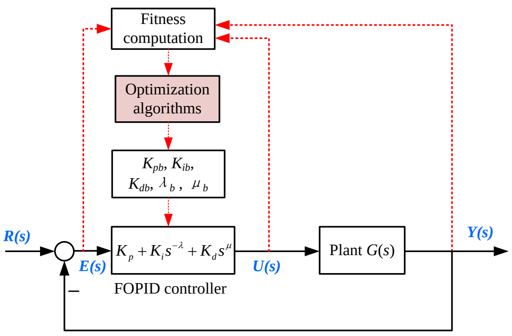


Figure 4. The design framework of optimization algorithms-based FOPID controller

3.2. Performance criterion for a FOPID controller

In general, the basic design framework of optimization algorithm-based FOPID controller is shown in Fig. 4. More specifically, the basic idea behind the proposed RCEO-based FOPID controller design algorithm is the population-based iterated optimization, which consists of generation of a real-coded random initial population by encoding the parameters of FOPID controller into a set of real values, evaluation of individual and population fitness by using a more reasonable control performance index according to definition 2, generation of a new population based on multi-non-uniform mutation (MNUM), and updating the population by accepting the new population unconditionally. Fig. 5 presents the flowchart of the proposed RCEO-based FOPID controller design algorithm. The details of the proposed algorithm are described as follows:

RCEO-based FOPID contrller design algorithm

Input: A control system with a FOPID controller and the ajustable parameters including population size NP, maximum number of iterations $I_{\max}$, and shape parameter b used in MNUM.

Output: the best solution S_{best} and the corresponding best fitness F_{best}.

Step 1. Generate an initial population $\mathbf{P_I}=\{S_1, S_2, \ldots, S_{NP}\}$with the size NP randomly, where each solution $S_i = [K_{Pi}, K_{Ii}, K_{Di}, \lambda_i, \mu_i]$, and set $\mathbf{P} = \mathbf{P_I}$. The detailed process of S_i is $S_i = L + (U - L).R_i$, $i = 1, 2, \ldots, NP$, where L and U are the lower and upper bounds of FOPID control parameters, respectively, and R_i is a set of uniformly distributed random values between 0 and 1.

Step 2. Evaluate the fitness F_i of each solution S_i in population $\mathbf{P}$ according to equation (7), rank all the solutions according to $\{F_i, i=1, 2, \ldots, NP\}$, i.e., find a permutation Π of the labels i such that $F_{\Pi(1)} \le F_{\Pi(2)} \le \ldots \le F_{\Pi(NP)}$, and obtain the best fitness $F_{\text{best}}=\min\{F_i, i=1,2,\ldots,NP\}$ and the corresponding best solution S_{best}.

Step 3. Select the solutions associated with the fitness ranks from $\Pi(1)$ to $\Pi(NP/2)$ to replace those with the ranks from $\Pi(1+NP/2)$ to $\Pi(NP)$, and set the population $\mathbf{P_M}=\{S_{M1}, S_{M2}, \ldots, S_{MNP}\}$, where $S_{Mj} = S_{M(j+NP/2)} = S_{\Pi(j)}$, $j=1, 2, \ldots, NP/2$.

Step 4. Generate a new population $\mathbf{P_N}=\{S_{N1}, S_{N2}, \ldots, S_{NNP}\}$ from $\mathbf{P_M}$ by adopting multi-non-uniform mutation (MNUM) [55]. The detailed process of S_{Ni} is described as the following equations (8) and (9):

$$S_{Ni} = \begin{cases} S_{Mi} + (U - S_{Mi}).A(t), \text{if } r<0.5 \\ S_{Mi} + (S_{Mi} - L).A(t), \text{if } r \geq 0.5, \quad i = 1,...,NP \\ S_{Mi}, \quad \text{otherwise} \end{cases} \quad (8)$$

$$A(t) = \left[r_1 \left(1 - \frac{t}{I_{\max}} \right) \right]^b \quad (9)$$

where t is the current number of iteration, both r and r_1 are uniform random numbers between 0 and 1, and b is the shape parameter used in MNUM.

Step 5. Evaluate the fitness F_{Ni} of each solution S_{Ni} in $\mathbf{P_N}$ according to equation (8) and obtain the best fitness F_{Nb} = min{F_{Ni}, i=1,2,…,NP} in $\mathbf{P_N}$ and the corresponding best solution S_{Nb}.

Step 6. If $F_{\text{best}} \geq F_{Nb}$, then set $S_{\text{best}} = S_{Nb}$ and $F_{\text{best}} = F_{Nb}$; otherwise, keep S_{best} and F_{best} unchanged.

Step 7. Accept $\mathbf{P}=\mathbf{P_N}$ with $S_{NNP}=S_{\text{best}}$ unconditionally.

Step 8. Repeat the steps 2 to 6 until the stopping criterion, e.g., the maximum number of iteration I_{max} is satisfied.

Step 9. Output the best solution $S_{\text{best}}=[K_{Pb}, K_{Ib}, K_{Db}, \lambda_b, \mu_b]$ and the corresponding best fitness F_{best}.

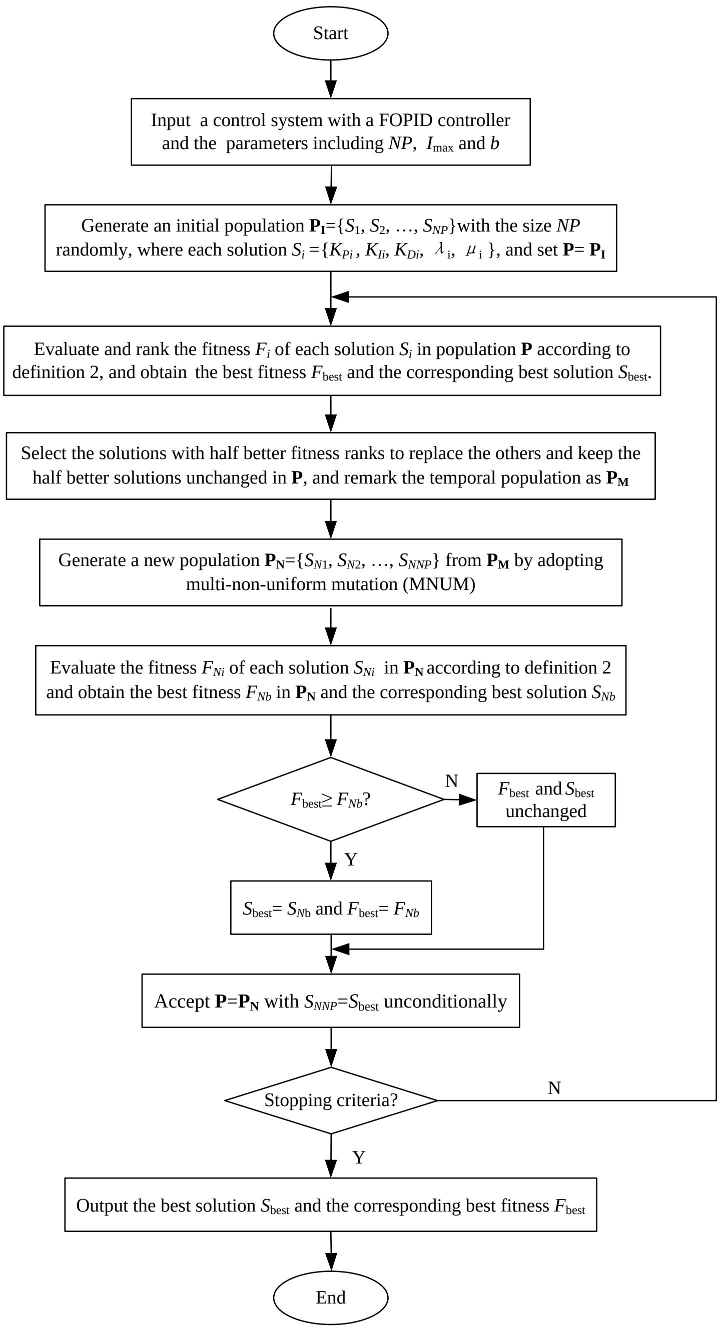


Figure 5. Flowchart of RCEO-based FOPID controller design algorithm

Table 3. The main adjustable parameter settings for different optimization algorithms-based FOPID and PID controllers design methods for AVR system

Algorithm	Main adjustable parameter settings
GA-FOPID [13], GA-PID [56]	NP=50, I_{max}=200, select parameter=0.08, crossover "heuristicXover" with parameter option [2 3], mutation "multiNonUnifMutation" with parameter option [6 genMax 3].
PSO-FOPID [13], PSO-PID [56]	NP=50, I_{max}=200, inertia weight factor ω_{max}=0.9 and ω_{min}=0.4, acceleration parameter c_1=2, c_2=2, the limit of change in velocity $V_{kp}^{max}=K_p^{max}/2$, $V_{ki}^{max}=K_i^{max}/2$, $V_{kd}^{max}=K_d^{max}/2$.
CAS-FOPID, CAS-PID [13]	K=20, I_{max}=300, a=300, b= 2/3, r_i=0.04+0.1*rand, $y_i(0)$=0.9999, ψ_d (d=1, 2, …,5) ≈7.5/ω_d, β=1.0 or 1.5, L=1000.
RCEO-PM-FOPID	NP=30, I_{max}=200, p=8.0+2*gen/I_{max} used in PM.
RCEO-PLM-FOPID	NP=30, I_{max}=200, q=1.0+5*gen/I_{max} used in PLM.
RCEO-NUM-FOPID	NP=30, I_{max}=200, b=2.0+3*gen/I_{max} used in NUM.
RCEO-MNUM-FOPID (RCEO-FOPID)	NP=30, I_{max}=200, b=5.5 used in MNUM.

