Optimum Design of Fractional Order PID Controller for an AVR System Using an Improved Artificial Bee Colony Algorithm

ZHANG Dong-Li[1] TANG Ying-Gan[1, 2] GUAN Xin-Ping[3]

Abstract Fractional order proportional-integral-derivative (FOPID) controller generalizes the standard PID controller. Compared to PID controller, FOPID controller has more parameters and the tuning of parameters is more complex. In this paper, an improved artificial bee colony algorithm, which combines cyclic exchange neighborhood with chaos (CNC-ABC), is proposed for the sake of tuning the parameters of FOPID controller. The characteristic of the proposed CNC-ABC exists in two folds: one is that it enlarges the search scope of the solution by utilizing cyclic exchange neighborhood techniques, speeds up the convergence of artificial bee colony algorithm (ABC). The other is that it has potential to get out of local optima by exploiting the ergodicity of chaos. The proposed CNC-ABC algorithm is used to optimize the parameters of the FOPID controller for an automatic voltage regulator (AVR) system. Numerical simulations show that the CNC-ABC FOPID controller has better performance than other FOPID and PID controllers.

Key words Fractional order proportional-integral-derivative (FOPID) controller, optimal control, artificial bee colony algorithm, cyclic exchange neighborhood, chaos, automatic voltage regualator (AVR) system

Citation Zhang Dong-Li, Tang Ying-Gan, Guan Xin-Ping. Optimum design of fractional order PID controller for an AVR system using an improved artificial bee colony algorithm. *Acta Automatica Sinica*, 2014, **40**(5): 973−980

DOI 10.3724/SP.J.1004.2014.00973

Fractional calculus is a generalization of integer orders calculus and it extends regular integer orders to non-integer orders case. Compared to classical integer calculus, fractional calculus computationally requires large memory which makes it able to represent the dynamics of the real system more exactly. In the area of control theory and control engineering, the proportional-integral-derivative (PID) control is the most widely utilized control method because of its simple structure and strong robustness. Nowadays, more than 90 % controllers in industry are PID controllers[1].

In recent years, the combination of fractional calculus and PID control theory has got much attention of researchers. Fractional order PID (FOPID) controller was proposed by Podlubny in 1999[2], which is written as $PI^{\lambda}D^{\mu}$, where λ and μ are the integrating and derivative orders and they are non-integers. FOPID controller has five parameters, i.e., the proportional gain, the integral gain, the derivative gain, the integrating order, and the derivative order. Compared to PID controller, FOPID has extra two parameters, i.e., the integrating order λ and the derivative order μ, which means that researchers have more freedom in the designing of FOPID controller. On the other hand, its parameters tuning is more complex. Different methods for tuning the parameters of FOPID controllers have been reported in the literatures. Valério et al. proposed Ziegler-Nichols type tuning rules for FOPID controller[3]. Cervera et al. applied quantitative feedback theory (QFT) to tune the FOPID controller[4]. Genetic algorithm[5], differential evolution (DE)[6], particle swarm optimization (PSO)[7], and ant colony optimization (ACO)[8] were also used to design FOPID controller.

Artificial bee colony (ABC) algorithm was proposed by Karaboga in 2005[9]. It imitates the intelligent foraging behavior of honey bees to solve numerical optimization problems[9−13]. ABC algorithm has the characteristics of simplicity and ease of implementation. Many practical problems such as fractional order controller[14], clustering[15] and lot-streaming flow shop scheduling problem[16] have been solved using ABC algorithm. However, there are two disadvantages associated with ABC algorithm, i.e., the convergence speed is slow and it is often trapped into local optima. Many authors make efforts to overcome the drawbacks of ABC. Bao et al. proposed a chaos-artificial bee colony algorithm of self-adapting search space[17]. Lee et al. improved ABC algorithm based on diversity strategy[18]. Gao et al. improved ABC combined differential evolution algorithm[19].

In nature, ABC algorithm can be viewed as a class of neighborhood search algorithm (NSA). Many difficult optimization problems have been solved by using NSA. NSA obtains the optimal solution by searching the "neighborhood" of the current solution. Selecting an appropriate neighborhood structure is important for NSA because the neighborhood structure determines whether the solutions are highly accurate or very poor. Several powerful neighborhood structures have been presented such as compound neighborhood structures[20−21], multi-exchange neighborhood structure[22] etc. The multi-exchange neighborhood consists of two types of structure: cyclic exchange and path exchange. The advantage of multi-exchange neighborhood is that it can identify improvement moves in an associated improvement graph. In the literature, multi-exchange neighborhood structure algorithms have been applied to many complicated optimization problems. Tang et al. proposed an iterated local search (ILS) algorithm based on cyclic exchange neighborhood to solve parallel machine scheduling problems[23]. Ahuja et al. solved the capacitated minimum spanning tree problems by multi-exchange neighborhood[22]. Frangioni et al. applied multi-exchange neighborhood for minimum makespan parallel machine

Manuscript received June 19, 2013; accepted November 12, 2013
Supported by National Natural Science Foundation of China (61273260), Specialized Research Fund for the Doctoral Program of Higher Education of China (20121333120010), Natural Scientific Research Foundation of the Higher Education Institutions of Hebei Province (2010165), the Major Program of the National Natural Science Foundation of China (61290322), Foundation of Key Laboratory of System Control and Information Processing, Ministry of Education (SCIP2012008), and Science and Technology Research and Development Plan of Qinhuangdao City (2012021A041)
Recommended by Associate Editor CHEN Jie
1. Institute of Electrical Engineering, Yanshan University, Qinhuangdao 066004, China 2. Key Laboratory of System Control and Information Processing, Ministry of Education, Shanghai 200240, China 3. Institute of Electronic, Information, and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China

scheduling problems[24]. Thompson et al. presented a cyclic transfer neighborhood method for multi-vehicle routing and scheduling problems[25].

