Expert Systems with Applications 39 (2012) 6887–6896

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Optimum design of fractional order $PI^{\lambda}D^{\mu}$ controller for AVR system using chaotic ant swarm

Yinggan Tang [a,c,*], Mingyong Cui [a], Changchun Hua [a], Lixiang Li [b,d], Yixian Yang [b,d]

[a] Key Lab of Industrial Computer Control Engineering of Hebei Province, Yanshan University, Qinhuangdao, Hebei 066004, China
[b] Research Center on Fictitious Economy and Data Science, Chinese Academy of Sciences, Beijing 100190, China
[c] National Engineering Research Center for Equipment and Technology of Cold Strip Rolling, Qinhuangdao, Hebei 066004, China
[d] Information Security Center, Beijing University of Posts and Telecommunications, Beijing 100876, China

ARTICLE INFO

Keywords:
Fractional PID controller
Optimal control
Chaotic ant swarm
AVR system

ABSTRACT

Fractional-order PID (FOPID) controller is a generalization of standard PID controller using fractional calculus. Compared to PID controller, the tuning of FOPID is more complex and remains a challenge problem. This paper focuses on the design of FOPID controller using chaotic ant swarm (CAS) optimization method. The tuning of FOPID controller is formulated as a nonlinear optimization problem, in which the objective function is composed of overshoot, steady-state error, raising time and settling time. CAS algorithm, a newly developed evolutionary algorithm inspired by the chaotic behavior of individual ant and the self-organization of ant swarm, is used as the optimizer to search the best parameters of FOPID controller. The designed CAS-FOPID controller is applied to an automatic regulator voltage (AVR) system. Numerous numerical simulations and comparisons with other FOPID/PID controllers show that the CAS-FOPID controller can not only ensure good control performance with respect to reference input but also improve the system robustness with respect to model uncertainties.

1. Introduction

In the past decades, modern control theories have made great advances. Control techniques including optimal control, H_∞/H_2 control, fuzzy control, neural network control, predictive control, and so on, have been developed significantly. Nevertheless, the proportional-integral-derivative (PID) control technique has still been widely utilized in many industrial applications such as process control, motor drives, flight control, etc. Nowadays, more than 90% control loops in industry are PID control. This is mainly due to the fact that PID controller possesses robust performance to meet the global change of industry process, simple structure to be easily understood by engineers, and easiness to design and implement.

Recently, there are increasing interests to enhance the performance of PID controller by using the concept of fractional calculus, where the orders of derivatives and integrals are non-integer. Fractional calculus is non-local, which makes it able to emphasize mathematically the long-memory. Fractional order PID controller (FOPID or $PI^{\lambda}D^{\mu}$, where λ and μ are the integrating and derivative orders and they are non-integers) proposed by Podlubny (1999) is a generalization of the PID controller using fractional calculus. A FOPID controller is characterize by five parameters, i.e., the proportional gain, the integrating gain, the derivative gain, the integrating order and the derivative order. Over the last years, FOPID controllers find many applications in irrigation canal control (Domingues, Valerio, & da Costa, 2009), temperature tracking (Ahn, Bhambhani, & Chen, 2009), motion control of DC motor (Petráš, 2009; Xue, Zhao, & Chen, 2006), boost converter control (Tehrani et al., 2010), hypersonic flight vehicle control (Changmao, Naiming, & Zhiguo, 2010), servo press control system (Fan, Sun, & Zhang, 2007). The above research results show that FOPID controller has better performance and robustness than conventional PID controller. Though it is so, the parameter tuning of FOPID controller is an important and critical issue. Compare to conventional PID controller, FOPID controller has two extra parameters. On one hand, it enables people have more degrees of freedom to design FOPID controller, and, on the other hand, it means that it is more complex in the synthesis of FOPID controller. In the literature, many approaches have been documented to design FOPID controller. These approaches can be classified into two classes: analytic methods and heuristic methods. In the analytic context, the parameters of FOPID controller are tuned by minimizing a nonlinear objective function depending on the specifications imposed by the designers. In Cervera, Banos, Monje, and Vinagre (2006), tuning of FOPID controller is recasted as a quantitative feedback theory (QFT) loop shaping problem, where the optimization objective is the high frequency gain of the nominal loop subjected to the

* Corresponding author at: The Westward direction of Hebei street 438, Yanshan University, Qinhuangdao, Hebei 066004, China. Fax: +86 335 8072979.
E-mail address: ygtang@ysu.edu.cn (Y. Tang).

0957-4174/$ - see front matter Crown Copyright © 2012 Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2012.01.007

restrictions given by the specifications. Bouafoura and Braiek (2010) proposed a new analytic method to design FOPID controller by expanding the control loop signal and reference model input and output over a piecewise orthogonal functions. In such a manner, the fractional differential calculus are replaced by the generalized operational matrices of differentiation related to these bases, and thus, the controller tuning is elaborated simply with a matrix manipulator manner. However, the tuning of orders of integrator and derivation are not considered in Bouafoura and Braiek (2010). Monje et al. (2004), Monje, Vinagre, Feliu, and Chen (2008) proposed to tuning parameters of FOPID controller by taking into account five conditions about phase and gain margins specifications and constrains over the sensitivity functions. As far as the heuristic methods, rule-based methods and evolutionary algorithm based methods were explored by several authors. In Padula and Visioli (2011), a set of tuning rules were devised based on a first order plus time model of the process by minimizing the integrated absolute error with a constraint on the maximum sensitivity. Ziegler–Nichols like tuning rules for FOPID controller were given in Valéio and da Costa (2006). Evolutionary algorithms including genetic algorithm (GA), particle swarm optimization (PSO), electromagnetism-like algorithm (EM), differential evolution are also used to design FOPID controller. Genetic algorithms were adopted by Cao and Meng and Xue (2009) to design FOPID controller by recasting the problem to an optimization problem. In Zamani, Karimi-Ghartemani, Sadati, and Parniani (2009), particle swarm optimization was used to design FOPID controller for an AVR system via minimizing an objective function consisting of overshoot, rising time, settling time, steady-state error, the integral of absolute error (IAE), integral of squared input, gain margin and phase margin. An improved electromagnetism-like algorithm with genetic algorithm (IEMGA) technique was proposed by Lee and Chang (2010) for FOPID controller design through minimizing the integrated-square-error (ISE). Biswas, Das, Abraham, and Dasgupta (2009) proposed to design FOPID based on the root locus method using improved differential evolution.

Chaotic ant swarm (CAS) is a new evolutionary algorithm developed by Li, Peng, Wang, and Yang (2006a). CAS algorithm searches the optimum solution for a concrete optimization problem by utilizing the self-organization behavior of ant swarm and the chaotic behavior of individual ant. CAS integrates the advantages of chaotic search and the powerful ability of swarm collectiveness. It shows great potential in solving difficult optimization problems encountered in scientific and engineering fields such as parameter identification of dynamic systems (Li, Yang, Peng, & Wang, 2006b; Peng, Li, Yang, & Liu, 2010; Tang, Cui, Li, Peng, & Guan, 2009), fuzzy system identification (Li, Yang, & Peng, 2009), economic dispatch problems (Cai et al., 2007, Cai, Ma, Li, Li, & Peng, 2010), electric load forecasting (Hong, 2010), TSP problem (Wei, Ge, Lu, Li, & Yang, 2010), parameter tuning of PID controller (Zhu, Li, Zhao, Guo, & Yang, 2009) and clustering (Wan et al., 2010).

In this paper, we pay our attention to FOPID controller design using CAS algorithm for an AVR system. The parameters of FOPID controller are determined through optimizing a nonlinear function consisting of overshoot, steady-state error, raising time and settling time. These measures are calculated from the step response of AVR system. CAS algorithm is used as optimizer to search the optimal parameters of FOPID. The controller is called CAS-FOPID controller. The designed CAS-FOPID controller is applied to an automatic voltage regulator (AVR) system. The rest of the paper is organized as follows. In Section 2, the basic of fractional calculus is briefly reviewed. Chaotic ant swarm is introduced in Section 3. The details of FOPID controller design using CAS for AVR system are given in Section 4. Results obtained from simulation and observations are described in Section 5. Finally, conclusion remarks are given in Section 6.

