Commun Nonlinear Sci Numer Simulat 15 (2010) 3632–3639

Contents lists available at ScienceDirect

Commun Nonlinear Sci Numer Simulat

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

PID controller design of nonlinear systems using an improved particle swarm optimization approach

Wei-Der Chang *, Shun-Peng Shih

Department of Computer and Communication, Shu-Te University, Kaohsiung 824, Taiwan

ARTICLE INFO

Article history:
Received 9 May 2009
Received in revised form 2 January 2010
Accepted 6 January 2010
Available online 14 January 2010

Keywords:
PID controller
Improved particle swarm optimization
Nonlinear systems
Inverted pendulum control

ABSTRACT

In this paper, an improved particle swarm optimization is presented to search for the optimal PID controller gains for a class of nonlinear systems. The proposed algorithm is to modify the velocity formula of the general PSO systems in order for improving the searching efficiency. In the improved PSO-based nonlinear PID control system design, three PID control gains, i.e., the proportional gain K_p, integral gain K_i, and derivative gain K_d are required to form a parameter vector which is called a particle. It is the basic component of PSO systems and many such particles further constitute a population. To derive the optimal PID gains for nonlinear systems, two principle equations, the modified velocity updating and position updating equations, are employed to move the positions of all particles in the population. In the meanwhile, an objective function defined for PID controller optimization problems may be minimized. To validate the control performance of the proposed method, a typical nonlinear system control, the inverted pendulum tracking control, is illustrated. The results testify that the improved PSO algorithm can perform well in the nonlinear PID control system design.

1. Introduction

The particle swarm optimization (PSO) is a new evolutionary computation technique and has been introduced in various application fields in recent years [1–7]. This algorithm combines the social psychology principles in socio-cognition human agents and evolutionary computations. It is initially motivated by the behavior of organisms, such as a fish school and bird flock. The computational efficiency of the PSO algorithm is rather excellent and it is also easy to be implemented. Moreover, unlike other heuristic optimization methods, PSO has a flexible and well-balanced mechanism to enhance the global and local exploration abilities. The algorithm begins with randomly generating an initial population, which is composed of a number of candidate solutions. In the PSO algorithm, every candidate solution is called a particle or an individual and such a particle is moved by a velocity updating in accordance with itself own experience and other particles' experiences. It possesses a constructive cooperation and sharing information relatively between particles of the population [6,7].

Due to the good features of PSO algorithm, nowadays it has been emerged as a new and attractive optimizer and applied in variety of research fields. For example, the system identification for a class of nonlinear rational filters was solved using the PSO algorithm [5]. The rational filter parameters can be correctly estimated. In [6], the author used the PSO algorithm to solve the optimal design problem of multimachine power-system stabilizers (PSSs). Two eigenvalue-based objective functions were considered to enhance the system damping of electromechanical modes. In [7], a hybrid PSO algorithm which incorporates chaos dynamics was proposed to enhance the performance of PSO. In the case the population-based evolutionary searching and chaotic searching are well combined.

* Corresponding author.
E-mail address: wdchang@stu.edu.tw (W.-D. Chang).

1007-5704/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.cnsns.2010.01.005

The design of PID controller is another important subject in this study. In the control engineering, the PID controller has been employed for a very long time due to its good characteristics such as the simplicity in architecture, easy implementation, and mature theoretical analysis. Until now, it is still widely applied to the industrial control, even if many new control approaches have been consequently developed. This controller contains three adjustable gain parameters, the proportional gain K_p, integral gain K_i, and derivative gain K_d, respectively. In the traditional PID controller design, there are the following methods for the linear model plant or linear model plant with time delay: (1) process reaction curve (Z-N tuning method), (2) sustained oscillation PID tuning, (3) damped oscillation or quarter amplitude decay PID tuning, and (4) relay tuning method, etc. [8]. In [9], the author utilized the PSO algorithm to the optimum design of PID controller in the automatic voltage regulator (AVR) system comprising four components: amplifier, exciter, generator, and sensor. Each component is expressed as a simple first-order linear system. Three PID control gains were determined by minimizing a new time-domain performance criterion via the PSO algorithm. This work is about the PSO-based PID controller design for linear systems.

This paper proposes a new improved PSO algorithm and uses it to solve the PID controller design problem for a class of nonlinear systems. The remainder of this paper is organized as follows. In Section 2, the improved PSO algorithm is clearly presented and some function optimization tests are demonstrated. Section 3 presents the full design steps of the improved PSO-based nonlinear PID control system. In Section 4, the tracking control of the inverted pendulum system is typically illustrated to verify the feasibility of the proposed method. Many simulation examinations are also given. Finally, a brief conclusion is stated in Section 5.

2. Improved PSO algorithm

In 1995, Kennedy and Eberhart initially proposed the particle swarm concept and PSO algorithm [1]. This algorithm is one of optimization methods and evolutionary computations. It has been proven to be efficient in solving optimization problem especially for nonlinearity and nondifferentiability, multiple optimum, and high dimensionality. It guides searches using a population constructed by many particles rather than individuals. In the PSO algorithm, each particle then represents a candidate solution to the optimization problem. The particle keeps track of its coordinates in the problem space which are associated with the best solution it has achieved so far. This value is called *pbest*. Another "best" value that is tracked by the global version of the particle swarm optimizer is the overall best value and its location obtained from any particle of the population so far. This location is named *gbest*. At each time step, the PSO algorithm concept consists of changing the velocity that accelerates each particle toward its *pbest* and *gbest* locations. Acceleration is weighted by a random term with separate random numbers being generated for acceleration toward *pbest* and *gbest* locations, respectively [3].