In this paper, a modified ABC algorithm is presented using cyclic exchange neighborhood and chaos for tuning the parameters of the FOPID controller. Exchanging neighborhood enables to enlarge the search scope of the solutions, so as to speed up the convergence of the algorithm. The ergodicity and randomness of the chaos make the algorithm be able to jump out of local optima.

This paper proceeds as follow. Section 1 briefly reviews the basic of fractional calculus. In Section 2, the behavior of honey bees foraging, ABC algorithm and the cyclic exchange neighborhood structure are firstly reviewed. Then, the modified ABC algorithm based on cyclic exchange neighborhood and chaos is introduced. The design method of FOPID controller using CNC-ABC algorithm for an AVR system is described in Section 3. The numerical results are given in Section 4. Finally, the concluding remarks are presented in Section 5.

1 Fractional calculus

Fractional calculus generalizes the ordinary calculus. The basic operation ${}_aD_t^\alpha$ is defined as:

$$
{}_aD_t^\alpha = \begin{cases} \dfrac{\mathrm{d}^\alpha}{\mathrm{d}t^\alpha}, & R(\alpha > 0) \\ 1, & R(\alpha = 0) \\ \displaystyle\int_a^t (\mathrm{d}t)^{-\alpha}, & R(\alpha < 0) \end{cases} \tag{1}
$$

where a and t are the lower and upper limits, α is the order of the operation[26] and it is a complex number, $R(\alpha)$ is the real part of α. Fractional derivative has several definitions. The usually used definitions are given by Riemann-Liouville (RL), Grü Nwald-Letnikor (GL) and Caputo. The RL definition is

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

where m is the first integer which is not less than α, i.e., $m - 1 < \alpha < m$. $\Gamma(\cdot)$ is famous Euler Gamma function. The GL definition is

$$
{}_aD_t^\alpha f(t) = \lim_{h\to 0} \frac{1}{\Gamma(\alpha) h^\alpha} \sum_{k=0}^{\frac{(t-\alpha)}{h}} \frac{\Gamma(k+\alpha)}{\Gamma(k+1)} f(t-kh) \tag{3}
$$

where h is the time increment.

Laplace transform is a popular tool in the analysis of control system. The Laplace transform of RL derivative under zero initial condition is

$$
L\{D^\alpha x(t)\} = s^\alpha X(s) \tag{4}
$$

2 CNC-ABC algorithm

2.1 Behavior of honey bees foraging

Bees are classical social insects. The behavior of only one bee is simple, however, a colony of bees manifest complicated intelligent behavior. In the real bee colony, the honey bees work in cooperation with appropriate division of labour. A minimal model of honey bees that can form swarm intelligence contains three elements: food source, employed bees and unemployed bees and three behaviors: exploring food source, recruiting bees for the food source and abandoning the food source.

The value of the food source is decided by several factors such as the distance to the beehive, the nectar amount and ease of extracting. For the sake of simplicity, it is represented as profitability with a simple quantity.

Employed bees are associated with a particular food source which they are currently exploiting or are "employed" at. They share the information of the food source by dancing in the dance area with a certain probability. The dance′s property is proportional to the profitability of the food source.

Unemployed bees are composed of onlookers and scouts. Scouts explore new food source randomly. The number of scouts averaged over conditions is about $5 \sim 10\,\%$[27]. When onlookers see the dances of employed bees, they estimate the profitability of the food sources and decide which one they will choose. So, more profitable food source recruits more onlookers to gather honey.

The behavior of honey bees foraging is shown as Fig. 1. Assuming there are two discovered food sources: A and B. At the beginning, a potential forager has no knowledge about the food sources as an unemployed bee. There are two choices for the bee:

1) It becomes a scout and explores new food source spontaneously around the nest (S in Fig. 1).

2) It becomes a recruit after watching the dances and starts searching for the food source (R in Fig. 1).

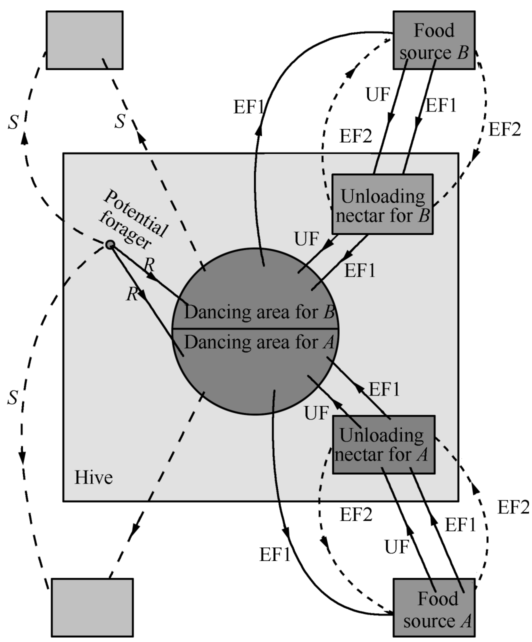

Fig. 1 Behavior of honey bee foraging

After locating the food source, the bee will start to exploit it and remember its situation. At this moment, the bee becomes an employed forager. When the bee returns to the hive carrying with nectar and unloads the nectar, it faces three choices:

1) It abandons the food source and becomes an onlooker (UF in Fig. 1);

2) It comes back to the same food source and continues to forage without recruiting the other bees (EF2 in Fig. 1);

3) It dances in the dancing area to recruit more bees before returning to the food source (EF1 in Fig. 1).