2. The basic of fractional calculus

Fractional calculus is a generalization of the integration and differentiation to the non-integer order fundamental operator ${}_aD_t^\alpha$, where a and t are limits and $\alpha(\alpha \in R)$ is the order of the operation. There are several definitions of fractional integration and differentiation. Among these definitions, the Grünwald–Letnikor (GL) and the Riemann–Liouville (RL) definitions are commonly used. The GL definition is

$$ {}_aD_t^\alpha f(t) = \lim_{h\to 0} h^{-\alpha} \sum_{j=0}^{[(t-a)/h]} (-1)^j \binom{\alpha}{j} f(t-jh), \tag{1}$$

where $[\cdot]$ means the integer part. The RL definition is

$$ {}_aD_t^\alpha f(t) = \frac{1}{\Gamma(n-\alpha)} \frac{\mathrm{d}^n}{\mathrm{d}t^n} \int_a^t \frac{f(\tau)}{(t-\tau)^{\alpha-n+1}} \mathrm{d}\tau \tag{2}$$

for $(n-1<\alpha<n)$ and $\Gamma(\cdot)$ is *Euler's Gamma* function.

For convenience, Laplace domain notion is commonly used to describe the fractional integro-differential operation. The Laplace transform of the RL fractional derivative/integral under zero initial conditions for order α is given by

$$ \mathcal{L}\{{}_aD_t^{\pm\alpha} f(t)\} = s^{\pm\alpha} F(s). \tag{3}$$

In practice, the transfer functions involving fractional powers of s are approximated with usual (integer order) transfer functions with a similar behavior. Although many possible approximations exist (Vinagre & Podlubny, 2000), the well-known approximation is due to (Oustaloup, 1991). Oustaloup (1991) approximation makes use of a recursive distribution of poles and zeroes. The approximating transfer function is given by

$$ s^\alpha \approx k \prod_{n=1}^{N} \frac{1+(s/\omega_{z,n})}{1+(s/\omega_{p,n})}, \quad \alpha>0 \tag{4}$$

The approximation is valid in the frequency range $[\omega_l;\omega_h]$. Gain k is adjusted so that both sides of (4) shall have unit gain at 1 rad/s. The number of poles and zeros N is chosen beforehand. The good performance of approximation strongly depends on N, low values of N lead to simpler approximations and can cause the appearance of a ripple in both gain and phase behaviors while hight values of N will be computationally heavier though such a ripple can be removed. Frequencies of poles and zeroes in (4) are given as

$$ \omega_{z,1} = \omega_l\sqrt{\eta} \tag{5}$$

$$ \omega_{p,n} = \omega_{z,n}\varepsilon, \quad n=1,2,\ldots,N \tag{6}$$

$$ \omega_{z,n+1} = \omega_{p,n}\eta \quad n=1,2,\ldots,N-1 \tag{7}$$

$$ \varepsilon = (\omega_h/\omega_l)^{\alpha/N} \tag{8}$$

$$ \eta = (\omega_h/\omega_l)^{(1-\alpha)/N} \tag{9}$$

The case $\alpha<0$ can be handled by inverting (4). For $|\alpha|>1$, the approximation becomes unsatisfactory; for that reason, it is usual to split fractional powers of s like this

$$ s^\alpha = s^n s^\delta, \quad \alpha = n+\delta, \quad n\in\mathbb{Z}, \ \delta\in[0,1]. \tag{10}$$

As a result, only the latter term has to be approximated.

3. Chaotic ant swarm optimization method

To solve difficult optimum problems in science and engineering fields, researchers resorted to the swarm intelligence of ant colonies (or generally social insect societies) to develop new heuristic algorithms. ACO developed by Dorigo is one of the typical algorithms that stimulates the foraging behavior of ant colonies (Dorigo, Maniezzo, & Colorni, 1996). Cole (1991) revealed that ant colony exhibits a periodic behavior while a single ant shows low-dimensional deterministic chaotic activity patterns. The problem of how the chaotic behavior of single ant relates to the self-organization and foraging behaviors of the ant colony, however, has received little attention. Inspired by the self-organization behavior of ant swarm and the chaotic behavior of individual ant, Li et al. (2006a) proposed a new heuristic optimization algorithm, called chaotic ant swarm (CAS) optimization.

CAS is also a population-based search algorithm which combines the chaotic behavior of individual ant and the intelligent organization actions of ant colonies. In CAS, each ant can find food source, decide which one of the food source is better and memorize the information of the best food source. Ants move in the search space and exchange information with their neighbors. The search process of each ant consists of two phases, i.e., chaotic phase and organization phase. In order to achieve self-organization from chaotic state, an organization variable r_i is introduced for each ant i. Initially, the influence of the organization variable on the behavior individual ant is sufficiently small and each ant searches food in a chaotic manner. Due to the ergodic characteristic of chaos, ants have more chances to explore the global optimum in the search space. Then, with time evolving, the influence of organization variable on the behavior of each ant becomes stronger and stronger, the chaotic behavior of individual ant disappears and all the ants move toward the same position.

Consider an optimization problem in l-dimension space S, K ants locate in the space S and they try to minimize a function $f: S \to R$. Each points s in S is a valid solution to the specific optimization problem. The position of ith ant in the ant colony is assigned an algebraic variable symbol $z_i = (z_{i1}, z_{i2}, \ldots, z_{il})$, where $i = 1, 2, \ldots, K$. During the period of ants's motion, the position of single ant is influenced by three factors, i.e., the current position, the best position found by itself and any member of its neighbors and the organization variable. Mathematically, it can be described as

$$\begin{cases} y_i(n) = y_i(n-1)^{(1+r_i)} \\ z_{id}(n) = z_{id}(n-1)e^{(1-e^{-ay_i(n)})(3-\psi_d z_{id}(n-1))} \\ \qquad + (p_{id}(n-1) - z_{id}(n-1))e^{-2ay_i(n)+b} \end{cases} \tag{11}$$

where n indicates the current time step, $n-1$ the previous step. $z_{id}(n)$ is the current state of the dth dimension of the ith ant and $p_{id}(n)$ is the best position found by ith ant and its neighbors within $n-1$ steps, which is evaluated according to fitness of ith ant and its neighbors. $y_i(n)$ is the current state of the organization variable and its initial value usually selected as $y_i(0) = 0.999$. a is a sufficient large positive constant and can be selected as $a = 200$ or $a = 300$. b is a constant and $0 \leqslant b \leqslant \frac{2}{3}$. $r_i \in (0,1)$ is the organization factor of the ith ant. ψ_d determines the search ranges of dth element of variable in search space. Initially, since the organization variable $y_i(n)$ slightly influence the movement of ants, the motion of the ants approximately are governed by the following equation,

$$z_{id}(n) = z_{id}(n-1)e^{(3-\psi_d z_{id}(n-1))} \tag{12}$$

Eq. (12) is a chaotic map suggested by Solé, Miramontes, and Goodwin (1993). Thus, each ant in the colony walks in a chaotic way. As time evolves, the influence of organization variable $y_i(n)$ becomes stronger and stronger, the chaotic behavior of each ant disappears. After the "chaotic" search, the convergence rate will be mainly determined by

$$z_{id}(n) = z_{id}(n-1) + e^b(p_{id}(n-1) - z_{id}(n-1)) \tag{13}$$

In CAS, r_i and ψ_d are two important parameters. r_i has an effect on the convergence speed of CAS. If r_i is very large, the time of "chaotic" search is short, then the algorithm converges quickly and the desired optima or near-optima cannot be obtained. On the contrary, if r_i is very small, the time of "chaotic" search is long and the algorithm converges slowly. Typically, r_i is chosen as $0 < r_i \leqslant 0.5$ to allow small change as time evolves. ψ_d has the effect on the search range of CAS. If ψ_d is very small, the search scopes of CAS is large, and vice versa. If the search range is $[0, \omega_d]$, the relationship of ω_d between ψ_d is approximated by $\omega_d \approx 7.5/\psi_d$. The value of ψ_d should be suitably selected according to the concrete optimization problem. It shown that, if $\psi_d > 0$, Eq. (11) can be used to realize the search process in the intervals in which all $z_{id} \geqslant 0$. Contrarily, if $\psi_d < 0$, Eq. (11) can be used to realize the search process in the intervals in which all $z_{id} \leqslant 0$. If the elements of the optima located in all the ranges of real-numbered space, the following formula can be used, which is a modified version of Eq. (11)

