Chaos, Solitons and Fractals 42 (2009) 792–800

Contents lists available at ScienceDirect

Chaos, Solitons and Fractals

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

CAS algorithm-based optimum design of PID controller in AVR system

Hui Zhu *, Lixiang Li, Ying Zhao, Yu Guo, Yixian Yang

Information Security Center, State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, P.O. Box 145, Beijing 100876, China
Key Laboratory of Network and Information Attack and Defence Technology of Ministry of Education, Beijing University of Posts and Telecommunications, Beijing 100876, China
National Engineering Laboratory for Disaster Backup and Recovery, Beijing 100876, China

ARTICLE INFO

Article history:
Accepted 9 February 2009

Communicated by Prof. Ji-Huan He

ABSTRACT

This paper presents a novel design method for determining the optimal PID controller parameters of an automatic voltage regulator (AVR) system using the chaotic ant swarm (CAS) algorithm. In the tuning process of parameters, the CAS algorithm is iterated to give the optimal parameters of the PID controller based on the fitness theory, where the position vector of each ant in the CAS algorithm corresponds to the parameter vector of the PID controller. The proposed CAS-PID controllers can ensure better control system performance with respect to the reference input in comparison with GA-PID controllers. Numerical simulations are provided to verify the effectiveness and feasibility of PID controller based on CAS algorithm.

1. Introduction

PID control is the most extensive control law in engineering. Nowadays more than 90% of control loops in industry are PID [1]. Despite numerous control advanced methods, such as adaptive control, fuzzy control, and neural network control, have been introduced to the motion control field [2], the PID and its variations(P,PI,PD) still are widely applied in the motion control because of their strong robustness to meet the global change of industry process and their simple structures which consist of only three parameters, that is, proportional parameter, integral parameter, and derivative parameter.

Over the past half century, researchers have sought the key techniques for PID tuning. Plenty of methods have been proposed for the tuning of PID controllers, such as Ziegler and Nichols methods [3], frequency domain method, time domain method, and intellectual optimization methods [4–6]. For the intellectual optimization methods, PID parameters are obtained by numerical optimization for a weighted objective in the time domain. Intellectual optimization methods, such as evolutionary algorithm, genetic algorithm [7,8], and particle swarm algorithm [9] have been applied to improve performances of the PID controller.

The chaotic ant swarm (CAS) algorithm is inspired by the chaotic and self-organization behavior of ants. It belongs to the categories of swarm intelligence methods and chaos optimization approaches. The CAS algorithm has good ability of global search. One may now find applications of this algorithms in different areas such as parameters estimation of chaotic systems [10] and economic dispatch of power system [11].

In this paper a design method which optimizes PID parameters of AVR system via the CAS algorithm is presented. Then the corresponding CAS-PID controller is built. A practical high-order AVR system with a PID controller is adopted to test the performance of the proposed CAS-PID controller. In order to demonstrate the performance of the proposed method is

* Corresponding author. Address: Information Security Center, State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, P.O. Box 145, Beijing 100876, China.
E-mail address: zhuhui05608@hotmail.com (H. Zhu).

0960-0779/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.chaos.2009.02.006

excellent, some comparison results are given between the CAS-PID and GA-PID controller. Simultaneously, a suitable fitness function was proposed to evaluate the performance of different PID controllers.

2. PID controller

2.1. Standard structures of PID controllers

The transfer function of a PID controller is often expressed in the following form [12]

$$G_{PID}(s) = \frac{U(s)}{E(s)} = K_p\left(1 + \frac{1}{T_i s} + T_d s\right), \tag{1}$$

where $U(s)$ is the control signal acting on the error signal $E(s), K_p$ is the proportional gain, T_i is the integral time constant, T_d is the derivative time constant, and s is the argument of the Laplace transform.

The functionalities of PID controller include:

- the proportional term provides an overall control action proportional to the error signal through the allpass gain factor;
- the integral term reduces steady-state errors through low-frequency compensation;
- the derivative term improves transient response through high-frequency compensation.

A PID controller can be considered as an extreme form of a phase lead-lag compensator with one pole at the origin and the other at infinity. Similarly, its cousins, the PI and the PD controllers, can also be regarded as extreme forms of phase-lag and phase-lead compensators, respectively. For optimum performance, K_p, K_i (or T_i), and K_d (or T_d) must be tuned jointly.