It is worthwhile to note that not all the bees start foraging simultaneously. The new bees begin foraging at a rate proportional to the eventual total number of bees and the number of present foraging.

2.2 Artificial bee colony (ABC) algorithm

In ABC, there are three groups of bees: employed bees, onlookers and scouts. The first half of the colony are employed bees and the other half are onlookers. The number of employed bees is equal to that of food sources. That is, each food source is exploited by only one employed bee. The employed bee whose food source has been exhausted becomes a scout.

In ABC algorithm, the position of a food source represents a possible solution to be optimized. The profitability of the food source corresponds to the quality (fitness) of the optimized solution. The main steps of the algorithm are described as follows.

Step 1. Initialize. The initial solutions (food sources) are generated randomly. Each solution $\boldsymbol{X}_i$ $(i = 1, 2, \cdots, N)$ (N is the number of solution) is a d-dimension real vector. $\boldsymbol{X}_i = (x_{i1}, x_{i2}, \cdots, x_{id})^{\rm T}$ represents the position of the ith food source.

Step 2. Each employed bee searches for the food source depending on the information in its memory and produces a modified position given by (5), then evaluates the profitability (fitness value). If the profitability is better than the previous one the employed bee will remember it and forget the old one.

$$x'_{ij} = x_{ij} + r_{ij}(x_{ij} - x_{kj}) \tag{5}$$

where $j \in 1, 2, \cdots, d$ and $k \in 1, 2, \cdots, N$ are randomly chosen indexes, and k is different from i. r_{ij} is a real random number in the range $[-1, 1]$.

Step 3. Onlooker bee selects food source according to the probability. The probability value $prob(i)$ is calculated by (6).

$$prob(i) = \frac{fit(i)}{\sum\limits_{i=1}^{N} fit(i)} \tag{6}$$

When an onlooker decides to select a food source, it starts searching for the food source and produces a modified position by (5), then evaluates the profitability and remember the better position. In (6), $fit(i)$ is the fitness value of the ith food source $\boldsymbol{X}_i$.

Step 4. Scout bee explores new food source. If a food source has been exhausted after a number of trials (limit), it will be abandoned. The scout will explore a new food source randomly according to (7).

$$x_{ij} = x_{\min,j} + (x_{\max,j} - x_{\min,j}) \times \text{rand}(0, 1) \tag{7}$$

It is important to note that only one of the employed bees can change to scout, which is controlled by "limit". So "limit" is an important control parameter. If a solution cannot be further improved by "limit" times of trials, which means the solution will get into local optimum and it will be abandoned by its employed bee and the employed bee becomes a scout.

In ABC algorithm, employed bees and onlookers carry out exploitation while scouts control the exploration process, which make ABC algorithm get balance between exploitation and exploration.

2.3 Cyclic exchange neighborhood

Cyclic exchange is one of the multi-exchange neighborhood structures and path exchange is another one. In this paper, cyclic exchange is mainly considered.

Let $J = \{j_1, j_2, \cdots, j_n\}$ be a set of n elements. The collection $S = \{S_1, S_2, \cdots, S_k\}$ defines a k-partition of J, where $k \leq n$ and each subset S_j is non-empty, the subsets are pair-wise disjoint, and their union is S. For any subset S of J, let $d[S]$ denote the cost of S. Then the set partitioning problem is to find a partition of J into at most k subsets so as to minimize $\Sigma_k d[S_k]$.

Let $\{S_1, S_2, \cdots, S_k\}$ be any feasible partition, a cyclic exchange is that element sequence $j_1 - j_2 - \cdots - j_k$ performs changes as follows: element j_1 moves from the subset S_1 to S_2, j_2 moves from S_2 to S_3, and so on until j_k moves from S_k to S_1. Finally, the move process of elements forms a cyclic structure and such a move process is called cyclic exchange. The illustration of the cyclic exchange structure is shown as Fig. 2.

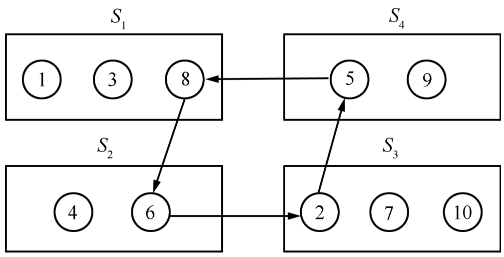

Fig. 2 Illustration of the cyclic exchange neighborhood

Given S, the cyclic exchange neighborhood is the set of all the assignments of solutions which can be obtained from S by performing cyclic exchange.

2.4 CNC-ABC algorithm

ABC can be viewed as a neighborhood search algorithm. Employed bees and onlookers produce new solution by searching the neighborhood of the current solution repeatedly. The search is limited to the neighborhood of the current solution. In ABC algorithm, the neighborhood is small which leads to slow convergence. On the other hand, when the algorithm gets into local optimum, one scout will explore a substitute randomly using (7). Owing to the fact that only one scout explores for new solution and the search is random, it is not easy to get out of the local optimum. The proposed CNC-ABC algorithm has the potential to solve these problems.