$$\begin{cases} y_i(n) = y_i(n-1)^{(1+r_i)} \\ z_{id}(n) = \left(z_{id}(n-1) + V_i \times \frac{7.5}{\psi_d}\right) e^{(1-e^{-ay_i(n)})\left(3-\psi_d\left(z_{id}(n-1)+V_i\times\frac{7.5}{\psi_d}\right)\right)} \\ \qquad + (p_{id}(n-1) - z_{id}(n-1))e^{-2ay_i(n)+b} - V_i \times \frac{7.5}{\psi_d} \end{cases} \tag{14}$$

where $0 \leqslant V_i \leqslant 1$, V_i determines the search region of ith ant and offers the advantage that ants could search diverse regions of the problem space.

The neighbor selection can be defined in two ways. The first is the nearest fixed number of neighbors. The nearest m ants are selected as the neighbors of single ant. The second way is to consider the situation in which the number of neighbors increasing with iterative steps. This is due to the influence of self-organization behavior of ant i. The impact of organization will become stronger than before and the neighbor of the ant will increase. That is to say, the number of nearest neighbor is dynamically changed as time evolves or iterative steps increase. The number q of single ant is defined to increase for every T iterative steps.

4. Optimum FOPID cotroller design for AVR system using CAS

4.1. FOPID controller

The idea of using fractional-order controller for dynamic system belongs to Oustaloup (1991), who developed the so-called CRONE controller (CRONE is an abbreviation of Commande Robuste d'Ordre Non Entier). Then, Podlubny (1999) proposed a generalization of PID controller, which is called $PI^{\lambda}D^{\mu}$ controller. The transfer function of such a $PI^{\lambda}D^{\mu}$ controller is given by

$$G_c(s) = K_P + K_I s^{-\lambda} + K_D s^{\mu}, \quad (\lambda, \mu > 0) \tag{15}$$

The equation for the $PI^{\lambda}D^{\mu}$ controller's output in the time domain is

$$u(t) = K_p e(t) + K_I D^{-\lambda} e(t) + K_D D^{\mu} e(t) \tag{16}$$

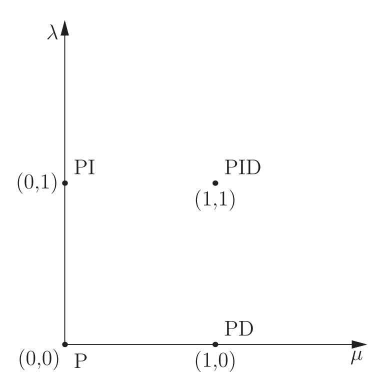


Fig. 1. The plane of FOPID controller.

If $\lambda = 1$ and $\mu = 1$, standard PID controller is obtained, and if $\lambda = 0$ and $\mu = 1$, we get a normal PD controller, $\lambda = 1$ and $\mu = 0$, a PI controller and $\lambda = 0$ and $\mu = 0$ a proportional gain. All these classical types of PID controllers are the special cases of the FOPID controller. Fig. 1 explains the relationship between FOPID controller and standard PID controller. It can be seen from Fig. 1 that FOPID controller generalizes the integer-order PID controller and expands it from point to plane. This expansion adds more flexibility to controller design and we can control our real-world processes more accurately. One of the advantage of FOPID controller is that FOPID controller is less sensitive to parameters change both of a controlled system and the controller itself.

4.2. Description of AVR system

A synchronous generator exhibits an oscillatory behavior around the equilibrium state when affected by disturbance such as sudden change in loads, change in transmission line parameters, fluctuations in the output of turbine and others. Such electromechanical oscillations is harmful to the stability of power system and even imposes some limitations on the power transfer capability in some cases. To enhance the dynamic stability of power system and guarantee the quality of it provides, most of the synchronous generators are equipped with excitation system, which is controlled by an automatic voltage regulator (AVR) and a power system stabilizer (PSS). Essentially, an AVR is to hold the terminal voltage magnitude of a synchronous generator at a specified level. The stability of the AVR system would seriously affect the security of the power system (Gaing, 2004). Therefore, it is necessary to design a controller for an AVR system to improve its stability and transient performance.

Usually, an AVR system consists of four components, i.e., amplifier, exciter, generator, and sensor. For convenience, a linearized model is considered here by taking into account the major time constant and ignoring the saturation or other nonlinearities. Transfer functions of these components are represented, respectively, as follows (Gaing, 2004; Yoshida, Kawata, Fukuyama, Takayama, & Nakanishi, 2000).

Amplifier model: The amplifier model is given by

$$G_A(s) = \frac{K_A}{1 + \tau_A s}, \tag{17}$$

where the typical value of K_A is in the range of [10, 400], and τ_A is very small ranging from 0.02 to 0.1 s.

Exciter model: The transfer function of a modern exciter may be represented by a gain K_E and a single time constant τ_E and is given by

$$G_E(s) = \frac{K_E}{1 + \tau_E s}, \tag{18}$$

where the typical value of K_E is in the range of [10, 400], and time constant τ_E ranges from 0.5 to 1.0 s.

Generator model: In the linearized model, the transfer function relating the generator terminal voltage to its field voltage can be represented by a gain K_G and a time constant τ_G

Table 1
Best CAS-FOPID controller parameters with different β values.

β	K_p	K_I	K_D	μ	λ
1.0	1.0537	0.4418	0.2510	1.1122	1.0624
1.5	0.9315	0.4776	0.2536	1.0838	1.0275

Table 2
Performance indices of the AVR system without controller and with CAS-FOPID controller.

Controller	M_p(%)	E_{ss}	t_r	t_s
without	65.68	0.0900	0.2612	6.9896
CAS-FOPID ($\beta = 1.0$)	0.1678	0.0014	0.2223	0.3037
CAS-FOPID ($\beta = 1.5$)	0.0640	0.0012	0.2305	0.3187

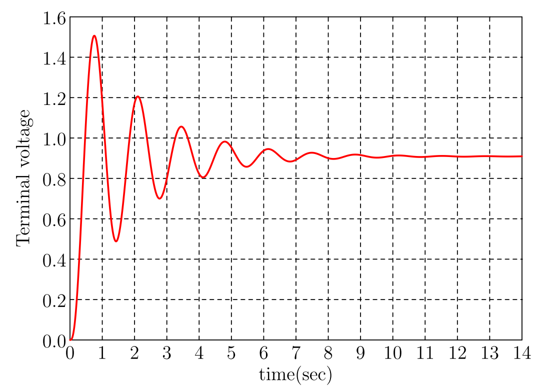


Fig. 3. Terminal voltage step response of AVR without controller.

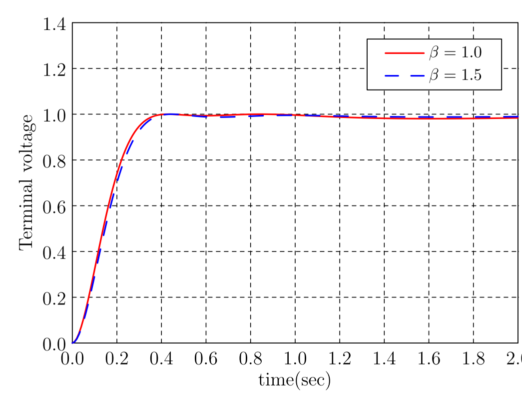


Fig. 4. Terminal voltage step response of AVR system controlled by CAS-FOPID controller.

$$G_G(s) = \frac{K_G}{1 + \tau_G s}. \tag{19}$$

These constants are load dependent, K_G may vary between 0.7 to 1.0, and τ_G between 1.0 to 2.0 from full load to null load.