Table 2. The adjustable parameters used in different optimization algorithms-based FOPID and PID controllers design algorithms

Algorithm	Number of parameters	Adjustable parameters
GA-FOPID [13], GA-PID [56]	5	Population size NP, maximum number of iterations I_{max}, select parameter, crossover rate P_c, mutation rate P_m.
PSO-FOPID [13], PSO-PID [56]	6	NP, I_{max}, inertia weight factors w_{max} and w_{min}, acceleration parameters c_1 and c_2.
CAS-FOPID, CAS-PID [13]	8	Number of ants K, I_{max}, sufficient large positive parameter a, parameter $b\in[0, 2/3]$, organization factor of the ith ant r_i, initial value of the organization variable $y_i(0)$, weighting factor β and large positive real number L used in fitness evaluation.
RCEO-FOPID	3	NP, I_{max}, the parameter b used in MNUM.

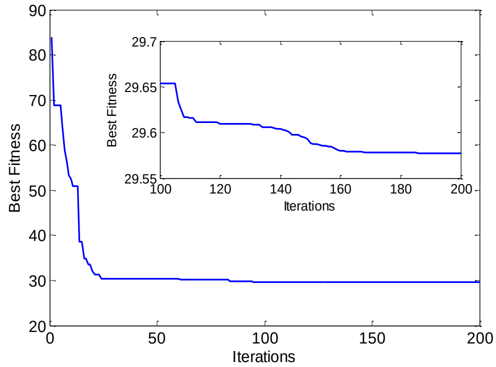
In the above described algorithm, RCEO has only selection and mutation operators, but without crossover operator. The parameters including the size of population (NP), the maximum number of iterations (I_{max}), and the parameter b used in MNUM play critical roles in controlling the performance of RCEO. The effects of these parameters on the performance of RCEO will be analyzed in the next section. The comparison of adjustable parameters used in different optimization algorithms-based FOPID and PID controller design algorithms is shown in Table 2. It is clear that the proposed RCEO is simpler than other reported evolutionary algorithms, e.g., GA [13, 56], PSO [13, 56], and CAS [13], due to not only its fewer control parameters, but also with only selection and mutation operators. Some previous research works (e.g., reference [44]) have compared the computational complexity of population-based EO, GA and PSO. The computational complexity of EO is lower than that of GA and PSO. Furthermore, the superiority of the proposed RCEO-based FOPID controller to these reported evolutionary algorithms-based FOPID and PID controllers in terms of accuracy and robustness will be demonstrated by a large number of experimental results in the next section.

4. Simulation results for AVR system

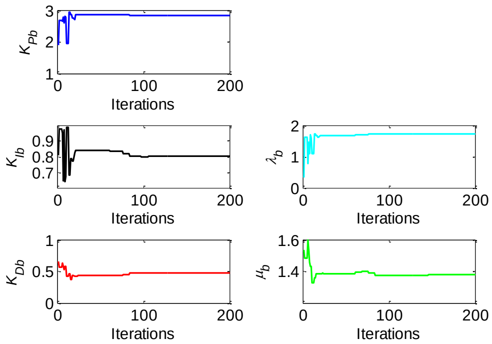
To demonstrate the superiority of RCEO to other reported evolutionary algorithms, such as GA[13, 56], PSO [13, 56], CAS [13] in terms of accuracy and robustness, this section gives the simulation results on AVR system with FOPID or PID controller based on these evolutionary algorithms. For a fair comparison, the parameters of AVR system are set as the same as in the previous research work [13, 56]: K_A = 10, τ_A = 0.1, K_E = 1, τ_E = 0.4, K_G = 1, τ_G = 1, K_S = 1, τ_S = 0.01, K_R = 1, τ_R=0.01. The lower and upper bounds of each FOPID control parameter are set as in [13]: $0\le K_P\le 3$, $0\le K_I\le 1$, $0\le K_D\le 1$, $0\le\lambda\le 2$, $0\le\mu\le 2$ and the sample time T_s is set as 0.01 second. The parameters used in Oustaloup approximation are set as ω_l=0.001ω_c, ω_h=1000ω_c, approximation order N=6, where ω_c represents the gain cross frequency. The weight coefficients are set as follows: w_1=1, w_2=50, w_3=1000, w_4=0.999, w_5=0.001 and w_6=100 by considering the control performance comprehensively based on some experiential rules [53]. In the practical experiments, these weight coefficients are also determined appropriately by trial and error. The main adjustable parameter settings for different optimization algorithms-based FOPID and PID controller design methods in experiments are shown in Table 3. It should be noted that each evolutionary algorithm is executed ten independent runs and all the experiments have been implemented by using MATLAB software based on FOMCON toolbox [58] on a 3.10 GHz PC with processor i5-2400 and 2 GB RAM.

4.1. Comparison with other evolutionary algorithms-based FOPID controllers

To illustrate the good convergence characteristic of the proposed RCEO algorithm for FOPID controller, we present typical optimization process of the best fitness so far and five parameters in FOPID controller based on RCEO algorithm shown in Fig. 6.



(a)



(b)

Figure 6. The optimization process of best fitness (a) and FOPID parameters (b) by RCEO-MNUM for AVR system

In order to demonstrate the effectiveness of MNUM operator in RCEO called RCEO-MNUM-FOPID or RCEO-FOPID for the design of FOPID controller for AVR system, Table 4 presents the comparative simulation results of RCEO with different mutation operators including MNUM, NUM, PLM and PM. The statistical performance of these RCEO algorithms with different mutation operations is evaluated by the best fitness, the average fitness, median fitness, the worst fitness and standard Deviation (SD) obtained by 20 independent runs for each algorithm. It is clear that RCEO-MNUM-FOPID obtains better statistical measures than RCEO-NUM-FOPID, RCEO-PLM-FOPID and RCEO-PM-FOPID.

Table 5 presents the FOPID parameters and the performance corresponding to the median fitness obtained by RCEO with different mutation operators, and the best parameters of FOPID controller and the best performance obtained by CAS with β=1 and β=1.5 [13], GA [13], PSO [13], and MOEO [54]. For the convenience of comparison, the performance of these algorithms is evaluated by the best fitness F_b according to definition 2, overshoot M_p (%), rise time t_r (seconds), settling time t_s with 5% steady-state error (seconds), and steady-state error E_{ss}. The terminal voltage step responses of AVR system with different evolutionary algorithm based FOPID controllers are compared in Fig. 7. Clearly, all performance indices obtained by RCEO-FOPID are better than or at least the same good as those by CAS-FOPID with β=1 and β=1.5, PSO-FOPID, GA-FOPID, RCEO-NUM-FOPID, RCEO-PLM-FOPID, and RCEO-PM-FOPID. In addition, RCEO-FOPID outperforms MOEO-FOPID [54] in terms of all performance indices but t_r.

Table 4. Statistical measures of fitness obtained by RCEO algorithms with different mutation operators for AVR system

Algorithm	Best fitness	Average fitness	Median fitness	Worst fitness	*SD*
RCEO-PM-FOPID	30.0926	32.3765	31.4906	35.3746	1.8909
RCEO-PLM-FOPID	31.4813	35.6219	32.5246	43.9432	4.8517
RCEO-NUM-FOPID	30.0130	33.2225	32.2152	39.2480	3.1642
RCEO-FOPID	**29.3951**	**30.0006**	**29.5776**	**30.6614**	**0.4731**

Table 5. Best FOPID controller parameters and performance obtained by different optimization algorithms

Algorithm	K_{Pb}	K_{Ib}	K_{Db}	λ_b	μ_b	F_b	M_p(%)	t_r(sec.)	t_s(sec.)	E_{ss}
GA-FOPID [13]	1.6947	0.8849	0.3964	1.0248	1.1296	71.5627	9.2600	0.1298	0.3395	0.0006
PSO-FOPID [13]	1.6264	0.2956	0.3226	1.3183	1.1980	78.6795	0.0953	0.1375	0.4563	0.0047
CAS-FOPID(β=1) [13]	1.0537	0.4418	0.2510	1.0624	1.1122	76.0701	0.1678	0.2223	0.3037	0.0014
CAS-FOPID (β=1.5) [13]	0.9315	0.4776	0.2536	1.0275	1.0838	70.6476	0.0642	0.2305	0.3187	0.0012
MOEO-FOPID [54]	2.9737	0.9089	0.5383	1.1446	1.3462	44.2466	3.2038	0.1300	0.1800	6.58E-09
RCEO-PM-FOPID	2.7152	0.7194	0.4045	1.9920	1.4061	31.4906	0.3353	0.1700	0.1700	0
RCEO-PLM-FOPID	2.5970	0.7362	0.3918	1.7641	1.3940	32.5246	0.5992	0.1800	0.1800	5.97E-07
RCEO-NUM-FOPID	2.3867	0.6754	0.3848	1.7651	1.3645	32.2152	0.0759	0.1800	0.1800	0
RCEO-FOPID	2.8316	0.8013	0.4726	1.7294	1.3775	**29.5776**	**0.0644**	0.1400	**0.1400**	**0**