2.2. Performance estimation of PID controller

In a PID design, controller parameters are usually tuned so that the closed-loop system meets the following five objectives:

(1) stability and stability robustness, usually measured in the frequency domain;
(2) transient response, including rise time, overshoot, and settling time;
(3) steady-state accuracy;
(4) disturbance attenuation and robustness against environmental uncertainty, often at steady-state;
(5) robustness against plant modeling uncertainty, usually measured in the frequency domain.

PID design methods can be grouped according to the underlying nature of a given control objective. Most methods target one objective or a weighted composite of the objectives listed above [12].

When the objective is given, we must select a criterion formula to evaluate the performance of different PID controllers. In general, the integrated absolute error (IAE), the integral of squared-error (ISE), and the integral of time multiplied by the absolute value of error (ITAE) are often employed in control system design. When these classical standards are applied into practical industry, good effects are often achieved except these which have high requirement for small overshoot. In order to obtain small overshoot, this paper employs the following performance criterion:

$$W(\rho) = (1-\rho)\int_0^{\infty} t|e(t)|dt + \rho M_p, \tag{2}$$

where M_p is the overshoot, ρ is the weighting factor ranging from 0 to 1. The performance criterion $W(\rho)$ can satisfy the designer requirements by adjusting the suitable value of ρ. We can set large ρ to reduce the overshoot. On the other hand, we can set small ρ to reduce the setting time. When $\rho = 0$, this performance criterion become the well-known ITAE.

3. Description of an AVR system

In a power station there is usually more than one generator connected to the same busbar and each has an individual automatic voltage regulator (AVR) [13]. The design objective for an AVR is to control the voltage at the terminals of the generator, namely achieve primary voltage control [14]. With the advancement in the design of fast acting AVRs as well as the increasing complexity of large interconnected power systems, oscillations may continue for an extended period and even instability may occur following some system disturbances. The control algorithms to overcome these problems are generally implemented using analog components [15]. A machine model chosen for power system dynamic studies depends not only on the nature of the problem, but also on the computational facilities and available control techniques. In order to describe the mathematical model more simpler, this paper linearizes the model of AVR system, which takes into account the major time constant and ignores the saturation or other nonlinearities.

A simple AVR system comprises four main components, namely amplifier, exciter, excitation voltage, generator, and sensor. To analyze dynamic performance of AVR, the reasonable transfer functions of these components are shown as follows [15–18]:

- Generator model
  A simplified first-order approximation of the machine is chosen here and is given by

$$G_G = \frac{K_G}{1+\tau_G s}, \tag{3}$$

where K_G is the gain and τ_G is the time constant. These constants are load dependent. K_G may vary between 0.1 and 1.0, and τ_G is between 1.0 and 2.0 s.
- Amplifier model
  The transfer function of amplifier is

$$G_A = \frac{K_A}{1+\tau_A s}, \tag{4}$$

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
  Exciters can also be very easily included in the design procedure. 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

$$G_E = \frac{K_E}{1+\tau_E s}, \tag{5}$$

where the typical value of K_E is in the range of [10, 400], and the time constant τ_E ranges from 0.5 to 1.0 s.
- Sensor model
  The sensor circuit, which rectifies, filters, and reduces the terminal voltage, is modeled by the following simple first-order transfer function

$$G_S = \frac{K_S}{1+\tau_S s}, \tag{6}$$

where τ_S ranges from of 0.001 to 0.06 s.

4. Chaotic ant swarm algorithm

In recent years, ant colonies, and more generally social insect societies, have always fascinated human beings [19–21]. The study of ant chaotic behavior and of their self-organizing capacities brings great interest from computer scientists to develop models of distributed organization which are useful to solve difficult optimization and distributed control problems. In the following, we give the detailed chaotic ant swarm algorithm based on the chaotic and self-organizing behaviors of ants [22].

Our model searches for optimum or near-optimum in the search space symbolized as R^L, the L-dimensional continuous space of real numbers. We consider a population of n ants. These ants are located in a search space S and they try to minimize a function $f : S \to R$. Each point s in S is a valid solution to the considered problem. In this letter, we only consider the search space to be a continuous space ($S = R^L$). The position of an ant i is the algebraic variable symbol $s_i = (z_{i1}, \ldots, z_{iL})$, where $i = 1, 2, \ldots, n$. Naturally each variable can be of any finite dimension.