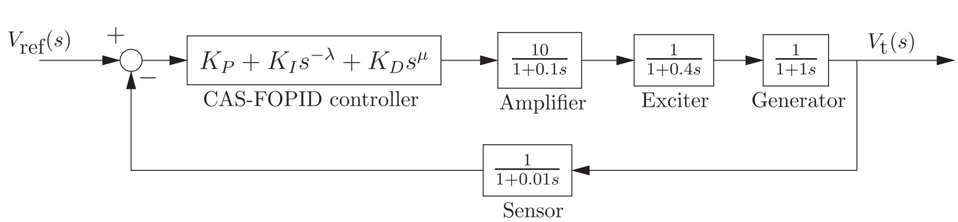


Fig. 2. Block diagram of AVR with FOPID controller.

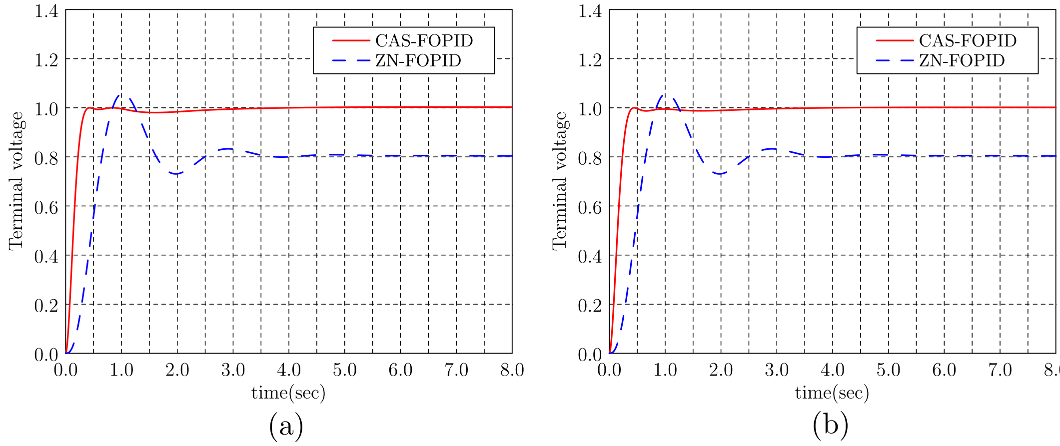


Fig. 5. Terminal voltage step response of AVR system controlled by CAS-FOPID and ZN-FOPID controller (a) $\beta = 1.0$ (b) $\beta = 1.5$.

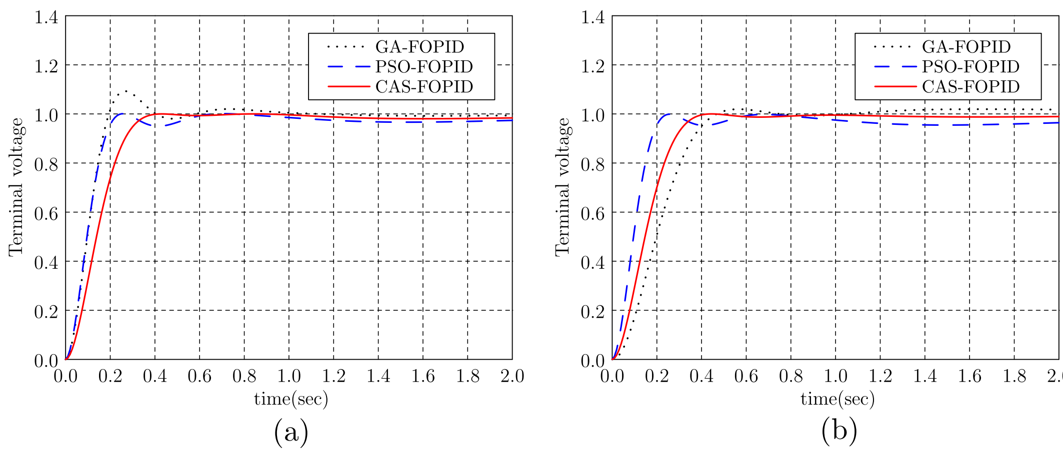


Fig. 6. Terminal voltage step response of the AVR system controller by GA-FOPID, PSO-FOPID, and CAS-FOPID (a) $\beta = 1.0$ (b) $\beta = 1.5$.

Table 3
Best controller parameters and performance indices of different FOPID controllers.

	Controller	K_P	K_I	K_D	μ	λ	$M_p(\%)$	E_{ss}	t_r	t_s
$\beta = 1.0$	ZN-FOPID	0.3261	0.1313	−0.022	−0.1277	0.1424	32.054	0.1982	0.3952	2.2646
	GA-FOPID	1.6947	0.8849	0.3964	1.1296	1.0248	9.26	0.0006	0.1298	0.3395
	PSO-FOPID	1.6264	0.2956	0.3226	1.1980	1.3183	0.0953	0.0047	0.13752	0.4563
	CAS-FOPID	1.0537	0.4418	0.2510	1.1122	1.0624	0.1678	0.0014	0.2223	0.3037
$\beta = 1.5$	GA-FOPID	0.5073	0.5724	0.2550	0.8851	0.9384	2.0929	0.0016	0.2745	0.39973
	PSO-FOPID	1.6986	0.1797	0.3122	1.2081	1.8373	2.0689	0.0300	0.1372	0.1972
	CAS-FOPID	0.9315	0.4776	0.2536	1.0838	1.0275	0.0642	0.0012	0.2305	0.3187

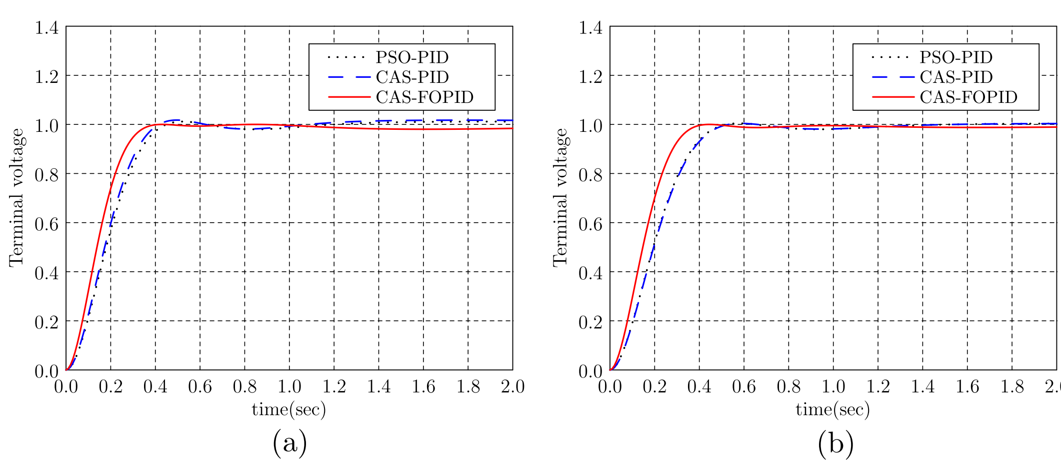


Fig. 7. Terminal voltage step response of the AVR system controlled by CAS-FOPID, CAS-PID and PSO-PID (a) $\beta = 1.0$ (b) $\beta = 1.5$.

Table 4
Best controller parameters and performance indices of different PID controllers.