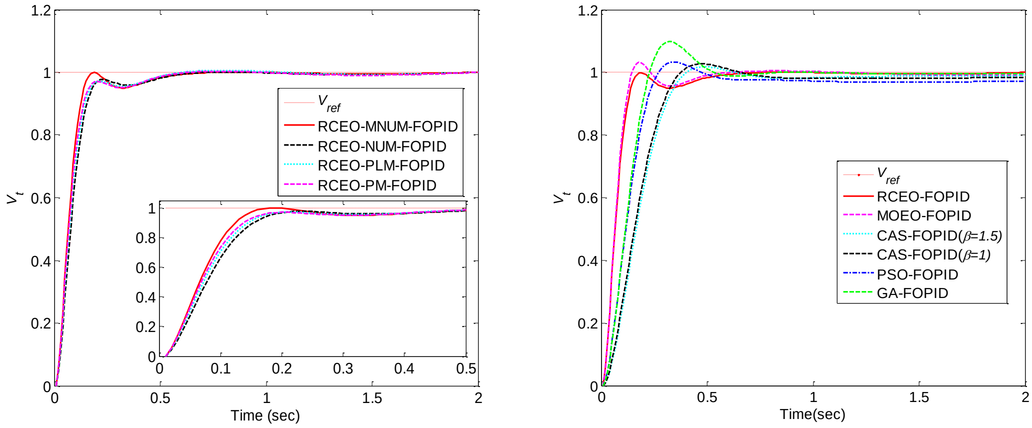


Figure 7. Comparison of terminal voltage step responses of AVR system with different evolutionary algorithm-based FOPID controllers

Table 6. Best controller parameters and comparative performance of RCEO-FOPID, RCEO-PID, CAS-PID with β=1 and β=1.5 [13], PSO-PID [56], and GA-PID [56]

Algorithm	K_{Pb}	K_{Ib}	K_{Db}	λ_b	μ_b	F_b	M_p(%)	t_r(sec.)	t_s(sec.)	E_{ss}
GA-PID [56]	0.8861	0.7984	0.3158	1	1	90.1070	4.54	0.2138	0.8645	0
PSO-PID [56]	0.6254	0.45779	0.2187	1	1	74.1789	1.1592	0.2678	0.3756	1.343E-07
CAS-PID (β=1) [13]	0.6746	0.6009	0.2618	1	1	81.6639	1.7678	0.2425	0.3550	5.630E-08
CAS-PID (β=1.5) [13]	0.6202	0.4531	0.2152	1	1	75.0949	0.4000	0.3156	0.4212	2.688E-08
RCEO-PID	0.7854	0.5451	0.3048	1	1	63.5348	2.5299	0.3100	0.3400	0
RCEO-FOPID	2.8316	0.8013	0.4726	1.7294	1.3775	29.5776	0.0644	0.1400	0.1400	0

4.2. Comparison with other evolutionary algorithms-based PID controllers

On the other hand, to further demonstrate the superiority of RCEO-FOPID controller to other evolutionary algorithms-based PID controllers, we give the experimental results on AVR system with RCEO-FOPID, RCEO-PID, CAS-PID with β=1 and β=1.5 [13], PSO-PID [56], and GA-PID [56]. The best parameters of these above controllers and the corresponding performance are shown in Table 6, and the comparison of the terminal voltage step responses of AVR with RCEO-FOPID controller and other evolutionary algorithms-based PID controllers are presented in Fig. 8. Evidently, the performance of RCEO-FOPID controller is better than that of other evolutionary algorithms-based PID controllers, such as CAS-PID with β=1 and β=1.5 [13], PSO-PID [56], and GA-PID [56]. Even for the same RCEO algorithm, RCEO-FOPID controller can obtain better performance than RCEO-PID controller.

4.3. Robustness test

To illustrate the robustness of RCEO-FOPID controller againest the uncertainties of AVR system parameters, the following experiments considering generator, exciter and amplifier parameters uncertainties due to the changes in load conditions are implemented.

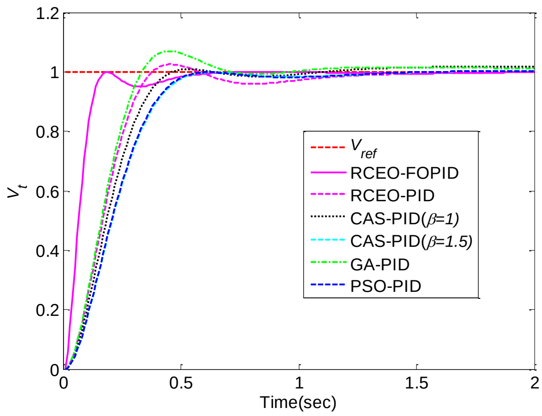


Figure 8. Comparison of the terminal voltage step responses of AVR with RCEO-FOPID controller and other evolutionary algorithms-based PID controllers

4.3.1. Generator uncertainty

Tables 7 and 8 present the comparative performance of RCEO-FOPID controller with other evolutionary algorithms-based FOPID and PID controllers, respectively, when the parameter K_G changes from 1 to 0.8 due to the change in load condition. Additionally, the similar comparative experimental results are shown in Tables 9 and 10 when the parameter τ_G changes from 1 to 1.5, respectively, due to the change in load condition. The corresponding terminal voltage step responses of AVR system are given in Figures 9 and 10. Clearly, the proposed RCEO-FOPID controller is more robust than other evolutionary algorithms-based FOPID controllers, such as CAS-FOPID with $\beta=1$ and $\beta=1.5$ [13], PSO-FOPID [13], and GA-FOPID [13], and also than these evolutionary algorithms-based PID controllers, such as RCEO-PID, CAS-PID with $\beta=1$ and $\beta=1.5$ [13], PSO-PID [56], and GA-PID [56], under the uncertainty of the generator.

Table 7. Comparative performance of different evolutionary algorithms-based FOPID controllers when K_G changes from 1 to 0.8

Algorithm	F_b	M_p(%)	t_r(sec.)	t_s(sec.)	E_{ss}
GA-FOPID [13]	69.1275	6.0830	0.2600	0.4500	0.0045
PSO-FOPID [13]	89.9313	0.0355	0.3100	0.4300	0.0344
CAS-FOPID(β=1) [13]	87.8525	0.1898	0.4200	0.6000	0.0202
CAS-FOPID(β=1.5) [13]	84.0367	1.2419	0.4500	0.6300	0.0130
RCEO-FOPID	**44.8806**	**0.0111**	**0.2000**	**0.4000**	**0.0007**

Table 8. Comparative performance of RCEO-FOPID controller with other evolutionary algorithms-based PID controllers when K_G changes from 1 to 0.8

Algorithm	F_b	M_p(%)	t_r(sec.)	t_s(sec.)	E_{ss}
GA-PID [56]	82.2549	3.0664	0.3800	0.3800	0.0154
PSO-PID [56]	94.6911	0.3057	0.6200	0.6200	0.0031
CAS-PID (β=1) [13]	101.0508	2.0503	0.5200	0.5200	0.0204
CAS-PID (β=1.5) [13]	95.7261	0.2873	0.6300	0.6300	0.0029
RCEO-PID	72.0043	0	0.4400	0.4400	0.0012
RCEO-FOPID	**44.8806**	0.0111	**0.2000**	**0.4000**	**0.0007**

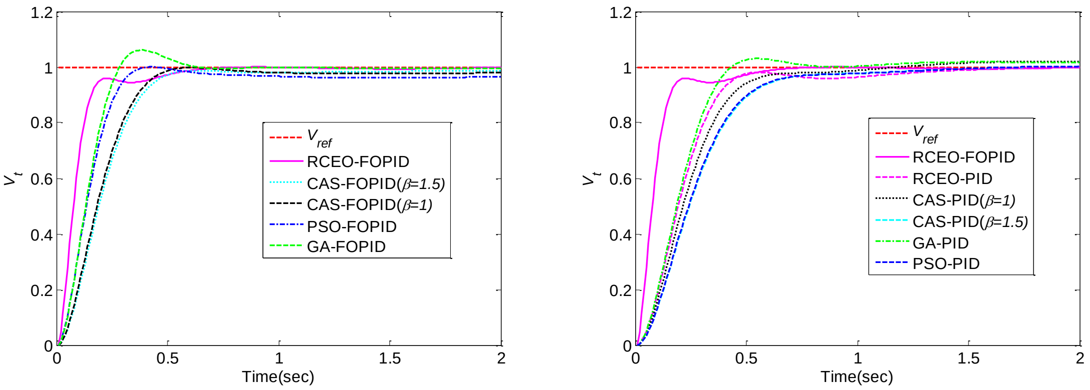


Figure 9. Comparison of the terminal voltage step responses when K_G changes from 1 to 0.8

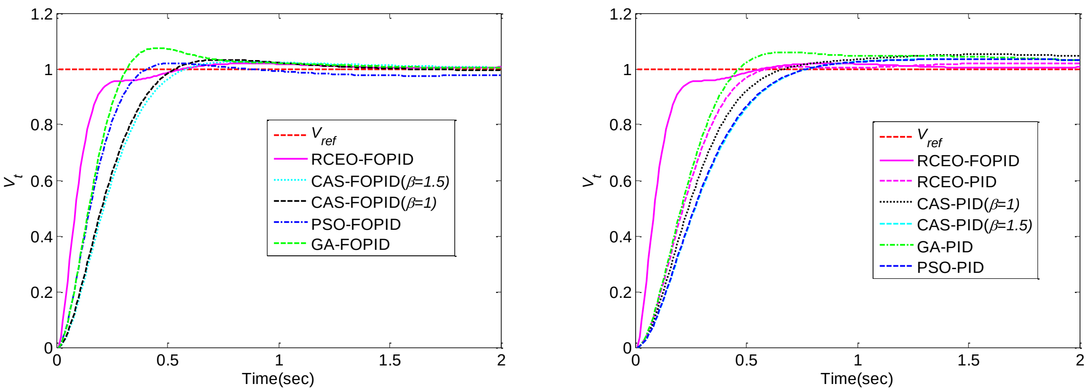


Figure 10. Comparison of the terminal voltage step responses with different evolutionary algorithm-based FOPID or PID controllers when τ_G changes from 1 to 1.5