2.4.1 Cyclic exchange neighborhood search for employed bees

In CNC-ABC algorithm, the solution sequence $S = \{\boldsymbol{X}_1, \boldsymbol{X}_2, \boldsymbol{X}_3, \cdots, \boldsymbol{X}_N\}$ is seemed as the simplest cyclic-exchange, that is, each subset only has one solution, and $\boldsymbol{X}_q \neq \boldsymbol{X}_k$ ($q \neq k$, and $q, k \in \{1, 2, \cdots, N\}$). The cyclic-exchange performs the following exchanges: the solution $\boldsymbol{X}_1$ moves to $\boldsymbol{X}_2$, solution $\boldsymbol{X}_2$ moves to $\boldsymbol{X}_3$, and so on until solution $\boldsymbol{X}_{N-1}$ moves to $\boldsymbol{X}_N$ and solution $\boldsymbol{X}_N$ moves to $\boldsymbol{X}_1$. The moving process of the solutions is shown in Fig. 3. The exchange can be expressed as (8).

$$x'_{ij} = \begin{cases} x_{Nj} + r_{ij}(x_{ij} - x_{kj}), & i = 1 \\ x_{i-1j} + r_{ij}(x_{ij} - x_{kj}), & 2 \leq i \leq N \end{cases} \tag{8}$$

where x_{i-1j} is the neighbor of solution x_{ij} and it is the better one chosen by previous repetition. The cost of a cyclic exchange is the change in optimal objective function value caused by the cyclic exchange. The cyclic exchange is valid when the cost is negative. The cost of exchanged cycle is $\Sigma_N d(x'_i)$ and the cost before exchanged is $\Sigma_N d(x_i)$, where $d(x_i)$ is the objective function value of x_i. In order to find negative cost cycle, we suppose the cost of each node is negative, that is

$$d(x'_{ij}) - d(x_{ij}) < 0 \tag{9}$$

so $\Sigma_{j=1}^{d}(d(x'_{ij}) - d(x_{ij})) < 0$, that is $d(x'_i) - d(x_i) < 0$. Then

$$\sum_N d(x'_i) - \sum_N d(x_i) < 0 \tag{10}$$

We can see from (10) that the cost of exchanged cycle is negative. Thus, the solution can be found only satisfying (9).

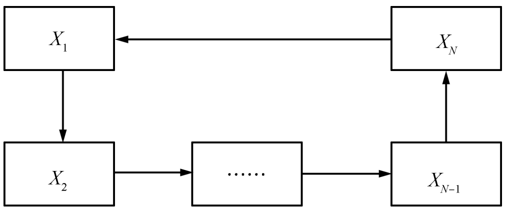

Fig. 3 Cyclic exchange of ABC algorithm

In CNC-ABC algorithm, (8) is only applied in the exploitation process of employed bees. Onlooker bees still keep local search using (5).

2.4.2 Chaos search of scout bee

In ABC algorithm, if a solution cannot be further improved through a number of trials limit, it will be instead by (7). In CNC-ABC algorithm, the chaotic search is applied. Suppose the needed improved solution is $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \cdots, x_{id})$, chaotic sequence (C_i) is produced using the well-known logistic equation:

$$C_{n+1} = 4C_n(1 - C_n) \tag{11}$$

where $0 < C_n < 1$, the initial value of the chaotic sequences is:

$$C^0_{ij} = \frac{x_{ij} - x^{\min}_{ij}}{x^{\max}_{ij} - x^{\min}_{ij}} \tag{12}$$

The needed improved solution is instead by

$$x'_{ij} = x_{ij} + C_{ij}(x^{\max}_{ij} - x^{\min}_{ij}) \tag{13}$$

The pseudocode of chaos search of scout is:

Step 1. T = maximum number of chaotic iterations; t = 0;

Step 2. Generate the initial chaotic sequence value by (12);

Repeat

Step 3. Produce the chaotic sequences by (11);

Step 4. Produce new solution using (13) and calculate its fitness, then select the better solution;

Step 5. $t = t + 1$;

Until $(t > T)$.

In CNC-ABC algorithm, employed bees, onlookers, and scouts use different strategies in the searching process. Employed bees search for the solutions in a larger space by applying cyclic exchange, which can enlarge the neighborhood of the solutions by performing multi-exchange neighborhood between different solutions. Onlookers produce new solutions by local search that is the same as basic ABC algorithm. The larger space searching of employed bees and local search of onlookers can find better solution quickly and speed up convergence. In the process of scouts exploring, sequences generated from logistic chaos system replace the random number of ABC algorithm. The traversal of chaos makes the solution get out of local optimum quickly, so that, the convergence of the algorithm gets improved.