	Controller	K_P	K_I	K_D	μ	λ	$M_p(\%)$	E_{ss}	t_r	t_s
$\beta = 1.0$	PSO-PID	0.6254	0.45779	0.2187	–	–	1.1592	1.3428e−07	0.2678	0.3756
	CAS-PID	0.6746	0.6009	0.2618	–	–	1.7678	5.6296e−08	0.2425	0.3550
	CAS-FOPID	1.0537	0.4418	0.2510	1.1122	1.0624	0.1678	0.0014	0.2223	0.3037
$\beta = 1.5$	PSO-PID	0.6254	0.4577	0.2187	–	–	0.4400	3.6139e−08	0.2997	0.4156
	CAS-PID	0.6202	0.4531	0.2152	–	–	0.4000	2.6883e−08	0.3156	0.4212
	CAS-FOPID	0.9315	0.4776	0.2536	1.0838	1.0275	0.0642	0.0012	0.2305	0.3187

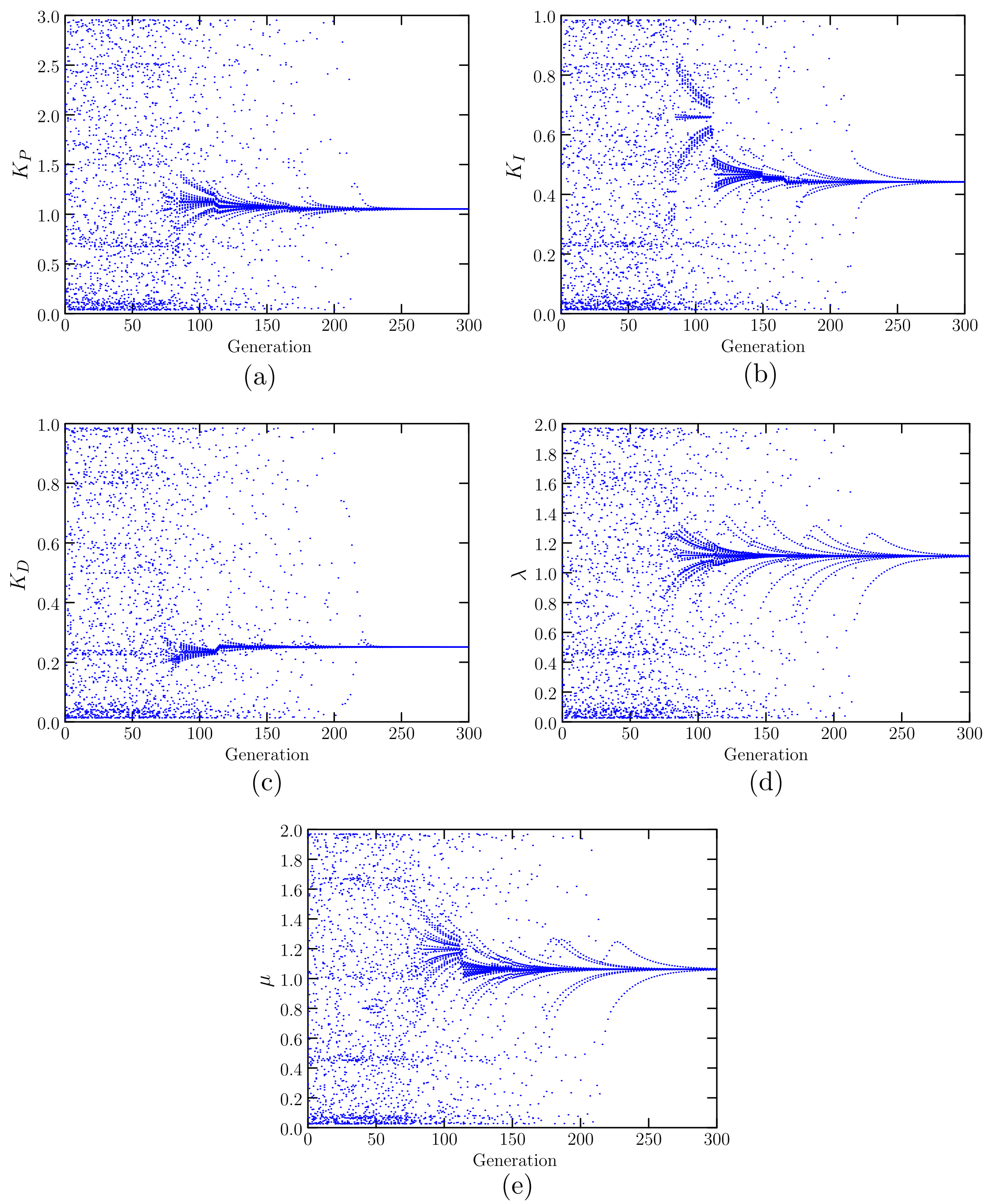


Fig. 8. Evolutionary process of parameters of CAS-FOPID, $\beta = 1.5$.

Sensor model: The sensor circuit, which rectifies, filters, and reduces the terminal voltage, is modeled by the following simple first-order transfer function

$$G_S = \frac{K_S}{1 + \tau_S s}, \qquad (20)$$

where τ_S ranges from 0.001 to 0.06 s.

4.3. CAS-FOPID controller design for AVR system

In this subsection, a FOPID controller is developed to improve the step transient response of an AVR system. The optimal FOPID controller parameters K_P, K_I, K_D, λ, μ are determined using CAS algorithm through optimizing a performance criterion. It is expected that a good step response of the AVR system controlled by the CAS-FOPID controller can be obtained.

4.3.1. Solution representation

Searching the optimal parameters of FOPID controller is an optimum problem in essence. CAS is used for this task in this paper. In this context, the position of each ant is represented by a real vector with five components, i.e., $Z = [K_P, K_I, K_D, \lambda, \mu]$. If there exist K ants in the swarm, then, the whole swarm is represented by a matrix with dimension of $K \times 5$.

4.3.2. Fitness function definition

To evaluate the fitness of each ant, a fitness function should be defined. There are several evaluation criterions for design of controllers. Integral of absolute error (IAE), integral of squared error (ISE) and integral of time-weighted-squared-error (ITSE) are often used in literatures to design controllers. However, there are some disadvantages associated with these criterions as stated in Gaing (2004) and Zamani et al. (2009). Zamani et al. (2009) proposed a new criterion function with eight terms including time domain and frequency domain measures. The significance of each term is determined by a weight factor w_i. However, it is not convenient to select appropriated weighted factor. In Gaing (2004), a performance criterion $W(\widetilde{Z})$ in time domain is defined for PID controller design. $W(\widetilde{Z})$ is defined as

$$W(\widetilde{Z}) = (1 - e^{\beta}) \cdot (M_p + E_{ss}) + e^{-\beta} \cdot (t_s - t_r) \tag{21}$$

where $\widetilde{Z} = (K_P, K_I, K_D)$ is a parameter set of PID controller, β is a weighting factor, M_p, E_{ss}, t_r and t_s are respectively the overshoot, steady-state error, raising time and settling time. This performance criterion is adopted in this paper as the fitness function of each ant. Furthermore, the designed controller should guarantee stability of the closed loop. If it is not so, the controller is invalid and should be penalized; otherwise, it is not penalized. Taking the above requirements into account, the fitness function to evaluate each ant is