In order to obtain the chaotic search initially, the chaotic system described by Solé et al. is introduced into our metaheuristic equation. Obviously during its motion, each individual ant is influenced by their current position, the best position so far by itself and by its neighbors and organization process of the swarm. The adjustment of the chaotic behavior of individual ant is achieved by the introduction of a successive decrement of organization variable $y_i(t)$ and leads the individual to moving to the new site that acquires the best fitness eventually. To achieve the information exchange of individuals and the movements to new site taken on the best fitness, we introduce $(p_{id}(t-1) - z_{id}(t-1))$. The term p_{id} is selected based on the fitness theory which is very widely developed in optimization theory such as genetic algorithm and tabu search, and so on. Thus, we obtain the following detailed dynamical optimization system of chaotic ant swarm:

$$\begin{cases} y_i(t) = y_i(t-1)^{(1+r_i)} \\ z_{id}(t) = z_{id}(t-1)e^{(1-e^{-ay_i(t)})(3-\psi_d z_{id}(t-1))} \\ \qquad\quad +(p_{id}(t-1) - z_{id}(t-1))e^{-2ay_i(t)+b} \end{cases}, \tag{7}$$

where a is a sufficiently large positive constant and can be selected as $a = 200$, b is a constant and $0 \leqslant b \leqslant \frac{2}{3}$, ψ_d determines the selection of the search range of dth element of variable in search space, $r_i > 0$ is a positive constant less than one and is termed by us as organization factor of ant i, $y_i(0) = 0.999$, $z_{id}(t)$ is the current state of the dth dimension of the individual ant

i, where $d = 1, 2, \ldots, L$. $p_{id}(t-1)$ is the best position found by ith ant and its neighbors within $t-1$ steps, $y_i(t)$ is the current state of the organization variable, t means the current time step, and $t-1$ is the previous step.

Via numerous simulations, we find that the above model of chaotic ant swarm (7) searches for optima in constrained positive or negative intervals. That is to say, if $\psi_d > 0$, Eq. (7) can be used to realize the search process in the intervals in which all $z_{id} \geqslant 0$. And if $\psi_d < 0$, Eq. (7) can be used to realize the search process in the intervals in which all $z_{id} \leqslant 0$. When all the optima are located in positive intervals (or negative intervals), Eq. (7) is effective to solve the numerical optimization problems. However, the elements of the optima can be located in all the ranges of real-numbered space. In order to solve the problem of search regions, we give the following version of CAS model which is more better than Eq. (7)

$$\begin{cases} y_i(t) = y_i(t-1)^{(1+r_i)} \\ z_{id}(t) = (z_{id}(t-1) + V_i \times \frac{7.5}{\psi_d})e^{(1-e^{-ay_i(t)})(3-\psi_d z_{id}(t-1))} \\ +(p_{id}(t-1) - z_{id}(t-1))e^{-2ay_i(t)+b} - V_i \times \frac{7.5}{\psi_d} \end{cases}, \tag{8}$$

where $0 \leqslant V_i \leqslant 1$ determines the search region of ant i and offers the advantage that ants could search diverse regions of the problem space. If $V_i = \frac{1}{2}$, it means the chaotic attractor of ant i moves a half to the negative orientation compare to the chaotic attractor of Eq. (7). The values V_i should be suitably selected according to the concrete optimization problems. We call the Eq. (8) the general algorithmic model of chaotic ant swarm. In this model we can select the initial position of individual ant as $z_{id}(0) = 7.5/\psi_d \times (1 - V_i) \times rand()$, where $\psi_d > 0$ [22].

Since the CAS algorithm has good capability of global searching, we combine the CAS algorithm with the PID controller and use the CAS algorithm for the dynamic training of PID parameters. The CAS-PID controller is applied to achieve the nonlinear control of the dynamical system.

5. CAS-PID controller in AVR system

In this section, a PID controller using the CAS algorithm, which could be called the CAS-PID controller, was developed to improve the step transient response of an AVR system. The block diagram of an AVR system with a CAS-PID controller is shown in Fig. 1. The CAS algorithm was mainly utilized to determine three optimal controller parameters K_p, K_i, and K_d, such that the controlled system could obtain a good step response output.