Table 9. Comparative performance of different evolutionary algorithms-based FOPID controllers when τ_G changes from 1 to 1.5

Algorithm	F_b	M_p(%)	t_r(sec.)	t_s(sec.)	E_{ss}
GA-FOPID [13]	83.2436	7.5394	0.2900	0.6200	0.0032
PSO-FOPID [13]	82.6642	2.0889	0.3400	0.5300	0.0222
CAS-FOPID(β=1) [13]	81.7349	3.3290	0.4600	0.7500	0.0041
CAS-FOPID(β=1.5) [13]	85.1029	2.5890	0.4900	1.1400	0.0055
RCEO-FOPID	**48.4107**	**1.9606**	**0.2500**	**0.2500**	**0.0063**

Table 10. Comparative performance of RCEO-FOPID controller with other evolutionary algorithms-based PID controllers when τ_G changes from 1 to 1.5

Algorithm	F_b	M_p(%)	t_r(sec.)	t_s(sec.)	E_{ss}
GA-PID [56]	135.2635	6.0588	0.4200	0.9100	0.0305
PSO-PID [56]	132.3835	3.5711	0.6300	0.6300	0.0318
CAS-PID (β=1) [13]	201.5966	5.2140	0.5500	1.8100	0.0456
CAS-PID (β=1.5) [13]	132.9880	3.6268	0.6300	0.6300	0.0319
RCEO-PID	96.6990	2.1103	0.4800	0.4800	0.0205
RCEO-FOPID	**48.4107**	**1.9606**	**0.2500**	**0.2500**	**0.0063**

Table 11. Comparative performance of RCEO-FOPID controller with other evolutionary algorithms-based FOPID controllers when K_E changes from 1.0 to 2.0 and τ_E changes from 0.4 to 0.5

Algorithm	F_b	M_p(%)	t_r(sec.)	t_s(sec.)	E_{ss}
GA-FOPID [13]	89.2994	21.9862	0.1500	0.3600	0.0015
PSO-FOPID [13]	84.7413	14.7697	0.1600	0.3500	0.0127
CAS-FOPID(β=1) [13]	89.5037	14.9609	0.2100	0.4800	0.0068
CAS-FOPID(β=1.5) [13]	87.5000	14.6604	0.2200	0.5300	0.0040
RCEO-FOPID	**48.4166**	**10.5594**	**0.0900**	**0.1800**	**0.0011**

Table 12. Comparative performance of RCEO-FOPID controller with other evolutionary algorithms-based PID controllers when K_E changes from 1.0 to 2.0 and τ_E changes from 0.4 to 0.5

Algorithm	F_b	M_p(%)	t_r(sec.)	t_s(sec.)	E_{ss}
GA-PID [56]	106.5724	21.3408	0.2100	0.4900	0.0053
PSO-PID [56]	95.4675	13.7402	0.2700	0.5900	0.0033
CAS-PID (β=1) [13]	99.3885	15.1805	0.2400	0.5300	0.0087
CAS-PID (β=1.5) [13]	95.6920	13.6418	0.2700	0.6500	0.0032
RCEO-PID	94.2162	17.6315	0.2100	0.4800	0.0018
RCEO-FOPID	**48.4166**	**10.5594**	**0.0900**	**0.1800**	**0.0012**

4.3.2. Exciter uncertainty

When the exciter model parameter K_E changes from actual value 1.0 to 2.0 and τ_E changes from actual value 0.4 to 0.5, the comparative performance of RCEO-FOPID controller with other evolutionary algorithms-based FOPID and PID controllers is presented in Tables 11 and 12, respectively, and corresponding terminal voltage step responses of AVR system are shown in Fig. 11. It is clear that the proposed RCEO-FOPID controller is illustrated to be more robust than other evolutionary algorithms-based FOPID and PID controllers in the case of changes of the exciter model parameters.

4.3.3. Amplifier uncertainty

Here, the uncertainty of amplifier model parameters is considered, for example, K_A changes from actual value 10 to 16 and τ_A changes from actual value 0.1 to 0.08. Tables 13 and 14 give the comparative performance of the proposed RCEO-FOPID with other evolutionary algorithms-based

FOPID and PID controllers, respectively, and Fig. 12 presents the corresponding terminal voltage step response of AVR system. It is obvious that the proposed RCEO-FOPID is also more robust than other evolutionary algorithms-based FOPID and PID controllers under the condition of some uncertainty of amplifier model parameters.

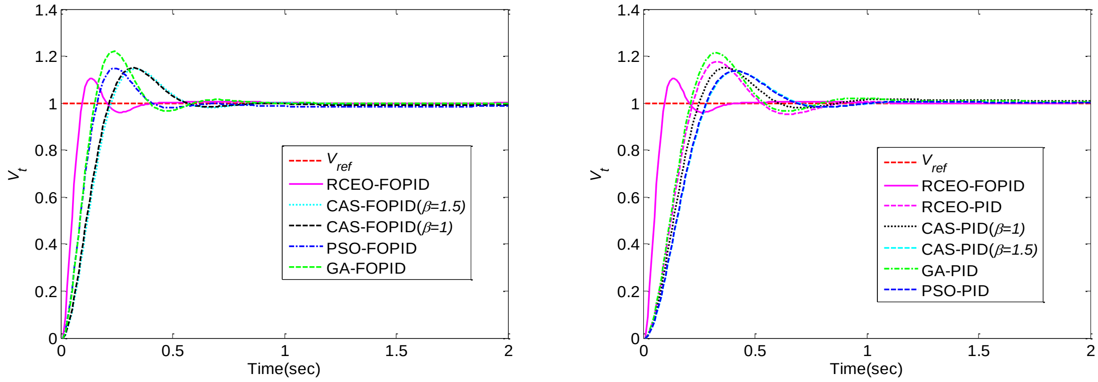


Figure 11. Comparison of the terminal voltage step response with different evolutionary algorithm-based FOPID or PID controllers when K_E changes from 1.0 to 2.0 and τ_E changes from 0.4 to 0.5

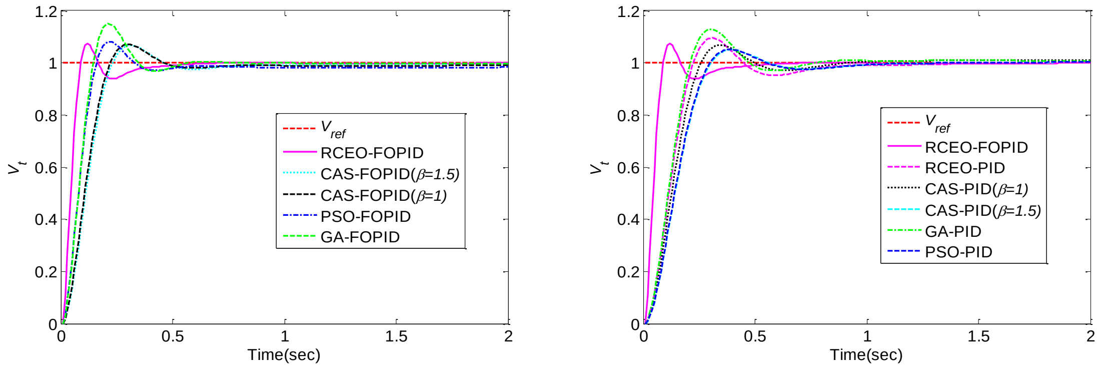


Figure 12. Comparison of the terminal voltage step response with different evolutionary algorithm-based FOPID or PID controllers when K_A changes from 10 to 16 and τ_A changes from 0.1 to 0.08

Table 13. Comparative performance of RCEO-FOPID controller with other evolutionary algorithms-based FOPID controllers when K_A changes from 10 to 16 and τ_A changes from 0.1 to 0.08

Algorithm	F_b	M_p(%)	t_r(sec.)	t_s(sec.)	E_{ss}
GA-FOPID [13]	69.6168	14.9167	0.1400	0.3100	0.0021
PSO-FOPID [13]	71.1200	8.1072	0.1500	0.2800	0.0168
CAS-FOPID(β=1) [13]	69.1841	7.0815	0.2100	0.3700	0.0093
CAS-FOPID(β=1.5) [13]	65.9057	6.7002	0.2200	0.4300	0.0060
RCEO-FOPID	**47.4053**	**7.2511**	**0.0900**	**0.2800**	**0.0004**

Table 14. Comparative performance of RCEO-FOPID controller with other evolutionary algorithms-based PID controllers when K_A changes from 10 to 16 and τ_A changes from 0.1 to 0.08

Algorithm	F_b	M_p(%)	t_r(sec.)	t_s(sec.)	E_{ss}
GA-PID [56]	82.4588	12.9147	0.2000	0.4200	0.0068
PSO-PID [56]	66.1446	5.1452	0.2700	0.4100	0.0025
CAS-PID (β=1) [13]	75.2759	6.8063	0.2400	0.4100	0.0099
CAS-PID (β=1.5) [13]	66.2905	5.0250	0.2800	0.5100	0.0024
RCEO-PID	69.8114	9.4874	0.2100	0.3900	0.0005
RCEO-FOPID	**47.4053**	**7.2511**	**0.0900**	**0.2800**	**0.0004**

4.4. Parameters vs. performance

As aforementioned in Section 3, the adjustable parameters NP, $I_{\max}$, and b used in the proposed RCEO algorithm for the design of FOPID controller play important roles in effecting the performance of RCEO and FOPID controller. This subsection presents the detailed experimental results to illustrate how these parameters affect the performance of the proposed algorithm. It should be noted that the RCEO-FOPID algorithm with each combination of values concerning NP, $I_{\max}$ and b in the following experiment is performed 10 independent runs. More specifically, Fig. 13 presents the variation of the best fitness F_b when the parameter NP varies from 10 to 50, and other parameters keep unchanged, i.e., $I_{\max}$=200 and b=5.5. The best RCEO-FOPID controller parameters and the corresponding performance under different NP values and the same $I_{\max}$=200, b=5.5 are given in Table 15, and the corresponding terminal voltage step responses of AVR are shown in Fig. 14. Clearly, the average value of F_b becomes smaller as the value of NP increases, but the corresponding computational time T_{CPU} also increases. In fact, the variation of the best performance of FOPID controller is relatively small though the value of NP increases. In this sense, the performance of best FOPID controller is robust for the parameter NP when the other two parameters keep unchanged.