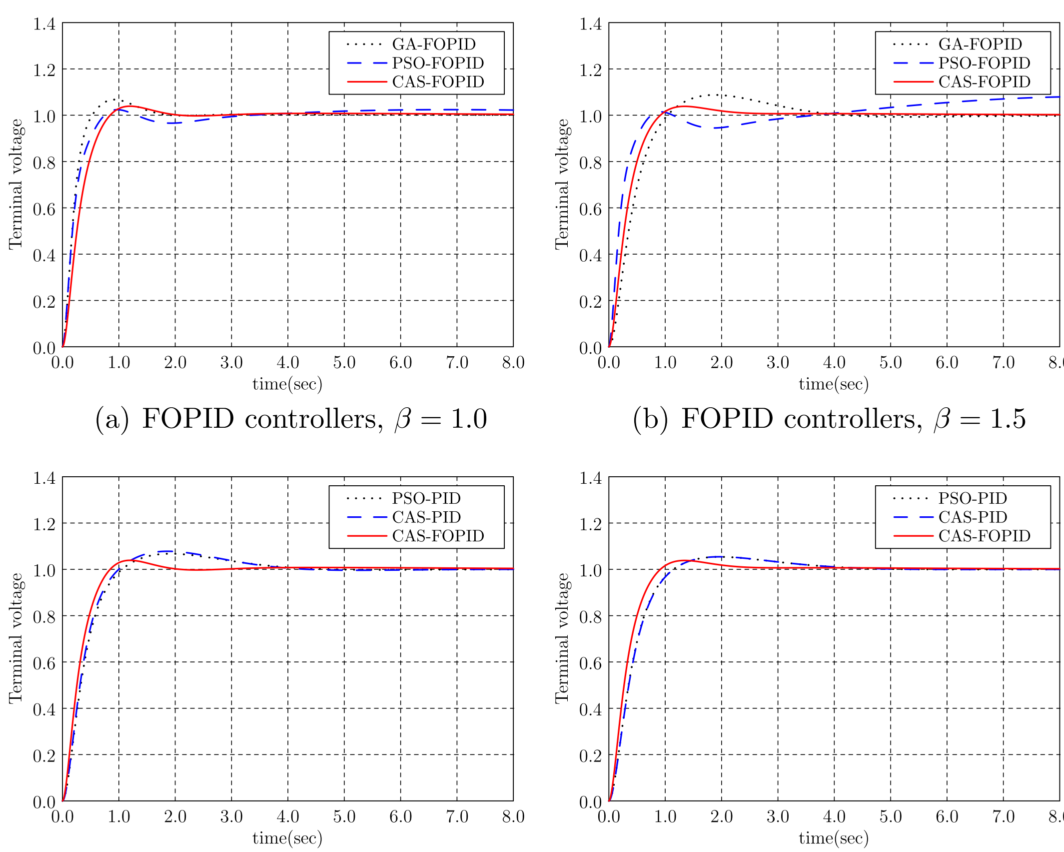
$$J(Z) = \begin{cases} (1 - e^{\beta}) \cdot (M_p + E_{ss}) + e^{-\beta} \cdot (t_s - t_r) & \text{if } Z \text{ is stable} \\ L & \text{if } Z \text{ is unstable} \end{cases}, \tag{22}$$

where L is a large positive real number. The optimum FOPID controller is one that minimizes the objective function $J(Z)$.

4.3.3. The design process of CAS-FOPID

The design process of CAS-FOPID controller for AVR system is summarized as follows:

Step 1 : Determine the parameters of AVR system.
Step 2 : Set the parameters of CAS algorithm.
Step 3 : Specify the lower and upper bounds of the five controller parameters.
Step 4 : Set $t = 0$ and randomly initialize the position of K ants within the range of each controller parameter.
Step 5 : For each ant, calculate the integer-order approximation transfer function of FOPID controller using Oustaloup method.



(a) FOPID controllers, $\beta = 1.0$ (b) FOPID controllers, $\beta = 1.5$ (c) PID controllers, $\beta = 1.0$ (d) PID controllers, $\beta = 1.5$

Fig. 9. Terminal voltage step response with parameter uncertainties in generator model.

Step 6 : Apply the integer-order approximation transfer function of FOPID controller to the AVR system.
Step 7 : Test the stability of the closed-loop system.
Step 8 : Calculate overshoot M_p, steady-state error E_{ss}, raise time t_r and settle time t_r from the step response of the closed-loop system.
Step 9 : Evaluate the fitness of each ant using Eq. (22);
Step 10 : Determine the neighbors of each ant and update the best position P_i found by the ith ant and its neighbors.
Step 11 : Move the ant to a new position according to Eq. (14).
Step 12 : Let $t = t + 1$ and go to Step 5 until the maximum number of iteration is reached.
Step 13 : Output the best position found by the whole ant swarm.

5. Numerical results

To demonstrate the efficiency of the FOPID controller, a practical high-order AVR is tested.

5.1. Related parameters

AVR system parameters: The AVR system parameters are selected as in Gaing (2004), i.e., $K_A = 10$, $\tau_A = 0.1$, $K_E = 1$, $\tau_E = 0.4$, $K_G = 1$, $\tau_G = 1$, $K_S = 1$, $\tau_S = 0.01$. The lower and upper bounds of each FOPID controller parameter is $0 \leqslant K_P \leqslant 3.0$, $0 \leqslant K_I \leqslant 1.0$, $0 \leqslant K_D \leqslant 1.0$, $0 \leqslant \lambda \leqslant 2.0$, $0 \leqslant \mu \leqslant 2.0$. In Oustaloup approximation, $\omega_l = 0.001\ \omega_c$, $\omega_h = 1000\ \omega_c$, where ω_c is the gain cross frequency, and $N = 6$ is used. An AVR system equipped with a FOPID controller is shown in Fig. 2.

CAS algorithm parameters: the number of ants $K = 20$, $a = 300$, $b = 2/3$, $r_i = 0.04 + 0.1{*}\mathrm{rand}$, $y_i(0) = 0.999$. ψ_d $(d = 1,2,\ldots,5)$ is set using the approximating relation $\psi_d \approx 7.5/\omega_d$ according to the lower and upper bounds of each parameter. The maximum iteration number is 300.

5.2. Performance of CAS-FOPID controller

In this section, the performance of CAS-FOPID is examined. First, we consider the AVR system without any controllers. The terminal voltage step response of such an AVR system is shown in Fig. 2. Large overshoot, long settling time and oscillation are observed in Fig. 2. Secondary, AVR system with a CAS-FOPID controller is considered. To obtain the optimal parameters of FOPID controller, the CAS algorithm is executed ten runs. The obtained best controller parameters are listed in Table 1. The terminal voltage step response of the AVR system controlled by CAS-FOPID controller with the best parameters is shown in Fig. 4. It can be seen from Fig. 4 that the step response of the AVR system controlled by CAS-FOPID controller has small overshoot, short settling time, and furthermore, the oscillation is eliminated. Table 2 lists the performance indices in the time domain of the step responses shown in Figs. 3 and 4.

5.3. Comparison with other controllers

To demonstrate its advantage, the proposed CAS-FOPID controller are extensively compared with other FOPID controllers and PID controllers.

5.3.1. Comparison with FOPID controllers

For comparison purpose, the FOPID controllers tuned by Ziegler–Nichols rules in Valéio and da Costa (2006) (ZN-FOPID),

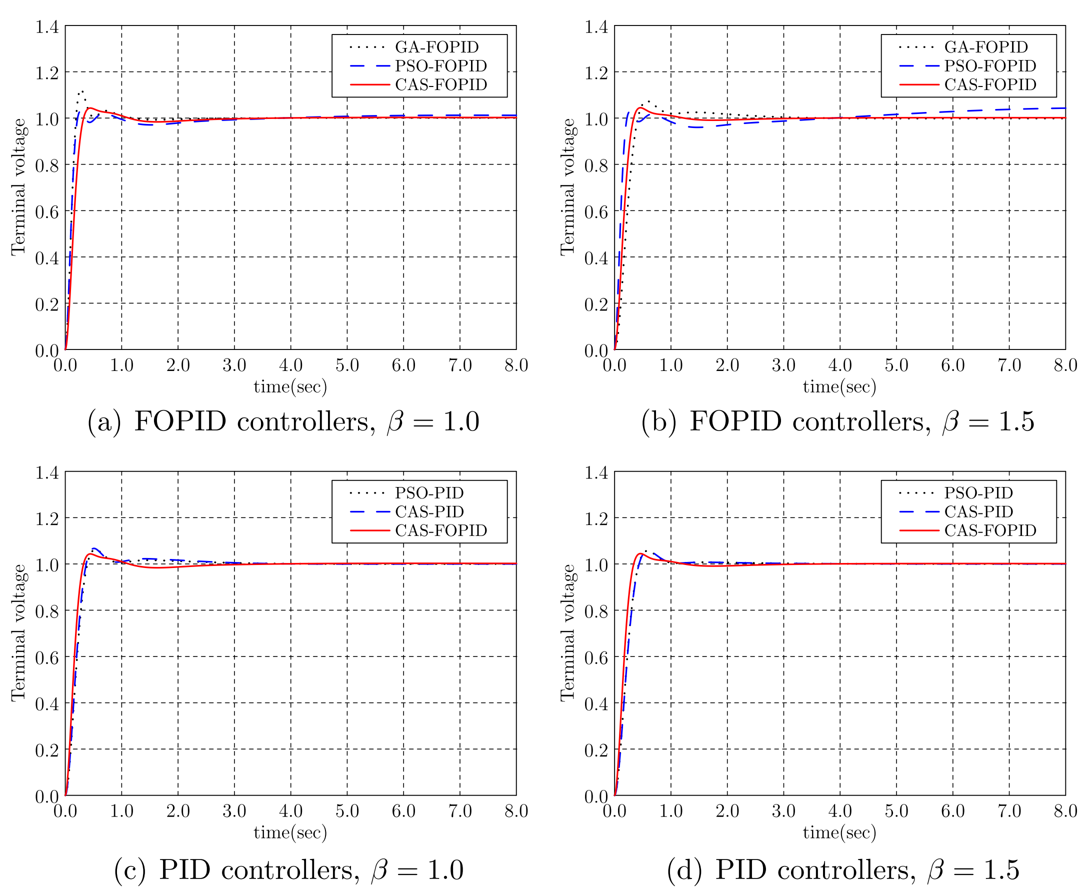


Fig. 10. Terminal voltage step response with parameter uncertainties in exciter model.