5.1. Individual string definition

To apply the CAS algorithm for searching the parameters of the PID controller, we use the "individual" to replace the "ant" and use the "population" to replace the "swarm" in this paper. We defined three controller parameters K_p, K_i, and K_d, to compose an individual vector K by $K = [K_p, K_i, K_d]$. Hence, there are three members in an individual. These members are assigned as real values. If there are n individuals in a population, then the dimension of a population is $n \times 3$.

5.2. CAS-PID controller in AVR system

Now we discuss how to attain suitable values of K_p, K_i, and K_d using the CAS algorithm. In the simulation, we consider the parameters K_p, K_i, and K_d as a parameter vector of the selected AVR system. Simultaneously, the performance criterion (2) is used as the evaluation function to measure the evaluation value of each individual in population. The design process for CAS-PID controller in AVR system is shown as follows:

1. Correspond the parameter vector of the selected AVR system to the position vector of each ant. Specify the lower and upper bounds of the three controller parameters. Ascertain a and b, V_i, organization factor r_i, ant swarm size n.
2. Initialization: Set $z_i(0)(i = 1, 2, \ldots, n)$ for n ants randomly in the definition domain. Specify the values of parameter $y_0(t)$ and interval ψ_{id} simultaneously.
3. Eq. (8) is iterated to give the optimal parameters of PID controller in AVR system based on the evaluation function. Calculate the values of each initial individual K of the population.

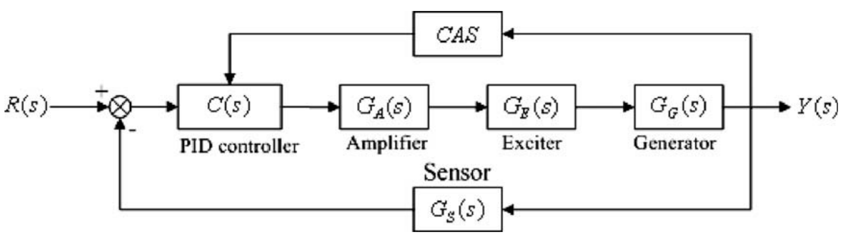

Fig. 1. Block diagram of an AVR system with a CAS-PID controller.

4. Let $t = t + 1$ and goto step 3 until the evaluation function is smaller than the specified precision or the iteration satisfies a specified maximal iterative step.

6. Numerical simulations

6.1. AVR system parameters

To verify the efficiency of the CAS-PID controller, a practical high-order AVR system was tested. The block diagram of the AVR system with a CAS-PID controller and the system parameters values are shown in Fig. 2. The three controller parameters K_p, K_i, K_d all range from 0 to 2.5. Fig. 3 shows the original terminal voltage step response of the AVR system without a CAS-PID controller. Through simulation, we found that $M_P = 72.03\%, E_{SS} = 0.0769, t_r = 0.3511(s)$, and $t_s = 8.2662(s)$, where M_P, E_{SS}, t_r, and t_s represent overshoot, steady-state error, rise time, and settling time, respectively.

6.2. Performance of the CAS-PID controller

The following CAS parameters are used to searching the optimal PID controller parameters:

- population size $n = 20$, generation $T = 150, V_i = 0$;
- $r_i = \begin{cases} 0, & i = 1,2 \\ 0.5, & i = 3 \\ 0.09 + 0.01 * rand, & others \end{cases}$, where *rand* is a random number ranging from 0 to 1;
- $b = 0.5$ and $\psi_d = 3$.

To verify the performance of CAS-PID controller, we execute 10 trials for $\rho = 0.1, 0.3, 0.5, 0.7, 0.9$, respectively. When $\rho = 0.7$, the results of 10 trials are shown in Table 1. When ρ equals other values, the statistic results are shown in Table 2, where $\overline{W}, D(W), \overline{E}_{SS}, D(E_{SS}), \bar{t}_r, D(t_r), \bar{t}_s, D(t_s), \overline{M}_P$, and $D(M_P)$ represent average evaluation value, variance of evaluation value, average steady-state error, variance of steady-state error, average rise time, variance of rise time, average settling time, variance of settling time, average overshoot, and variance of overshoot, respectively. We can see from Table 2 that the