3 Optimum FOPID controller design using CNC-ABC algorithm for AVR system

3.1 FOPID controller

The FOPID controller has five parameters: the proportional gain K_P, the integral gain K_I, the differential gain K_D, the integral order λ, and the differential order μ. The transfer function of $PI^{\lambda}D^{\mu}$ controller is given by

$$G_c(s) = K_P + K_I s^{-\lambda} + K_D s^{\mu}, \quad \lambda, \mu > 0 \tag{14}$$

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

$$u(t) = K_P e(t) + K_I D^{-\lambda} e(t) + K_D D^{\mu} e(t) \tag{15}$$

If $\lambda = 1$ and $\mu = 1$, the classical PID controller is gained, and if $\lambda = 1$ and $\mu = 0$, we get a PI controller, and if $\lambda = 0$ and $\mu = 1$, a PD controller, and if $\lambda = 0$ and $\mu = 0$ a proportional gain controller. These classical types of PID controllers are the special cases of the FOPID controller. The FOPID controller generalizes the conventional integer order PID controller.

3.2 Automatic voltage regulator (AVR) system

In general, an AVR system consists of four components: amplifier, exciter, generator, and sensor. To analyze dynamic character of AVR, a linearized model is considered. Their transfer functions are represented as follows[28].

1) Amplifier model. The amplifier model is given as

$$G_A(s) = \frac{K_A}{1 + \tau_A s} \tag{16}$$

where the value of K_A is in the range of [10, 400], and the time constant τ_A ranges from 0.02 s to 0.1 s usually.

2) Exciter model. The transfer function of an exciter is given by

$$G_E(s) = \frac{K_E}{1+\tau_E s} \tag{17}$$

where the typical value of the gain: K_E is in the range of [1, 400], and the time constant τ_E often ranges from 0.4 s to 1.0 s.

3) Generator model. The transfer function can be modeled by a gain K_G and a time constant τ_G

$$G_G(s) = \frac{K_G}{1+\tau_G s} \tag{18}$$

where K_G is in the range of [0.7, 1.0], and τ_G ranges from 1.0 s to 2.0 s.

4) Sensor mode. The transfer function of sensor circuit is modeled by

$$G_S = \frac{K_S}{1+\tau_S s} \tag{19}$$

where τ_S ranges from 0.001 s to 0.06 s.

3.3 Design of FOPID controller using CNC-ABC algorithm for AVR system

3.3.1 Performance criterion

In the design of controllers, there are several performance criterions. The often used criterions are integral of absolute error (IAE), integral of squared error (ISE) and integral of time-weighted-squared-error (ITSE)[29]. They are represented as follows:

$$IAE = \int_0^\infty |e(t)|\mathrm{d}t \tag{20}$$

$$ISE = \int_0^\infty e^2(t)\mathrm{d}t \tag{21}$$

$$ITSE = \int_0^\infty te^2(t)\mathrm{d}t \tag{22}$$

In this paper, the IAE criterion is applied. In general, it is necessary to select a reference model to get the error function $e(t)$. Smaller the error is, closer the controlled object is to the reference model. Here, the FOPID controller can be evaluated as

$$J(K_P, K_I, K_D, \lambda, \mu) = IAE = \int_0^\infty |y(t)^* - y(t)|\mathrm{d}t \tag{23}$$

where $y(t)^*$ is the real output response of the reference model. Thus, the optimal FOPID controller can be converted to an optimization problem as:

$$(K'_P, K'_I, K'_D, \lambda', \mu') = \arg\min J(K_P, K_I, K_D, \lambda, \mu) \tag{24}$$

where $(K'_P, K'_I, K'_D, \lambda', \mu')$ are the controlling parameters to be optimized.

In this paper, the Bode′s ideal reference model is applied as the reference model[30]. Bode has suggested an ideal open-loop transfer function in the design of feedback amplifiers, the transfer function is as follows

$$L(s) = \left(\frac{w_c}{s}\right)^\delta, \quad \delta \in \mathbf{R} \tag{25}$$

It is an interesting characteristic of Bode′s ideal transfer function to insert into the forward path that is insensitive to gain changes. Therefore, the closed-loop system with Bode′s ideal transfer function is often adopted as reference model[31−33].

3.3.2 Design of FOPID controller using CNC-ABC

The CNC-ABC algorithm is utilized to optimize the FOPID controller parameters K_P, K_I, K_D, λ and μ. The AVR system equipped with CNC-ABC-FOPID controller is shown in Fig. 4. A food source position represents a possible solution of the FOPID controller to be optimized which is represented by a vector with 5 elements, i.e $\boldsymbol{X} = (K_P, K_I, K_D, \lambda, \mu)$. The fitness of each solution is calculated by (23). The main design steps are expressed as follows.

Step 1. Initialization;

Step 2. Calculation of the fitness of each initial solution x_i;

Step 3. Let cycle =1;

Step 4. Production of new solutions $\boldsymbol{x}'_i$ by employed bees using (8) and calculation of the fitness by (23);

Step 5. Select the better one between $\boldsymbol{x}'_i$ and $\boldsymbol{x}_i$ applying greedy selection;

Step 6. Calculate the probability values $prob(i)$ by (6) of the solutions;

Step 7. Onlooker bees select the solution according to $prob(i)$, then search neighbor to produce new solutions by (5) and calculate the fitness by (23);

Step 8. Apply a greedy selection to select the better solution;

Step 9. Evaluate if a solution has to be improved, then scout carry out chaos search;

Step 10. Memorize the best solution found so far;

Step 11. cycle = cycle +1, check if cycle > MCN, terminate the algorithm; otherwise, go to Step 4.