Similarly, the effect of parameter $I_{\max}$ on the fitness F_b when NP =50 and b=5.5 is given in Fig. 15. Table 16 presents the best RCEO-FOPID controller parameters and the corresponding performance under different $I_{\max}$ values and the same NP =50 and b=5.5, and Fig. 16 gives the corresponding terminal voltage step responses of AVR. It is evident that the average value of F_b becomes smaller as the value of NP increases, but the corresponding computational time T_{CPU} also increases. Moreover, the performance of best FOPID controller is also robust for the parameter $I_{\max}$ when the other two parameters keep unchanged.

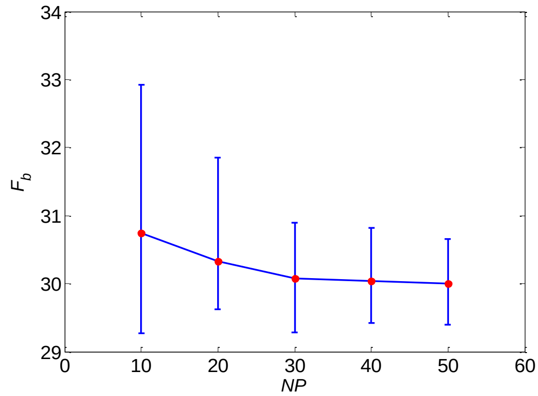


Figure 13. The effect of parameter NP on the fitness F_b when $I_{\max}$=200 and b=5.5.

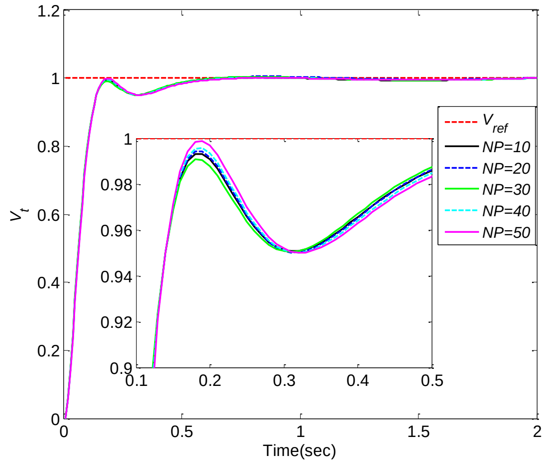


Figure 14. The terminal voltage step responses of AVR system with best RCEO-FOPID controllers as the population size NP varies from 10 to 50 when $I_{\max}$=200 and b=5.5

Table 15. The best RCEO-FOPID controller parameters and the corresponding performance under different NP values and the same values of $I_{\max}$=200 and b=5.5

NP	K_{Pb}	K_{Ib}	K_{Db}	λ_b	μ_b	F_b	M_p(%)	t_r(sec.)	t_s(sec.)	E_{ss}	T_{CPU}(sec.)
10	2.9186	0.7955	0.4626	1.8383	1.3926	29.2760	0.1733	0.1400	0.1400	0	34.500
20	2.8765	0.8582	0.4661	1.6340	1.3884	29.6256	0.4620	0.1400	0.1400	0	69.359
30	2.9482	0.8109	0.4591	1.8153	1.3990	29.2814	0.3265	0.1400	0.1400	0	103.062
40	2.8699	0.8052	0.4677	1.7550	1.3853	29.4282	0.1577	0.1400	0.1400	0	140.735
50	2.8316	0.8013	0.4726	1.7294	1.3775	29.5776	0.0644	0.1400	0.1400	0	172.828

Table 16. The best RCEO-FOPID controller parameters and the corresponding performance under different $I_{\max}$ values and the same values of NP =50 and b=5.5

$I_{\max}$	K_{Pb}	K_{Ib}	K_{Db}	λ_b	μ_b	F_b	M_p(%)	t_r(sec.)	t_s(sec.)	E_{ss}	T_{CPU}(sec.)
50	2.8379	0.7805	0.4322	1.8247	1.3987	29.9050	0.4270	0.1500	0.1500	0	46.782
100	2.8991	0.8673	0.4649	1.6307	1.3885	29.6465	0.5150	0.1400	0.1400	0	86.766
150	2.9294	0.7950	0.4613	1.8533	1.3950	29.2556	0.1915	0.1400	0.1400	0	134.218
200	2.8316	0.8013	0.4726	1.7294	1.3775	29.5776	0.0644	0.1400	0.1400	0	172.828
250	2.9407	0.7833	0.4599	1.9231	1.3976	29.2090	0.1327	0.1400	0.1400	0	216.375
300	2.9166	0.8036	0.4623	1.8075	1.3940	29.2994	0.2322	0.1400	0.1400	0	267.375

Additionally, Fig. 17 illustrates the effect of parameter b on the fitness F_b when I_{max}=200 and NP=50. The best RCEO-FOPID controller parameters and the corresponding performance under different b values and the same I_{max}=200 and NP=50 are given as Table 17, and the corresponding terminal voltage step responses of AVR are shown as Fig. 18. Obviously, the performance of best FOPID controller is also relatively robust for the parameter b when the other two parameters keep unchanged.

Similarly, the effect of parameter I_{max} on the fitness F_b when NP =50 and b=5.5 is given in Fig. 15. Table 16 presents the best RCEO-FOPID controller parameters and the corresponding performance under different I_{max} values and the same NP =50 and b=5.5, and Fig. 16 gives the corresponding terminal voltage step responses of AVR. It is evident that the average value of F_b becomes smaller as the value of NP increases, but the corresponding computational time T_{CPU} also increases. Moreover, the performance of best FOPID controller is also robust for the parameter I_{max} when the other two parameters keep unchanged.

Additionally, Fig. 17 illustrates the effect of parameter b on the fitness F_b when I_{max}=200 and NP=50. The best RCEO-FOPID controller parameters and the corresponding performance under different b values and the same I_{max}=200 and NP=50 are given as Table 17, and the corresponding terminal voltage step responses of AVR are shown as Fig. 18. Obviously, the performance of best FOPID controller is also relatively robust for the parameter b when the other two parameters keep unchanged.

5. Simulation results for AVR system

To demonstrate the superiority of the proposed BCEO-FOPID algorithm to other reported evolutionary algorithms based PID methods, such as adaptive real-coded GA (ARGA-PID) [63], probability binary coded PSO (PBPSO-PID) [64][64], binary-coded EO (BCEO-PID) [53], and RCEO-PID for multivariable control systems, the following binary distillation column plant $G_m(s)$ [65] described by equation (10) with 2-input and 2-output is chosen as a test benchmark:

$$G_m(s)=\begin{bmatrix} \dfrac{12.8e^{-s}}{1+16.7s} & \dfrac{-18.9e^{-3s}}{1+21s} \\ \dfrac{6.6e^{-7s}}{1+10.9s} & \dfrac{-19.4e^{-3s}}{1+14.4s} \end{bmatrix} \tag{10}$$

The steady-state decoupling matrix D_c described by equation (11) of the above multivariable plant model is given as follows:

$$D_c = G_m^{-1}(0)=\begin{bmatrix} 0.1570 & -0.1529 \\ 0.0534 & -0.1036 \end{bmatrix} \tag{11}$$

The lower and upper bounds of each FOPID control parameter are set as $-5\le K_{P1}\le 5$, $-1\le K_{I1}\le 1$, $-1\le K_{D1}\le 1$, $0\le\lambda_1\le 2$, $0\le\mu_1\le 2$, $-5\le K_{P2}\le 5$, $-1\le K_{I2}\le 1$, $-1\le K_{D2}\le 1$, $0\le\lambda_2\le 2$, $0\le\mu_2\le 2$ and the sample time T_s is set as 0.1 min. The parameters used in Oustaloup approximation are set as ω_l=0.01ω_c, ω_h=100ω_c, approximation order N=5, where ω_c represents the gain cross frequency. The weight coefficients are set