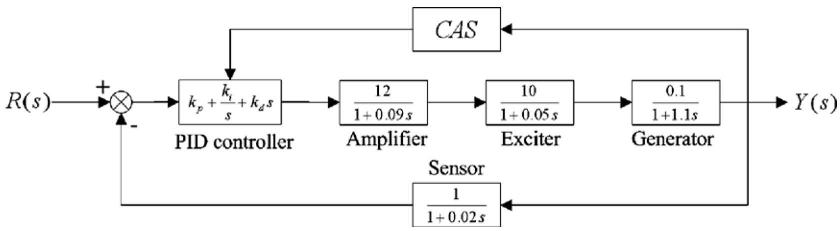

Fig. 2. Block diagram of an AVR system with a CAS-PID controller.

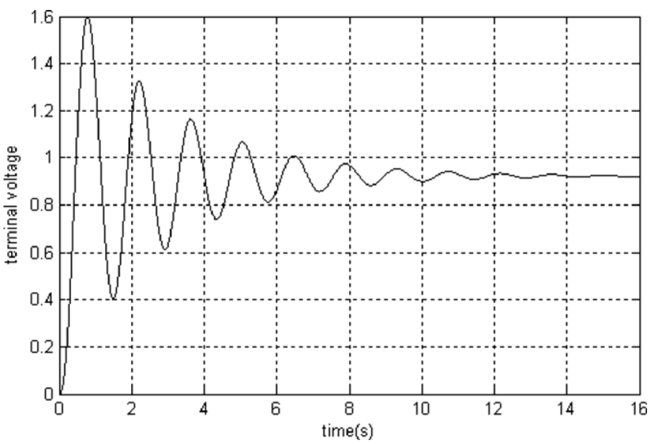

Fig. 3. Terminal voltage step response of an AVR system without PID controller.

Table 1
Solutions using CAS-PID controller when $\rho = 0.7$.

Solution	K_p	K_i	K_d	W	E_{SS}	t_r	t_s	M_P (%)
1	0.6885	0.4625	0.3133	0.0426	0	0.2232	0.3079	4.4803
2	0.5286	0.3440	0.2048	0.0147	0	0.3378	0.4676	0.3990
3	0.6087	0.2705	0.2785	0.0288	0	0.2633	1.1688	0
4	0.5613	0.3670	0.2326	0.0146	0	0.3028	0.4434	0.4653
5	0.5587	0.3584	0.3984	0.0406	0	0.1991	0.3737	1.2019
6	0.5880	0.4043	0.2697	0.0170	0	0.2667	0.3978	0.8023
7	0.5662	0.3968	0.3414	0.0204	0	0.2267	1.0159	1.0963
8	0.5964	0.4251	0.3189	0.0211	0	0.2348	0.9292	1.0629
9	0.5831	0.4130	0.3257	0.0195	0	0.2341	0.9757	1.0739
10	0.5818	0.4589	0.3487	0.0236	0	0.2209	0.9422	1.6518

Table 2
Statistical results of 10 trials when $\rho = 0.1, 0.3, 0.5, 0.7, 0.9$, respectively.

ρ	$\overline{W}$	$D(W)$	$\overline{E_{SS}}$	$D(E_{SS})$	$\overline{t_r}$	$D(t_r)$	$\overline{t_s}$	$D(t_s)$	$\overline{M_P}$ (%)	$D(M_P)$
0.1	0.0397	0.0000319	0	0	0.2050	0.001608	0.5116	0.01937	8.7781	0.002523
0.3	0.0411	0.0000511	0	0	0.2326	0.000667	0.6132	0.07084	2.6294	0.000171
0.5	0.0383	0.0001419	0	0	0.2257	0.001950	1.0729	0.72750	2.5015	0.000350
0.7	0.0243	0.0001012	0	0	0.2509	0.001792	0.7957	0.12776	1.2234	0.000153
0.9	0.0107	0.0000050	0	0	0.2221	0.000602	1.5520	0.79175	1.6088	0.000320