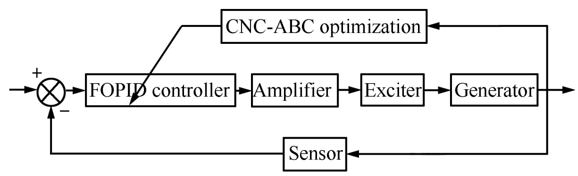

Fig. 4 Block diagram of AVR with CNC-ABC-FOPID controller

4 Numerical results

4.1 System parameters

The parameters of AVR system are set as in [30], i.e., $K_A = 10$, $\tau_A = 0.1$, $K_E = 1$, $\tau_E = 0.4$, $K_G = 1$, $\tau_G = 1$, $K_S = 1$, $\tau_s = 0.01$.

FOPID controller parameters: $0.001 \le K_P \le 15$, $0.001 \le K_I \le 10.0$, $0.001 \le K_D \le 10.0$, $0.001 \le \lambda \le 3.0$, $0.001 \le \mu \le 3.0$. We set $w_c = 10$ and $\delta = 1$ in Bode′s real reference model.

ABC and CNC-ABC algorithm parameters: the population size N is 40, the parameter limit is set to 40, the maximum cycle number MCN is 200.

To examine its performance, the proposed CNC-ABC is compared with ABC, PSO and GA algorithm. CNC-

ABC, ABC, PSO and GA are all used as optimizer for tuning the parameters of FOPID controller. The resulting controllers are called as CNC-ABC-FOPID, ABC-FOPID, PSO-FOPID and GA-FOPID. The parameters of PSO algorithm are set as: the population size is 40, $w_{\max} = 0.9$ and $w_{\min} = 0.4$, the constants $c_1 = c_2 = 2$, the maximum iteration is also 200. The GA algorithm is implemented using GAOT toolbox from http://www.ise.ncsu.edu/mirage/GAToolBox/gaot/ with parameters as follows: the selection operation is tournament, the crossover method is arithmetic with the probability 0.8 and the mutation method is uniform with the probability 0.2. The population and maximum iteration are the same as the PSO algorithm.

4.2 Performance of CNC-ABC FOPID controller

In this section, the FOPID controller for AVR system is examined.

In experiments, CNC-ABC, ABC, PSO and GA all run 20 times independently. The statistical results of FOPID controller′s parameters are listed in Table 1. The parameters include the best, the worst, mean and standard deviation (std), and overshoot M_P, the steady error E_{ss}, rise time t_r and settling time t_s in time domain.

Fig. 5 shows the terminal voltage step response of Bode′s real reference model and AVR system controlled by CNC-ABC-FOPID controller. It can be seen from Table 1 and Fig. 5 that the step response of the AVR system equipped with CNC-ABC-FOPID controller has small overshoot, short settling time, but it does not adapt to Bode′s real reference model fully.

The proposed CNC-ABC-FOPID is compared with ABC-FOPID and other FOPID controllers to demonstrate its advantage. Fig. 6 shows terminal voltage step response of AVR system controlled by the best CNC-ABC-FOPID and ABC-FOPID controller. We can see from Table 1 and Fig. 6 that CNC-ABC-FOPID controller has small overshoot, short settling time. The terminal voltage step responses of the AVR system with GA and PSO-FOPID controllers are shown in Fig. 7. It shows that the proposed CNC-ABC FOPID controller is the best and the PSO-FOPID controller is near to it, though Table 1 tells us that the PSO-FOPID controller is not stable.

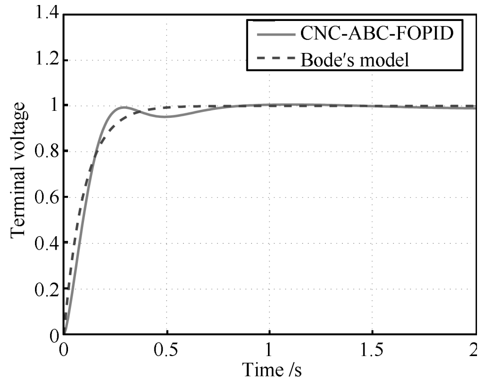

Fig. 5 The terminal voltage step response of the AVR system with CNC-ABC-FOPID controller and Bode′s model

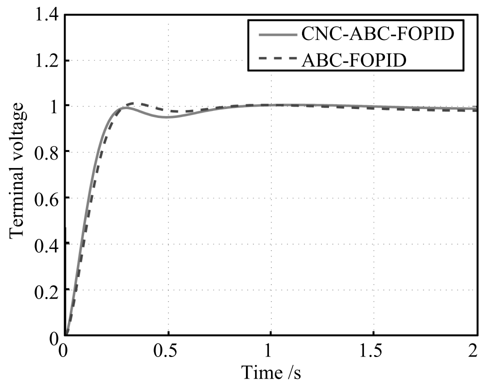

Fig. 6 The terminal voltage step response of the AVR system with CNC-ABC-FOPID and ABC-FOPID controllers