real value GA-FOPID controller and PSO-FOPID controller are implemented. The GA algorithm is implemented using GAOT toolbox from http://www.ise.ncsu.edu/mirage/GAToolBox/gaot/ with parameters as follows: the selection method is "normGeomSelect" with parameter option 0.08, the crossover method is "heuristicXover" with parameter option [2 3] and the mutation method is "multiNonUnifMutation" with parameter option [6 genMax 3]. The PSO algorithm is implemented as in Gaing (2004). Similarly, the GA and PSO algorithm are executed ten runs to obtain the best controller parameters using the same fitness function and individual definition as CAS. The terminal voltage step responses of the AVR system controlled by ZN-FOPID controller, GA-FOPID controller, PSO-FOPID controller and CAS-FOPID are shown in Figs. 5 and 6, respectively. The best controller parameters and the performance indices in the time domain are listed in Table 3. As revealed by Figs. 5, 6 and Table 3, the CAS-FOPID controller has better performance than the ZN-FOPID controller, GA-FOPID controller, PSO-FOPID controller. The CAS-FOPID controller can create perfect step response of the AVR system.

5.3.2. Comparison with PID controllers

Also, we compare the CAS-FOPID controller with PSO-PID controller, CAS-PID controller. The parameters of PSO-PID is the same as in Gaing (2004). The parameters of CAS-PID controller are obtained through optimizing the same fitness function using CAS algorithm. The terminal voltage step responses of the AVR system controlled by PSO-PID controller, CAS-PID controller and CAS-FOPID controller are shown in Fig. 7. The controller parameters and performance indices in the time domain are listed in Table 4. It is observed from Fig. 7 and Table 4 that the CAS-FOPID has better performance compared to PSO-PID controller and CAS-PID controller. The terminal voltage step response of the AVR system controlled by CAS-FOPID controller has smaller overshoot, raising time and settling time.

5.4. Convergence

To see the convergence characteristic of CAS algorithm, Fig. 8 shows the evolutionary process of each parameter of CAS-FOPID controller corresponding to $\beta = 1.5$. It can be seen from Fig. 8 that the evolutionary process of each parameter undergoes two phases, i.e., chaotic phase and organization phase. After an initial chaotic search, CAS algorithm converges toward the optimal value, which is consistent to theoretic analysis mentioned in Section 3.

5.5. Robustness test

To show the robustness of CAS-FOPID controller, three kinds of parameter uncertainties are considered.

5.5.1. Generator uncertainty

In this case, it is assumed that the generator model parameters change to $K_G = 0.7$, $\tau_G = 1.6$ from actual value $K_G = 1.0$, $\tau_G = 1.0$ due to the change in load condition. The terminal voltage step response with previously designed FOPID and PID controllers are shown in Fig. 9.

5.5.2. Exciter uncertainty

In this case, it is assumed that the exciter model parameters change to $K_E = 1.2$, $\tau_E = 0.5$ from actual value $K_E = 1.0$, $\tau_G = 0.4$. The terminal voltage step response with previously designed FOPID and PID controllers are shown in Fig. 10.

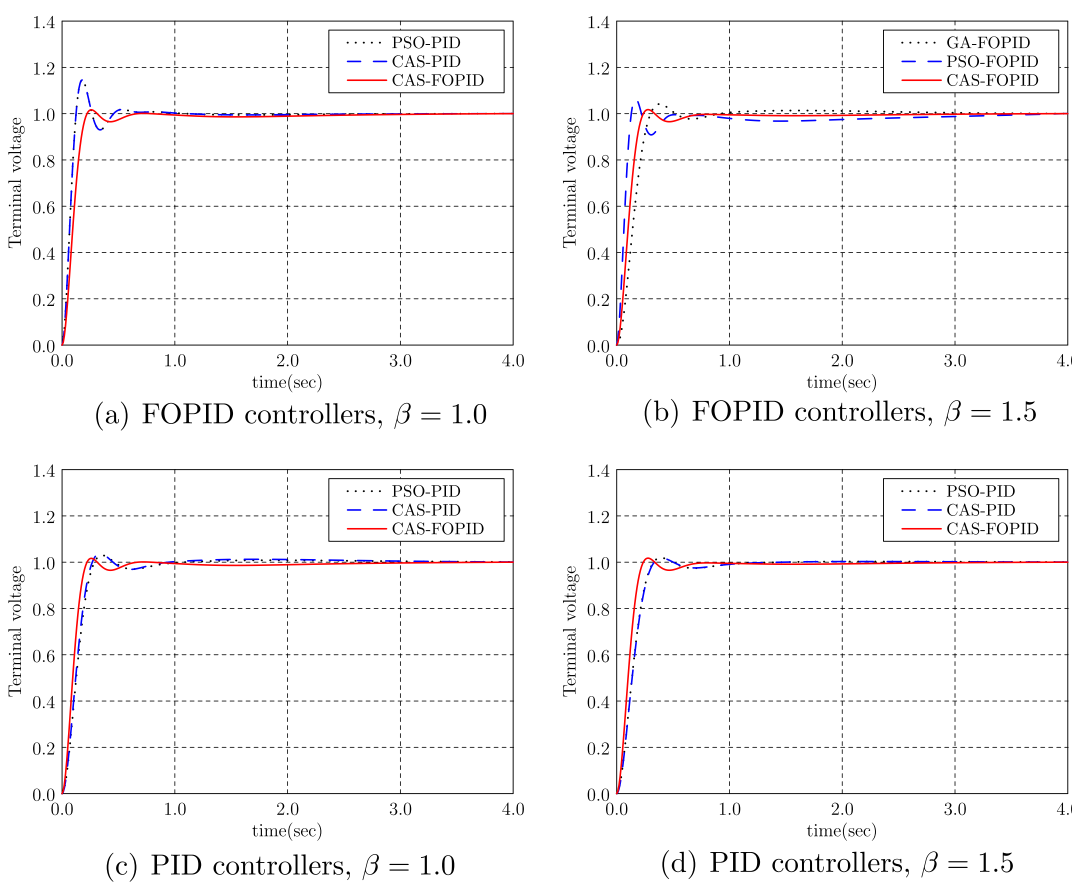


Fig. 11. Terminal voltage step response with parameter uncertainties in amplifier model.

5.5.3. Amplifier uncertainty

In this case, it is assumed that the parameters of amplifier change to $K_A = 14$, $\tau_A = 0.07$ from actual value $K_A = 10$, $\tau_A = 0.1$. The terminal voltage step response with previously designed FO-PID and PID controllers are shown in Fig. 11.

It can be seen from Figs. 9–11 that CAS-FOPID controller is more robust to uncertainties than other FOPID and PID controllers.

6. Conclusions

In this paper, a novel design method for the tuning of FOPID controller parameters is present. The proposed method determines the parameters of FOPID controller through minimizing a nonlinear objective function consisting of overshoot, steady-state error, raising time and settling time. CAS algorithm is responsible to search the optimal parameters. The designed CAS-FOPDI controller has been successfully applied to AVR system. Numerical simulations show that the CAS-FOPID controller has superior performance than other FOPID controllers and PID controllers. Furthermore, the CAS-FOPID controller is more robust to model uncertainties.