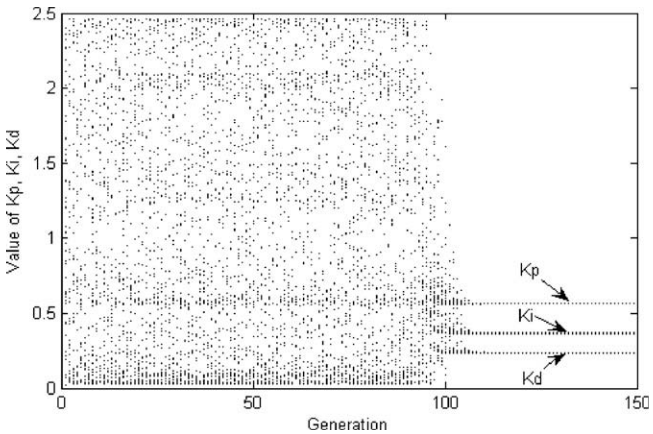

Fig. 4. Evolutionary process of K_p, K_i, K_d in the CAS-PID controller of the Solution 4 in Table 1.

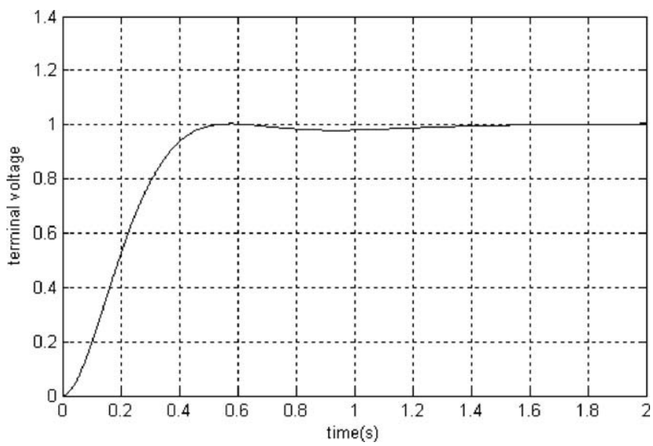

Fig. 5. Terminal voltage step response of an AVR system with the CAS-PID controller of the Solution 4 in Table 1.

overshoot is smaller with ρ being larger, just like what has been described in Section 2.2. And the settling time is in general smaller with ρ being smaller. In order to obtain good rise time, settling time, and overshoot, the best solution (i.e. Solution 4) in Table 1 is selected as the optimal PID controller parameters. Fig. 4 shows the corresponding evolutionary processes of K_p, K_i, K_d for the CAS-PID controller of the Solution 4. Fig. 5 shows terminal voltage step response of the AVR system of the Solution 4, i.e. $K_p = 0.5613, K_i = 0.3670, K_d = 0.2326$. As we can see from Figs. 4 and 5, excellent parameters of PID controller could be obtained with CAS methods.

6.3. Robustness test

To examine the robustness of the PID controller with respect to parameter uncertainties, we could assume the practical function of amplifier is as follows:

$$G_A(s) = \frac{12 + \Delta_1}{1 + (0.09 + \Delta_2)s}, \tag{9}$$

where $|\Delta_1| \leqslant 2$ and $|\Delta_2| \leqslant 0.01$. We use the best parameters in the Table 1 to test the robustness performance of the PID controller, i.e. $K_p = 0.5613, K_i = 0.3670, K_d = 0.2326$. Here we carry out four different tests. The uncertain parameters in these tests are shown in the Table 3. The terminal voltage unit step response is shown in Fig. 6. We can see from Fig. 6 that the system still has good performance in the presence of parameters uncertainty.

6.4. Comparison of CAS-PID and GA-PID controllers

In order to emphasize the advantages of the proposed CAS-PID controller, we also implemented the GA-PID controller derived from the real-coded GA method [23–25]. We have compared the characteristics of the two controllers by using the same evaluation function and individual definition in six different simulation examples. The following real-coded GA parameters have been used:

- the member of each individual is K_p, K_i, and K_d;
- population size = 20;
- crossover rate $P_C = 0.9$;
- mute rate $P_m = 0.033$.

Table 3
Four different values for uncertain parameters.