Table 1 Statistical results of the FOPID controller parameters

Algorithm		J	K_P	K_I	K_D	λ	μ	M_p	Ess	t_r	t_s
CNC-	Best	5.142	1.9566	0.4893	0.2345	1.5484	1.4317	0.0104	0.0106	0.1961	0.1991
ABC-	Worst	5.144	1.9699	0.4947	0.2362	1.5532	1.435	0.0111	0.0114	0.1779	0.2252
FOPID	Mean	5.143	1.9605	0.4922	0.2355	1.5508	1.4331	0.0109	0.0113	0.1816	0.2252
	Std	7.6E−5	0.0042	0.0014	0.00049	0.0014	0.00092				
	Best	5.1453	1.8749	0.452	0.2303	1.3653	1.3828	0.0129	0.0188	0.1651	0.2135
ABC-	Worst	5.8075	2.2835	0.6853	0.3214	1.6297	1.447	0.0152	0.0045	0.1472	0.5645
FOPID	Mean	5.3023	2.0204	0.5295	0.2497	1.5170	1.4202	0.0138	0.0092	0.1563	0.2262
	Std	0.20	0.089	0.052	0.020	0.064	0.016				
	Best	6.1125	2.7323	0.8275	0.3554	1.2651	1.3349	0.1173	0.0097	0.1170	0.3608
GA-	Worst	9.3053	10.2854	3.4731	1.958	1.5742	1.4358	0.3439	0.0041	0.0343	0.1851
FOPID	Mean	8.6411	7.7527	2.3430	1.2935	1.4047	1.3761	0.2974	0.0072	0.0390	0.2724
	Std	8.6411	1.6858	0.6295	0.3816	0.0802	0.03314				
	Best	5.1442	1.9102	0.4851	0.2298	1.5373	1.4244	0.0107	0.0116	0.1832	0.2303
PSO-	Worst	10.144	15	3.6986	2.3062	0.2895	1.3424	0.4670	0.0057	0.0316	0.2727
FOPID	Mean	5.6218	2.5085	1.069	0.4084	1.4848	1.4256	0.0330	0.0164	0.1096	0.2712
	Std	1.4695	2.9724	1.9599	0.5460	0.2814	0.02185				

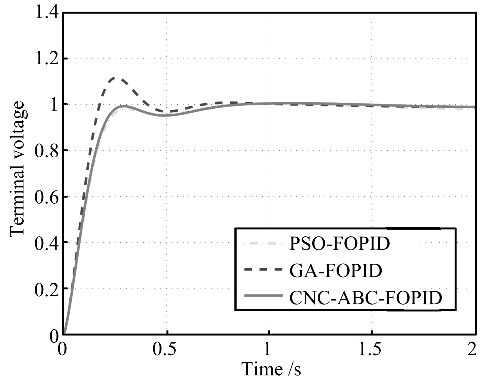

Fig. 7 The terminal voltage step response of the AVR system with CNC-ABC-FOPID, GA-FOPID and PSO-FOPID controllers

Finally, the CNC-ABC-FOPID controller is compared with PID controller. The PID controllers include CNC-ABC-PID controller, ABC-PID controller and GA-PID controller. The terminal voltage step responses of the best controllers are showed in Fig. 8. We can see that CNC-ABC-FOPID controller is the best one that has little overshoot, short settling time, that is, the proposed CNC-ABC-FOPID controller has better performance than PID controller.

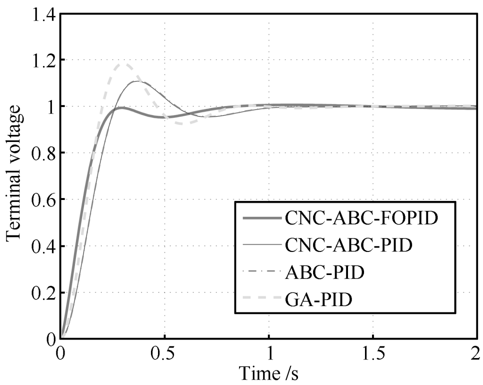

Fig. 8 The terminal voltage step response of the AVR system with CNC-ABC-FOPID, CNC-ABC-PID, ABC-PID and GA-PID controllers

5 Conclusion

In this paper, a novel method for designing FOPID controller is presented using an artificial bee colony algorithm which is based on cyclic exchange neighborhood and chaos. The proposed CNC-ABC algorithm has better convergence and performance than ABC and it is used to optimize the parameters of the FOPID controller. The proposed CNC-ABC-FOPID controller has been applied to AVR system. Simulation experiments show that the proposed CNC-ABC-FOPID controller has better performance than ABC-FOPID controller and other FOPID, PID controllers.

References

1 O'Dwyer A. *Handbook of PI and PID Controller Tuning Rules*. London: Imperial College Press, 2009

2 Podlubny I. Fractional-order systems and $PI^{\lambda}D^{\mu}$ controllers. *IEEE Transactions on Automatic Control*, 1999, **4**(1): 208−214

3 Valério D, Da Costa J S. Tuning of fractional PID controllers with Ziegler-Nichols-type rules. *Signal Processing*, 2006, **86**(10): 2771−2784

4 Cervera J, Banos A, Monje C A, Vinagre B M. Tuning of fractional PID controllers by using QFT. In: Proceedings of the 2006 IEEE Conference on Industrial Electronics. Paris, France: IEEE, 2006. 5402−5407

5 Pan I, Das S. Chaotic multi-objective optimization based design of fractional order $PI^{\lambda}D^{\mu}$ controller in AVR system. *International Journal of Electrical Power and Energy Systems*, 2012, **43**(1): 393−407

6 Biswas A, Das S, Abraham A, Dasgupta S. Design of fractional-order $PI^{\lambda}D^{\mu}$ controllers with an improved differential evolution. *Engineering Applications of Artificial Intelligence*, 2009, **22**(2): 343−350