Table 17. The best RCEO-FOPID controller parameters and the corresponding performance under different b values and the same values of I_{max}=200 and NP=50

b	K_{Pb}	K_{Ib}	K_{Db}	λ_b	μ_b	F_b	M_p(%)	t_r(sec.)	t_s(sec.)	E_{ss}	T_{CPU}(sec.)
1.0	2.6892	0.7594	0.4484	1.7433	1.3666	30.2086	0.0008	0.1500	0.1500	0	170.812
1.5	2.9896	0.8034	0.4554	1.8910	1.4063	29.2185	0.3461	0.1400	0.1400	0	171.219
2.0	2.8607	0.8164	0.4690	1.7147	1.3837	29.5016	0.2036	0.1400	0.1400	0	170.984
2.5	2.9321	0.7901	0.4610	1.8777	1.3958	29.2380	0.1654	0.1400	0.1400	0	172.640
3.0	2.9082	0.8258	0.4629	1.7328	1.3933	29.4026	0.3487	0.1400	0.1400	0	172.921
3.5	2.8956	0.7906	0.4648	1.8317	1.3896	29.3030	0.1115	0.1400	0.1400	0	170.750
4.0	2.8348	0.8158	0.4719	1.6933	1.3788	29.5967	0.1569	0.1400	0.1400	0	173.109
4.5	2.9016	0.8350	0.4639	1.7030	1.3910	29.4574	0.3771	0.1400	0.1400	0	172.963
5.0	2.8447	0.8337	0.4703	1.6596	1.3815	29.6215	0.2750	0.1400	0.1400	0	170.821
5.5	2.8316	0.8013	0.4726	1.7294	1.3775	29.5776	0.0644	0.1400	0.1400	0	172.828
6.0	2.9403	0.7980	0.4599	1.8546	1.3978	29.2449	0.2356	0.1400	0.1400	0	170.531

Table 18. The main adjustable parameter settings of RCEO-FOPID/PID and other reported algorithms for multivariable control system with decoupler

Algorithm	Main adjustable parameter settings
ARGA-PID [63]	NP=30, I_{max}=200, select parameter=0.08, crossover probability p_c=0.9, mutation probability p_c=0.1−0.01*SZ/NP, SZ=1, 2, …,NP.
PBPSO-PID [64]	NP=40, I_{max}=200, inertia weights w_{max}=0.8, w_{min}=0.8, acceleration factors c_1=2.0, c_2=2.0, V_{max}=50, length of binary code l=16.
BCEO-PID [53]	I_{max}=200, l=10, shape parameter of power law τ=1.30.
RCEO-PID, RCEO-FOPID	NP=30, I_{max}=200, b=5 used in MNUM.

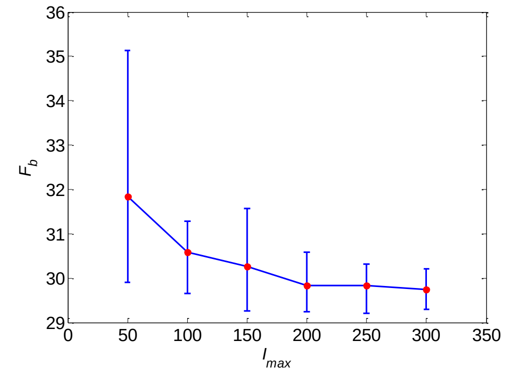


Figure 15. The effect of parameter $I_{\max}$ on the fitness F_b when NP =50 and b=5.5

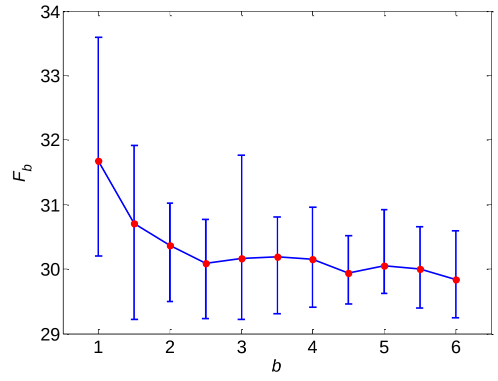


Figure 17. The effect of parameter b on the fitness F_b when $I_{\max}$=200 and NP=50

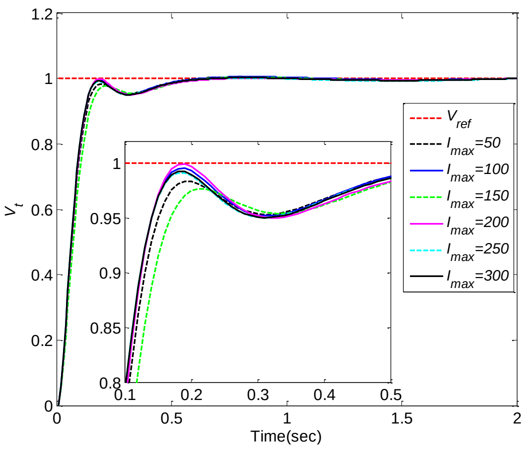


Figure 16. The terminal voltage step response of AVR system with best RCEO-FOPID controllers as $I_{\max}$ varies from 50 to 300 when NP=50 and b=5.5

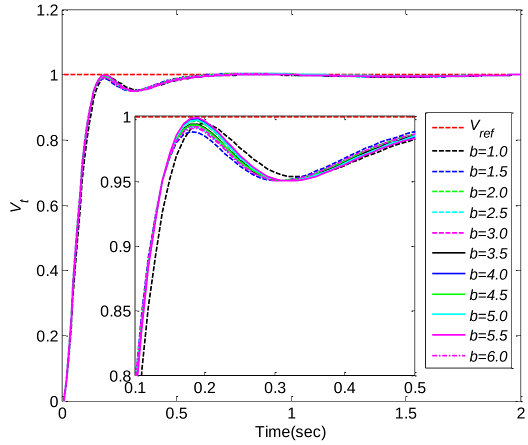


Figure 18. The terminal voltage step response of AVR system with best RCEO-FOPID controllers as b varies from 1.0 to 6.0 when $I_{\max}$=200 and NP=5

as follows: w_1=1, w_2=2, w_3=1000, w_4=1, w_5=0 and w_6=100 by considering the control performance comprehensively based on some experiential rules [53]. The simulation experiments are repeated 20 times for each algorithm. The main adjustable parameter settings of RCEO-FOPID/PID and other reported algorithms for multivariable control system with decoupler are given in Table 18.

The statistical measures of performance including the best fitness, average fitness, the worst fitness, standard Deviation (SD), and success rate (%) obtained by RCEO-FOPID and other reported evolutionary algorithms based PID controllers are shown in Table 19. Clearly, RCEO-FOPID performs better than ARGA-PID [63], PBPSO-PID [64], BCEO-PID [53] and RCEO-PID in terms of all statistical measures. Tables 20 and 21 show the best parameters and the corresponding control performance of multivariable PID/FOPID controller with decoupler obtained by RCEO and other reported algorithms. Fig. 19 presents output y_1 (left) and y_2 (right) under different algorithms-based FOPID/PID controllers with decoupler. It is obvious that RCEO-FOPID obtains better performance indices, including overshoot M_{p1} (%), M_{p2} (%), rise time t_{r1}, t_{r2}, settling time t_{s1}, t_{s2} with 5% steady-state error, and steady-state error E_{ss1}, E_{ss2} than ARGA-PID, BCEO-PID, and RCEO-PID, and it performs better than PBPSO-PID in terms of all indices except M_{p1} (%).

Table 19. Statistical measures of performance obtained by different optimization methods for multivariable control system

Algorithm	Best fitness	Average fitness	Worst fitness	SD	Success rate (%)
ARGA-PID [63]	267.9776	295.0359	335.5359	15.2923	100
PBPSO-PID [64]	435.9980	451.2725	473.8254	12.1056	100
BCEO-PID [53]	273.6514	364.5557	627.4126	112.1602	100
RCEO-PID	262.2600	276.8759	285.6451	7.8989	100
RCEO-FOPID	**226.2376**	**239.3524**	**255.9304**	**7.1862**	**100**

Table 20. Best parameters of multivariable PID/FOPID controllers with decoupler obtained by RCEO and other reported evolutionary algorithms

Algorithm	K_{P1}	K_{I1}	K_{D1}	λ_1	μ_1	K_{P2}	K_{I2}	K_{D2}	λ_2	μ_2
ARGA-PID [63]	2.945	0.159	−0.774	1	1	2.681	0.151	0.250	1	1
PBPSO-PID [64]	1.998	0.112	−0.544	1	1	1.999	0.149	−0.562	1	1
BCEO-PID [53]	2.994	0.159	0.842	1	1	2.877	0.163	0.771	1	1
RCEO-PID	2.987	0.159	−0.596	1	1	2.656	0.141	0.122	1	1
RCEO-FOPID	3.051	0.186	0.702	1.001	0.322	3.033	0.139	0.584	1.067	0.533

Table 21. Comparative best performance of RCEO-FOPID with other reported evolutionary algorithms-based multivariable PID controllers with decoupler

Algorithm	F_B	M_{p1} (%)	t_{r1}	5%t_{s1}	E_{ss1}	M_{p2} (%)	t_{r2}	5%t_{s2}	E_{ss2}
ARGA-PID [63]	267.9776	0.8306	11.4	16.2	4.23E-04	4.6157	10.4	14.8	1.11E-05
PBPSO-PID [64]	435.9980	0.3729	19.0	31.4	5.01E-04	4.9962	14.1	19.5	3.21E-05
BCEO-PID [53]	273.6514	1.3916	12.3	18.3	3.95E-04	4.8063	10.0	16.6	1.84E-05
RCEO-PID	262.2600	1.0421	10.8	13.1	3.96E-04	4.0499	9.4	12.0	1.02E-05
RCEO-FOPID	**226.2376**	0.7200	**9.46**	**11.63**	**1.21E-04**	**3.0747**	**8.52**	**11.36**	**1.09E-06**