Test	Δ_1	Δ_2
1	2	0.01
2	2	−0.01
3	−2	0.01
4	−2	−0.01

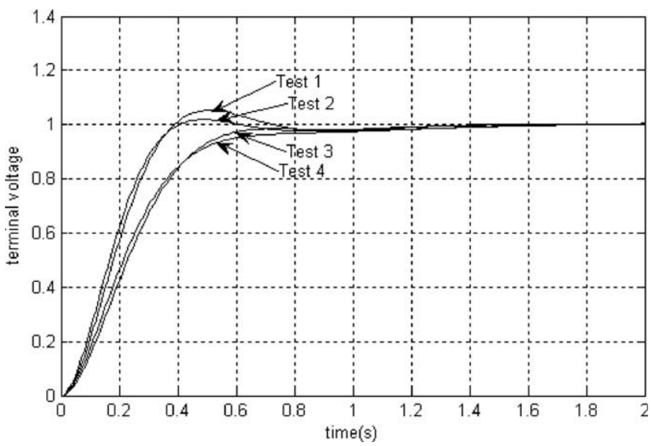

Fig. 6. Terminal voltage step response of an AVR system with PID controllers in the presence of parameters uncertainty.

Table 4
Comparison of the evaluation value between CAS-PID and GA-PID methods.

Ex.	ρ	T	Controller	K_p	K_i	K_d	W	M_P (%)	E_{SS}	t_s	t_r
1	0.3	50	GA-PID	0.7905	0.6923	0.4338	0.0620	10.1143	0	0.7761	0.1707
			CAS-PID	0.6568	0.4821	0.3210	0.0385	3.4319	0	0.3166	0.2244
2	0.3	100	GA-PID	0.7144	0.5344	0.3715	0.0466	6.1347	0	0.8101	0.1967
			CAS-PID	0.5575	0.3673	0.2284	0.0291	0.5042	0	0.4407	0.3073
3	0.5	50	GA-PID	0.4520	0.6322	0.4889	0.0655	4.9329	0	0.9858	0.1845
			CAS-PID	0.5942	0.5135	0.3520	0.0350	2.1365	0	0.8624	0.2174
4	0.5	100	GA-PID	0.5684	0.4940	0.3698	0.0443	2.2089	0	0.9477	0.2103
			CAS-PID	0.5942	0.5135	0.3520	0.0257	0.9492	0	0.3624	0.2516
5	0.7	50	GA-PID	0.3850	0.5308	0.4829	0.0402	5.1852	0	3.4408	0.1727
			CAS-PID	0.5235	0.4458	0.3940	0.0264	2.2187	0	1.0931	0.2029
6	0.7	100	GA-PID	0.4748	0.4841	0.4304	0.0320	3.3026	0	1.1431	0.2004
			CAS-PID	0.5874	0.3882	0.2810	0.0160	0.6239	0	0.3784	0.2596

CAS-PID and GA-PID controllers and their performance evaluation criteria in the time domain were implemented by the control system toolbox of Matlab, and executed on a Mobile Intel Pentium IV 1.8 GHz personal computer with 512-MB RAM.

The simulation results showing the best solution are summarized in Table 4. As can be seen, both controllers could give good PID controller parameters in each simulation example, providing good terminal voltage step response of the AVR system. Table 4 also shows the four performance criteria in the time domain of each example. As revealed by the above four performance criteria, the CAS-PID controller has better performance than the GA-PID controller.

7. Conclusions

This paper presents a novel design method for determining the optimal PID controller parameters of an practical AVR system using the CAS method. In the tuning process of the parameters, the CAS algorithm is iterated to give the optimal parameters of the PID controller based on the fitness theory. The simulation results showed that the proposed controller could perform an efficient search for the optimal PID controller parameters. In addition, in order to compare the CAS method with the GA method, many simulation examples for their terminal voltage step responses are performed. The results show that the proposed CAS method overcomes shortcoming of premature convergence of GA method and could obtain better solution and computation efficiency. Simultaneously, the robustness test indicates that obtained PID parameters from CAS method have excellent robust stability when the AVR system are in the presence of parameters uncertainty.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (Grant No. 60673098, 60805043, 60821001), the Specialized Research Fund for the Doctoral Program of Higher Education (Grant No. 20070013005, 200800131028), the National Basic Research Program of China (973 Program) (Grant No. 2007CB310704), the 111 Project (Grant No. B08004).