7 Zamani M, Karimi-Ghartemani M, Sadati N, Parniani M. Design of a fractional order PID controller for an AVR using particle swarm optimization. *Control Engineering Practice*, 2009, **17**(12): 1380−1387

8 Tang Y G, Cui M Y, Hua C C, Li L X, Yang Y X. Optimum design of fractional order $PI^{\lambda}D^{\mu}$ controller for AVR system using chaotic ant swarm. *Expert Systems with Applications*, 2012, **39**(8): 6887−6896

9 Karaboga D. An Idea Based on Honey Bee Swarm for Numerical Optimization. Technical Report TR06, Erciyes University Press, Erciyes, 2005

10 Akay B, Karaboga D. A modified artificial bee colony algorithm for real-parameter optimization. *Information Sciences*, 2012, **192**: 120−142

11 Karaboga D, Akay B. A modified artificial bee colony (ABC) algorithm for constrained optimization problems. *Applied Soft Computing*, 2011, **11**(3): 3021−3031

12 Sonmez M. Artificial bee colony algorithm for optimization of truss structures. *Applied Soft Computing*, 2011, **11**(2): 2406−2418

13 Gao W F, Liu S Y. A modified artificial bee colony algorithm. *Computers and Operations Research*, 2012, **39**(3): 687−697

14 Rajasekhar A, Das S, Suganthan P N. Design of fractional order controller for a servohydraulic positioning system with micro artificial bee colony algorithm. In: Proceedings of the 2012 IEEE Congress on Evolutionary Computation. Brisbane, Australia: IEEE, 2012. 1−8

15 Karaboga D, Ozturk C. A novel clustering approach: artificial bee colony (ABC) algorithm. *Applied Soft Computing*, 2011, **11**(1): 652−657

16 Pan Q K, Fatih Tasgetiren M, Suganthan P N, Chua T J. A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem. *Information Sciences*, 2011, **181**(12): 2455−2468

17 Bao Li, Zeng Jian-Hu. Self-adapting search space chaos-artificial bee colony algorithm. *Application Research of Computers*, 2010, **27**(4): 1330−1334 (in Chinese)

18 Lee W P, Cai W T. A novel artificial bee colony algorithm with diversity strategy. In: Proceedings of the 2011 International Conference on Natural Computation. Shanghai, China: IEEE, 2011. 1441−1444

19 Gao W F, Liu S Y. Improved artificial bee colony algorithm for global optimization. *Information Processing Letters*, 2011, **111**(17): 871−882

20 Glover F. Ejection chains, reference structures and alternating path methods for traveling salesman problems. *Discrete Applied Mathematics*, 1996, **65**(1−3): 223−253

21 Rego C. Relaxed tours and path ejections for the traveling salesman problem. *European Journal of Operational Research*, 1998, **106**(2): 522−538

22 Ahuja R K, Orlin J B, Sharma D. Multi-exchange neighborhood structures for the capacitated minimum spanning tree problem. *Mathematical Programming*, 2001, **91**(1): 71−97

23 Tang L X, Luo J X. A new ILS algorithm for parallel machine scheduling problems. *Journal of Intelligent Manufacturing*, 2006, **17**(5): 609−619

24 Frangioni A, Necciari E, Scutellá M G. A multi-exchange neighborhood for minimum makespan parallel machine scheduling problems. *Journal of Combinatorial Optimization*, 2004, **8**(2): 195−220

25 Thompson P M, Psaraftis H N. Cyclic transfer algorithm for multivehicle routing and scheduling problems. *Operations Research*, 1993, **41**(5): 935−946

26 Podlubny I. *Fractional Differential Equations*. San Diego: Academic Press, 1999

27 Seeley T. *The Wisdom of the Hive: the Social Physiology of Honey Bee Colonies*. Harvard: Harvard University Press, 1996

28 Gaing Z L. A particle swarm optimization approach for optimum design of PID controller in AVR system. *IEEE Transactions on Energy Conversion*, 2004, **19**(2): 384−391

29 Krohling R A, Rey J P. Design of optimal disturbance rejection PID controllers using genetic algorithms. *IEEE Transactions on Evolutionary Computation*, 2001, **5**(1): 78−82

30 Bode H W. *Network Analysis and Feedback Amplifier Design*. New Jersey: Van Nostrand Reinhold, 1956

31 Takyar M S, Georgiou T T. The fractional integrator as a control design element. In: Proceedings of the 46th IEEE Conference on Decision and Control. New Orleans, United States: IEEE, 2007. 239−244

32 Djouambi A, Charef A, Voda A. Fractional order controller based on bode′s ideal transfer function. *Control and Intelligent Systems*, 2010, **38**(2): 67−73

33 PetrášI. Fractional-order feedback control of a DC motor. *Journal of Electrical Engineering*, 2009, **60**(3): 117−128

ZHANG Dong-Li Ph. D. candidate at the Institute of Electrical Engineering, Yanshan University. Her research interest covers swarm intelligence and intelligent control theory. Corresponding author of this paper. E-mail: zdl@ysu.edu.cn

TANG Ying-Gan Professor in the Department of Automation, Yanshan University. His research interest covers image processing, system modeling and swarm intelligence optimization algorithm. E-mail: ygtang@ysu.edu.cn

GUAN Xin-Ping Professor in the Department of Automation, Shanghai Jiao Tong University. His research interest covers wireless sensor networks, congestion control of networks, robust control and intelligent control for nonlinear systems.
E-mail: xpguan@ysu.edu.cn