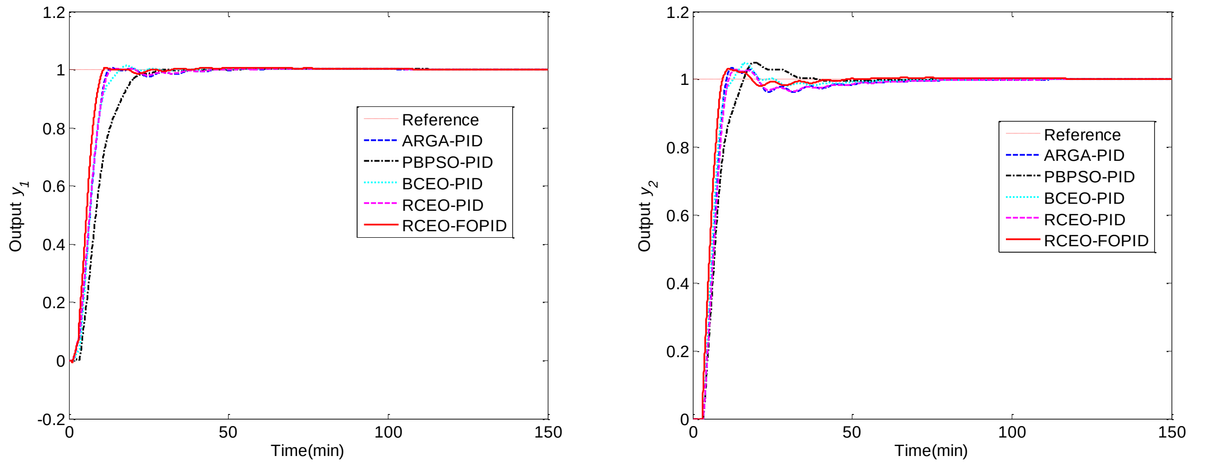


Figure 19. Comparison of output y_1 (left) and y_2 (right) under different algorithms-based FOPID/PID controllers with decoupler

6. Conclusions

In this paper, a novel evolutionary algorithm called RCEO with MNUM mutation operator is proposed for the design of FOPID controller. The key operations of the proposed algorithm includes generation of a real-coded random initial population by encoding the parameters of fractional-order PID controller into a set of real values, evaluation of the individual fitness by using a novel and reasonable control performance index, generation of new population based on MNUM operator, and updating the population by accepting the new population unconditionally. Extensive simulation experimental results on AVR system have demonstrated that the designed RCEO-FOPID controller provides more accurate and robust performance than other reported FOPID and PID controllers based on these evolutionary algorithms, such as GA [13, 56], PSO [13, 56], CAS [13], MOEO [54], and RCEO with other mutation operators, e.g., RCEO-NUM, RCEO-PLM, RCEO-PM. Furthermore, the simulation results on a multivariable control system have also shown that the proposed RCEO-FOPID performs better than these reported evolutionary algorithms, e.g., ARGA-PID [40], PBPSO-PID [9], BCEO-PID [31], and RCEO-PID. However, the performance of the proposed RCEO-FOPID method is further enhanced by choosing more appropriate weight coefficients and adjustable parameters. Additionally, the basic idea of the proposed RCEO algorithm will be extended to design other FOPID controllers in more complex industrial control problems.

Acknowledgments

The authors gratefully acknowledge the helpful comments and suggestions of editors and anonymous reviewers. This work was partially supported by the National Natural Science Foundation of China (No. 51207112), Zhejiang Province Science and Technology Planning Project (Nos. 2014C31074, 2014C31093, 2015C31157), Zhejiang Provincial Natural Science Foundation of China (Nos. LY16F030011, LZ16E050002, LQ14F030006, LQ14F030007), and the Program of Xinmiao (Potential) Talents in Zhejiang Province (No. 2014R424014).

References

[1] **H. Lin, H. Y. Su, Z. Shu, Z. G. Wu, Y. Xu.** Optimal estimation in UDP-like networked control systems with intermittent inputs: stability analysis and suboptimal filter design. *IEEE Transactions on Automatic Control*, 2016, Vol. 61, No. 7, 1794-1809.

[2] **Y. Q. Wu, H. Y. Su, R. Q. Lu, Z. G. Wu, Z. Shu**. Passivity-based non-fragile control for Markovian jump systems with aperiodic sampling. *Systems & Control Letters*, 2015, Vol. 84, 35-43.

[3] **Z. G.Wu, P. Shi , H. Y. Su, J. Chu**. Asynchronous l_2-l_∞ filtering for discrete-time stochastic Markov jump systems with randomly occurred sensor nonlinearities. *Automatica*, 2014, Vol. 50, No. 1, 180-186.

[4] **Y. Pan, H. Yu, M. J. Er.** Adaptive neural PD control with semiglobal asymptotic stabilization guarantee. *IEEE Transactions on Neural Networks and Learning Systems*, 2014, Vol. 25, No. 12, 2264-2274.

[5] **Y. Pan**, **J. E. Meng**. Enhanced adaptive fuzzy control with optimal approximation error convergence. *IEEE Transactions on Fuzzy Systems*, 2013, Vol. 21, No. 6, 1123-1132.

[6] **Y. Pan**, **J. E. Meng**, **D. P. Huang**, **Q. R. Wang**. Adaptive fuzzy control with guaranteed convergence of optimal approximation error. *IEEE Transactions on Fuzzy Systems*, 2011, Vol. 19, No. 5, 807-818.

[7] **Y. Pan**, **H. Yu**. Dynamic surface control via singular perturbation analysis. Automatica, 2015, Vol. 57, 29-33.

[8] **K. H. Ang, G. Chong, Y. Li**. PID control system analysis, design and technology. *IEEE Transactions on Control Systems Technology*, 2005, Vol. 13, No. 4, 559-576.

[9] **K. J. Åström, T. Hägglund**. The future of PID control. *Control Engineering Practice*, 2001, Vol. 9, No. 11, 1163-1175.

[10] **I. Podlubny**. Fractional-order systems and $PI^\lambda D^\mu$ controllers. *IEEE Transactions on Automatic Control*, 1999, Vol. 44, No. 1, 208-213.

[11] **Y. Q. Chen, I. Petráš, D.Y. Xue**. Fractional order control-A tutorial. In: *Procedings of 2009 American Control Conference*, 2009, pp.1397-1411.

[12] **C. A. Monje, Y. Q. Chen, B. M. Vinagre, D. Y. Xue, V. Feliu**. Fractional-order systems and controls: Fundamentals and Applications. *Springer*, 2010.

[13] **Y. G. Tang, M. Y. Cui, C. C. Hua, L. X. Li, Y. X. Yang**. Optimum design of fractional order $PI^\lambda D^\mu$ controller for AVR system using chaotic ant swarm. *Expert Systems with Applications*, 2012, Vol. 39, 6887-6896.

[14] **K. Tehrani, A. Amirahmadi, S. Rafiei, G. Griva, L. Barrandon, M. Hamzaoui, I. Rasoanarivo, F. Sargos**. Design of fractional order PID controller for boost converter based on multi-objective optimization. In: *Proceedings of 14th International power electronics and motion control conference (EPE/PEMC)*, 2010, pp.179-185.

[15] **H. S. Li, Y. Luo, Y. Q. Chen**. A Fractional Order Proportional and Derivative (FOPD) Motion Controller: Tuning Rule and Experiments. *IEEE Transactions on Control Systems Technology*, 2010, Vol. 18, No. 2, 516-520.

[16] **Y. Q. Chen**, **T. Bhaskaran**, **D. Y. Xue**. Practical tuning rule development for fractional order proportional integral controllers. *ASME Journal of Computational and Nonlinear Dynamics*, 2008, Vol. 3, No.2, 021403.

[17] **A. J. Calderón, B. M. Vinagre, V. Feliu**. Fractional Order Control Strategies for Power Electronic Buck Converters. *Signal Processing*, 2006, Vol. 86, No. 10, 2803-2819.

[18] **C. A. Monje, B. M. Vinagre, V. Feliu, Y. Q**. **Chen**. Tuning and auto-tuning of fractional order controllers for industry applications. *Control Engineering Practice*, 2008, Vol. 16, No. 7, 798-812.

[19] **J. Cervera, A. Banos, C. Monje, B. Vinagre**. Tuning of fractional PID controllers by using QFT. In: *Proceedings of 32nd annual conference on IEEE industrial electronics*, 2006, pp. 5402-5407.

[20] **S. E. Hamamci.** An algorithm for stabilization of fractional-order time delay systems using fractional-order PID controllers. *IEEE Transactions on Automatic Control*, 2007, Vol. 52, No. 10, 1964-1969.

[21] **M. K. Bouafoura, N. B. Braiek**. $PI^\lambda D^\mu$ controller design for integer and fractional plants using piecewise orthogonal functions. *Communications in Nonlinear Science and Numerical Simulation*, 2010, Vol. 15, No. 5, 1267-1278.

[22] **F. Padula, A.Visioli**. Tuning rules for optimal PID and fractional-order PID controllers. *Journal of Process Control*, 2011, Vol. 21, No. 1, 69-81.

[23] **M. S. Tavazoei.** Overshoot in the step response of fractional-order control systems. *Journal of Process Control*, 2012, Vol. 22, No. 1, 90-94.