References

[1] Morari M, Lee JH. Model predictive control: past, present and future. Comput Chem Eng 1999;23(4–5):667–82.
[2] Hu HB, Hu QB, Lu ZY, Xu DH. Optimal PID controller design in PMSM servo system via particle swarm optimization. In: The 31st annual conference of the IEEE Industrial Electronics Society, IECON 2005, Raleigh, NC, USA, November 2005. p. 6–10.
[3] Ziegler JG, Nichols NB. Optimum settings for automatic controllers. J Dyn Syst Meas Control 1993;115:220–2.
[4] Coelho LS. Tuning of PID controller for an automatic regulator voltage system using chaotic optimization approach. Chaos Solitons & Fractals, Corrected Proof, Available online 6 August 2007. 2009;39:1504–1514.
[5] Hung ML, Lin JS, Yan JJ, Liao TL. Optimal PID control design for synchronization of delayed discrete chaotic systems. Chaos Solitons & Fractals 2008;35(4):781–5.
[6] Aguilar-López R, Martinez-Guerra R. Partial synchronization of different chaotic oscillators using robust PID feedback. Chaos Solitons & Fractals 2007;33(2):572–81.
[7] Goldberg DE. Genetic algorithms in search, optimization and machine learning. 1st ed. MA: Addison-Wesley; 1989.
[8] Cohen GH, Coon GA. Theoretical consideration of retarded control. Trans ASME 1953;75:827–34.
[9] Chang WD. PID control for chaotic synchronization using particle swarm optimization. Chaos Solitons & Fractals, Corrected Proof, Available online 8 April 2007. 2009;39:910–917.
[10] Li LX, Yang YX, Peng HP, Wang XD. Parameters identification of chaotic systems via chaotic ant swarm. Chaos Solitons & Fractals 2006;28:1204–11.
[11] Cai JJ, Ma XQ, Li LX, Yang YX, Peng HP, Wang XD. Chaotic ant swarm optimization to economic dispatch. Electric Power Syst Res 2007;77(10):1373–80.
[12] Li Y, Ang KH, Chong GCY. PID control system analysis and design. IEEE Control Syst Mag 2006;26(1):32–41.
[13] Finch JW, Zachariah KJ, Farsi M. Turbogenerator self-tuning automatic voltage regulator. IEEE Trans Energy Conversion 1999;14(3):843–8.
[14] Brasca CL, Johnson MA. On automatic voltage regulator design for synchronous generators. In: Proceedings of the third IEEE conference on control applications, vol. 1; 1994. p. 199–205.
[15] Ula A, Hasan AR. Design and implementation of a personal computer based automatic voltage regulator for a synchronous generator. IEEE Trans Energy Conversion 1992;7(1):125–31.
[16] Kundur P. Power system stability and control. New York: McGraw-Hill; 1994.
[17] Yoshida H, Kawata K, Fukuyama Y, Takayama S, Nakanishi Y. A particle swarm optimization for reactive power and voltage control considering voltage security assessment. IEEE Trans Power Syst 2000;15:1232–9.

[18] Gaing ZL. A particle swarm optimization approach for optimum design of PID controller in AVR system. IEEE Trans Energy Conversion 2004;19(2):384–91.
[19] Cole BJ. Is animal behavior chaotic? Evidence from the activity of ants. In: Proceedings of the royal society of London series B – biological sciences, vol. 244; 1991. p. 253–9.
[20] Sole RV, Miramontes O, Goodwin BC. Oscillations and chaos in ant societies. J Theor Biol 1993;161(3):343–57.
[21] Nemes L, Roska T. A CNN model of oscillation and chaos in ant colonies: a case study. IEEE Trans Circuits Syst I 1995;42(10):741–5.
[22] Li LX, Yang YX, Peng HP, Wang XD. An optimization method inspired by "chaotic" ant behavior. Int J Bifurc Chaos 2006;16(8):2351–64.
[23] Eshelman LJ, Schaffer JD. Real-coded genetic algorithms and interval-schemata. Foundations of genetic algorithms 2. San Mateo: Morgan Kaufman Publishers; 1993. p. 187–202.
[24] Kawabe T, Tagami T. A real coded genetic algorithm for matrix inequality design approach of robust PID controller with two degrees of freedom. In: Proceedings of the 1997 IEEE international symposium on intelligent control; 1997. p. 119–24.
[25] Krohling RA, Jaschek H, Rey JP. Designing PI/PID controllers for a motion control system based on genetic algorithms. In: Proceedings of the 1997 IEEE international symposium on intelligent control; 1997. p. 125–30.