Acknowledgments

This work is supported in part by Science Fund for Distinguished Young Scholars of Hebei Province (No. F2011203110), the Program for New Century Excellent Talents in the University of China (No. NCET-08-0658), the National Natural Science Foundation of China (No.60974018, 60934003), the National Science Foundation of China Innovative Grant (Grant No. 70921061), the CAS/SAFEA International Partnership Program for Creative Research Teams, the Foundation for the Author of National Excellent Doctoral Dissertation of PR China (FANEDD) (No. 200951), the Asia Foresight Program under NSFC (No. 61161140320) and Natural Scientific Research Foundation of the Higher Education Institutions of Hebei Province(No. 2010165).

References

Ahn, H. S., Bhambhani, V., & Chen, Y. Q. (2009). Fractional-order integral and derivative controller for temperature profile tracking. *Sādhanā, 34*, 833–850.

Biswas, A., Das, S., Abraham, A., & Dasgupta, S. (2009). Design of fractional-order $PI^{\lambda}D^{\mu}$ controllers with an improved differential evolution. *Engineering Applications of Artificial Intelligence, 22*, 343–350.

Bouafoura, M. K., & Braiek, N. B. (2010). $PI^{\lambda}D^{\mu}$ controller design for integer and fractional plants using piecewise orthogonal functions. *Communications in Nonlinear Science and Numerical Simulation, 15*, 1267–1278.

Cai, J., Ma, X., Li, Q., Li, L., & Peng, H. (2010). A multi-objective chaotic ant swarm optimization for environmental/economic dispatch. *International Journal of Electrical Power & Energy Systems, 32*, 337–344.

Cai, J., Ma, X., Li, L., Yang, Y., Peng, H., & Wang, X. (2007). Chaotic ant swarm optimization to economic dispatch. *Electric Power Systems Research, 77*, 1373–1380.

Cervera, J., Banos, A., Monje, C., Vinagre, B., (2006). Tuning of fractional PID controllers by using QFT. In *32nd annual conference on IEEE industrial electronics, IECON 2006* (pp. 5402–5407).

Changmao, Q., Naiming, Q., Zhiguo, S. (2010). Fractional PID controller design of hypersonic flight vehicle. In *international conference on computer, mechatronics, control and electronic engineering (CMCE) 2010* (pp. 466–469).

Cole, B. J. (1991). Is animal behavior chaotic? Evidence from the activity of ants. *Proceedings of Royal Society of London B, Biological Science, 244*, 253–259.

Domingues, J., Valerio, D., da Costa, J.S. (2009). Rule-based fractional control of an irrigation canal. In *Proceedings 35th annual conference of IEEE industrial electronics IECON '09* (pp. 1712–1717).

Dorigo, M., Maniezzo, V., & Colorni, A. (1996). Ant system: Optimization by a colony of cooperating agents. *IEEE Trans Systems, Man Cybernetics-Part B: Cybernetics, 26*, 29–41.

Fan, H., Sun, Y., Zhang, X. (2007). Research on fractional order controller in servo press control system. In *International Conference on Mechatronics and Automation, ICMA 2007* (pp. 2934 –2938).

Gaing, Z. L. (2004). A particle swarm optimization approach for optimum design of PID controller in AVR system. *IEEE Transactions on Energy Conversion, 19*, 384–391.

Hong, W. C. (2010). Application of chaotic ant swarm optimization in electric load forecasting. Energy Policy 38, 5830–5839. The socio-economic transition towards a hydrogen economy – findings from European research, with regular papers.

Lee, C. H., & Chang, F. K. (2010). Fractional-order PID controller optimization via improved electromagnetism-like algorithm. *Expert Systems with Applications, 37*, 8871–8878.

Li, L. X., Peng, H. P., Wang, X. D., & Yang, Y. X. (2006a). An optimization method inspired by chaotic ant behavior. *International Journal of Bifurcation and Chaos, 16*, 2351–2364.

Li, L. X., Yang, Y. X., & Peng, H. P. (2009). Fuzzy system identification via chaotic ant swarm. *Chaos, Solitons & Fractals, 41*, 401–409.

Li, L. X., Yang, Y. X., Peng, H. P., & Wang, X. D. (2006b). Parameters identification of chaotic systems via chaotic ant swarm. *Chaos, Solitons & Fractals, 28*, 1204–1211.

Meng, L., Xue, D. (2009). Design of an optimal fractional-order PID controller using multi-objective GA optimization. In *Control and Decision Conference CCDC '09*. Chinese, pp. 3849–3853.

Monje, C. A., Vinagre, B. M., Chen, Y. Q., Feliu, V., Lanusse, P., Sabatier, J. (2004). Proposals for fractional $PI^{\lambda}D^{\mu}$ tuning. In *1st IFAC workshop on fractional derivatives and applications*, Bordeaux, France.

Monje, C. A., Vinagre, B. M., Feliu, V., & Chen, Y. (2008). Tuning and auto-tuning of fractional order controllers for industry applications. *Control Engineering Practice, 16*, 798–812.

Oustaloup, A. (1991). La commande CRONE: commande robuste d'order non entier. Hermès, Paris.

Padula, F., & Visioli, A. (2011). Tuning rules for optimal PID and fractional-order PID controllers. *Journal of Process Control, 21*, 69–81.

Peng, H., Li, L., Yang, Y., & Liu, F. (2010). Parameter estimation of dynamical systems via a chaotic ant swarm. *Physical Review E, 81*. 016207(11).

Petráš, I. (2009). Fractional-order feedback control of a DC motor. *Journal of Electrical Engineering, 60*, 117–128.

Podlubny, I. (1999). Fractional-order systems and $PI^{\lambda}D^{\mu}$ controllers. *IEEE Transactions on Automatic Control, 44*, 208–214.

Solé, R. V., Miramontes, O., & Goodwin, B. C. (1993). Oscillations and chaos in ant societies. *Journal of Theoretical Biology, 161*, 343–357.

Tang, Y. G., Cui, M. Y., Li, L. X., Peng, H. P., & Guan, X. P. (2009). Parameter identification of time-delay chaotic system using chaotic ant swarm. *Chaos, Solitons & Fractals, 41*, 2097–2102.

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

Valéio, D., & da Costa, J. S. (2006). Tuning of fractional PID controllers with Ziegler–Nichols-type rules. *Signal Processing, 86*, 2771–2784. Special Section: Fractional Calculus Applications in Signals and Systems.

Vinagre, B. M., & Podlubny, I. (2000). Some approximations of fractional order operators used in control theory and applications. *Fractional Calculus & Applications and Analysis, 3*, 231–248.

Wan, M., Li, L. X., Xiao, J. H., Yang, Y. X., Wang, C., & Guo, X. L. (2010). CAS based clustering algorithm for Web users. *Nonlinear Dynamics, 61*, 347–361.

Wei, Z., Ge, F. Z., Lu, Y., Li, L. X., & Yang, Y. X. (2011). Chaotic ant swarm for the traveling salesman problem. *Nonlinear Dynamics, 65*, 271–281.

Xue, D. Y., Zhao, C. N., Chen, Y. Q. (2006). Fractional order PID control of a DC-motor with elastic shaft: A case study. In *Proceedings american control conference*.

Yoshida, H., Kawata, K., Fukuyama, Y., Takayama, S., & Nakanishi, Y. (2000). A particle swarm optimization for reactive power and voltage control considering voltage security assessment. *IEEE Transactions on Power Systems, 15*, 1232–1239.

Zamani, M., Karimi-Ghartemani, M., Sadati, N., Parniani, M. (2009). Design of a fractional order PID controller for an AVR using particle swarm optimization. Control Engineering Practice: Vol. 17. In *Special Section: The 2007 IFAC symposium on advances in automotive control* (pp. 1380–1387).

Zhu, H., Li, L. X., Zhao, Y., Guo, Y., & Yang, Y. X. (2009). CAS algorithm-based optimum design of PID controller in AVR system. *Chaos, Solitons & Fractals, 42*, 792–800.