[24] **Z. Gao, M. Yan, J. X.Wei**. Robust stabilizing regions of fractional-order PD^μ controllers of time-delay fractional-order systems. *Journal of Process Control*, 2014, Vol. 24, No. 1, 37-47.

[25] **R. El-Khazali**. Fractional-order $PI^\lambda D^\mu$ controller design. *Computers and Mathematical with Applications*, 2013, Vol. 66, 639-646.

[26] **M. Zamani, M. K. Ghartemani, N. Sadati, M. Parniani**. Design of a fractional order PID controller for an AVR using particle swarm optimization. *Control Engineering Practice*, 2009, Vol. 17, No. 12, 1380-1387.

[27] **A. Biswas, S. Das, A. Abraham, S. Dasgupta**. Design of fractional-order $PI^\lambda D^\mu$ controllers with an improved differential evolution. *Engineering Applications of Artificial Intelligence*, 2009, Vol. 22, No. 2, 343-350.

[28] **H. Gozde, M. C. Taplamacioglu**. Comparative performance analysis of artificial bee colony algorithm for automatic voltage regulator (AVR) system. *Journal of Franklin Institute*, 2011, Vol. 348, No. 8, 1927-1946.

[29] **C. H. Lee, F. K. Chang**. Fractional-order PID controller optimization via improved electromagnetism-like algorithm. *Expert Systems with Applications*, 2010, Vol. 37, No. 12, 8871-8878.

[30] **L. Meng, D. Y. Xue**. Design of an optimal fractional-order PID controller using multi-objective GA optimization. In: *Proceedings of 21st Chinese Control and Decision Conference*, 2009, pp. 3849-3853.

[31] **I. Pan, S. Das**. Chaotic multi-objective optimization based design of fractional order $PI^{\lambda}D^{\mu}$ controller in AVR system. *International Journal of Electrical Power & Energy Systems*, 2012, Vol. 43, No. 1, 393-407.

[32] **I. Pan, S. Das**. Frequency domain design of fractional order PID controller for AVR system using chaotic multi-objective optimization. *International Journal of Electrical Power & Energy Systems*, 2013, Vol. 51, 106-118.

[33] **S. Boettcher, A. G. Percus**. Nature's way of optimizing. *Artificial Intelligence*, 2000, Vol. 119, Issue 1-2, 275-286.

[34] **S. Boettcher, A. G. Percus**. Optimization with extremal dynamics. *Physical Review Letters*, 2001, Vol. 86, No. 23, 5211-5214.

[35] **P. Bak, C. Tang, K. Wiesenfeld**. Self-organized criticality: An explanation of 1/*f* noise. *Physical Review Letters*, 1987, Vol. 59, 381-384.

[36] **P. Bak, K. Sneppen**. Punctuated equilibrium and criticality in a simple model of evolution. *Physical Review Letters*, 1993, Vol. 71, No. 24, 4083-4086.

[37] **Y. Z. Lu, M. R. Chen, Y. W. Chen**. Studies on extremal optimization and its applications in solving real world optimization problems. In: *Proceedings of the 2007 IEEE Symposium on Foundations of Computational Intelligence, Hawaii, USA*, 2007, pp.162-168.

[38] **S. Boettcher, A. G. Percus**. Extremal optimization for graph partitioning. *Physical Review E*, 2001, Vol. 64, No. 2, 026114.

[39] **S. Boettcher, A. G. Percus**. Extremal optimization at the phase transition for the three-coloring problem. *Physical Review E*, 2004, Vol. 69, No. 6, 066703.

[40] **Y. W. Chen, Y. Z. Lu, P. Chen**. Optimization with extremal dynamics for the traveling salesman problem. *Physica A*, 2007, Vol. 385, No. 1, 115-123.

[41] **J. M. Liu, Y. W. Chen, G. K. Yang, Y. Z. Lu**. Self-organized combinatorial optimization. *Expert Systems with Applications*, 2011, Vol. 38, No. 8, 10532-10540.

[42] **M. B. Menaï, M. Batouche**. An effective heuristic algorithm for the maximum satisfiability problem. *Applied Intelligence*, 2006, Vol. 24, No. 3, 227-239.

[43] **G. Q. Zeng, Y. Z. Lu, W. J. Mao**. Modified extremal optimization for the maximum satisfiability problem. *J. Zhejiang Univ-Sci C (Comput & Electron)*, 2011, Vol. 12, No. 7, 589-596.

[44] **M. R. Chen, X. Li, X. Zhang, Y. Z. Lu**. A novel particle swarm optimizer hybridized with extremal optimization. *Applied Soft Computing*, 2010, Vol. 10, No. 2, 367-373.

[45] **M. R. Chen, Y. Z. Lu**. A novel elitist multiobjective optimization algorithm: multiobjective extremal optimization. *European Journal of Operational Research*, 2008, Vol. 188, No. 3, 637-651.

[46] **J. Duch, A. Arenas**. Community detection in complex networks using extremal optimization. *Physical Review E*, 2005, Vol. 72, No. 2, 027104.

[47] **Y. W. Chen, Y. Z. Lu, G. K. Yang**. Hybrid evolutionary algorithm with marriage of genetic algorithm and extremal optimization for production scheduling. *International Journal of Advanced Manufacturing Technology*, 2008, Vol. 36, No. 9-10, 959-968.

[48] **F. L. Sousa, V. Vlassov, F. M. Ramos**. Heat pipe design through generalized extremal optimization. *Heat Transfer Engineering*, 2004, Vol. 25, No. 7, 34-45.

[49] **J. Ding, Y. Z. Lu, J. Chu.** Extremal optimization for unit commitment problem for power systems. In: *Proceedings of IEEE Power and Energy Society General Meeting*, 2012, pp.1-8.

[50] **S. Boettcher**. Evolutionary dynamics of extremal optimization. In: *3rd Conference on Learning and Intelligent Optimization Location*, Lecture Notes on Computer Science, 2009, Vol. 5851, pp. 1-14.

[51] **G. Q. Zeng, Y. Z. Lu**. Survey on computational complexity with phase transitions and extremal optimization. In: *Proceedings 48th IEEE conference Control and Decision&28th Chinese Control Conference*, 2009, pp. 4352-4359.

[52] **H. B. Huo, X. J. Zhu, G. Y. Cao.** Design for two-degree-of-freedom PID regulator based on improved generalized extremal optimization algorithm. *Journal of Shanghai Jiaotong University (Science)*, 2007, E-12, No. 2, 148-153.

[53] **G. Q. Zeng**, **K. D. Lu, Y. X. Dai, Z. J. Zhang, M. R. Chen, C. W. Zheng, D. Wu, W. W. Peng**. Binary-coded extremal optimization for the design of PID controllers. *Neurcomputing*, 2014, Vol. 138, 180-188.

[54] **G. Q. Zeng, J. Chen, Y. X. Dai, L. M. Li, C. W. Zheng, M. R. Chen**. Design of fractional order PID controller for automatic voltage regulator system based on multi-objective extremal optimization. *Neurocomputing*, 2015, Vol. 160,173-184.

[55] **C. R. Houck, J. A. Joines, M. G. Kay**. A Genetic Algorithm for Function Optimization: A Matlab Implementation. *Technical Report, North Carolina State University, Raleigh, NC*, 1996.

[56] **Z. L. Gaing**. A particle swarm optimization approach for optimum design of PID controller in AVR System. *IEEE Transactions Energy Conversion*, 2004, Vol. 19, No. 2, 384-391.

[57] **I. Podlubny**. Fractional Differential Equations. *Academic Press, San Diego*, 1999.

[58] **A. Tepljakov, E. Petlenkov, J. Belikov**. FOMCON: Fractional-Order Modeling and Control Toolbox for MATLAB. In: *Proceedings of 18th International Conference Mixed Design of Integrated Circuits and Systems*, 2011, pp. 684-689.

[59] **L. M. Li, K. D. Lu, G. Q. Zeng, L. Wu, 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*, 2016, Vol. 174, 577-587.

[60] **G. Q. Zeng, K. D. Lu, J. Chen, Z. J. Zhang, Y. X. Dai, W. W. Peng, C. W. Zheng**. An improved real-coded population-based extremal optimization method for continuous unconstrained optimization problems. *Mathematical Problems in Engineering*, 2014, Vol. 2014, Article ID 420652, 9 pages.

[61] **P. H. Tang, M. H. Tseng**. Adaptive directed mutation for real-coded genetic algorithms. *Applied Soft Computing*, 2013, Vol 13, No. 1, 600-614.

[62] **R. A. Krohling, H. Jaschek, J. P. Rey.** Designing PI/PID controller for a motion control system based on genetic algorithm. In: *Proceedings of 12th IEEE International Symposium on Intelligent Control*, Istanbul, Turkey, July 1997, pp. 125-130.

[63] **J. Zhang, S. H. H. Chung, W. L. Lo.** Clustering-based adaptive crossover and mutation probabilities for genetic algorithms. *IEEE Transactions on Evolutionary Computation*, 2007, Vol. 11, No. 3, 326-335.

[64] **M. I. Menhas, L. Wang, M. Fei, H. Pan**. Comparative performance analysis of various binary coded PSO algorithms in multivariable PID controller design. *Expert Systems with Applications*, 2012, Vol. 39, No. 4, 4390-4401.

[65] **M. W. Iruthayarajan, S. Baskar.** Covariance matrix adaptation evolution strategy based design of centralized PID controller. *Expert Systems with Applications*, 2010, Vol. 37, No. 8, 5775-5781.

Received October 2015.