Assume that each particle is considered in the N-dimensional space, $\Theta_i = [\theta_{i1}, \theta_{i2}, \ldots, \theta_{iN}]$ denotes the ith particle's position, and $V_i = [v_{i1}, v_{i2}, \ldots, v_{iN}]$ denotes the ith particle's velocity. The best previous position of the ith particle, i.e., particle best, is represented as $P_i = [p_{i1}, p_{i2}, \ldots, p_{iN}]$ and the index of the best particle among all particles in the population, i.e., global best, is assigned by the symbol g. The velocity and position updating equations of the original PSO algorithm, for $n = 1, 2, \ldots N$, are given below

$$v_{in} = w * v_{in} + c_1 * rand() * (p_{in} - \theta_{in}) + c_2 * Rand() * (p_{gn} - \theta_{in}), \tag{1}$$

$$\theta_{in} = \theta_{in} + v_{in}, \tag{2}$$

where w is called the inertia weight which balances the global and local search, c_1 and c_2 are two positive constants, $rand()$ and $Rand()$ are two random numbers in the interval $[0, 1]$. Eqs. (1) and (2) are the main evolutionary mechanisms of the PSO algorithm for solving the optimization problem.

In this study, an improved velocity updating equation is first proposed. The above Eq. (1) is modified as

$$v_{in} = w * v_{in} + c_1 * rand() * (p_{in} - \theta_{in}) + c_2 * Rand() * (p_{gn} - \theta_{in}) + c_3 * RAND() * (p_{sn} - \theta_{in}), \tag{3}$$

where p_{sn} represents the best particle's position among all particles in the sub-population that the ith particle belongs to, c_3 is a positive constant, and $RAND()$ is also a random number in the range $[0, 1]$. From Eq. (3), it can be seen that the third kind of best particle information is considered. We can illustrate the improved velocity updating formula of Eq. (3) by a student-class-school model. In the model a student can be regarded as a particle, a class as a sub-population, and a school as a population, respectively. In the original PSO algorithm, the experience of a student (particle) is only influenced by his/her own past best experience (*pbest*) and the experience of the global best student (*gbest*) in the school (population). Nevertheless, in the general environment the experience of a student may be not only influenced by these two factors, but really also by the experience of the best student (we here call sub-best, *sbest*) in his/her own class (sub-population). The improved velocity formula is reasonably motivated from this simple concept. The following gives the design step for implementing the improved version of the PSO algorithm:

Step 1. Randomly generate an initial population that contains H particles (population size) from certain search interval.
Step 2. If a prescribed number of iterations (generations) is achieved, then stop the algorithm.
Step 3. Evaluate the objective function of every particle and record each particle's best previous position (*pbest*), sub-best position (*sbest*), and global best position (*gbest*).

Step 4. Perform the improved velocity updating of Eq. (3) and the position updating of Eq. (2) for each particle.
Step 5. Check each particle's position using the following formula:

$$\theta_n = \begin{cases} \theta_{\min} & \text{if } \theta_n < \theta_{\min} \\ \theta_n & \text{if } \theta_{\min} \leqslant \theta_n \leqslant \theta_{\max} \\ \theta_{\max} & \text{if } \theta_n > \theta_{\max} \end{cases}, \text{for} \quad n = 1, 2, \ldots N.$$

Step 6. Go back to Step 2.

Three different test functions are simply illustrated to verify the effectiveness of the proposed algorithm. These test functions are as follows [10,11]:

1. Function 1 (Rosenbrock): $f_1(\theta_1, \theta_2) = 100(\theta_1^2 - \theta_2)^2 + (1 - \theta_1)^2$, Solutions: $\theta_1 = 1.0,\ \theta_2 = 1.0,\ f_1(\theta_1, \theta_2) = 0.0$,
2. Function 2 (Rastrigin): $f_2(\theta_1, \theta_2) = 20 + \theta_1^2 - 10\cos(2\pi\theta_1) + \theta_2^2 - 10\cos(2\pi\theta_2)$, Solutions: $\theta_1 = 0.0,\ \theta_2 = 0.0,\ f_2(\theta_1, \theta_2) = 0.0$,
3. Function 3: $f_3(\theta_1, \theta_2) = \theta_1^2 + 2\theta_2^2 - 0.3\cos(3\pi\theta_1) - 0.4\cos(4\pi\theta_2) + 0.7$, Solutions: $\theta_1 = 0.0,\ \theta_2 = 0.0,\ f_3(\theta_1, \theta_2) = 0.0$.

In the simulations, the variables used in the improved PSO algorithm are given by $w = 0.8$ and $c_1 = c_2 = c_3 = 1.0$. For each test function, the population size is given by $H = 50$, number of sub-populations by 5, and number of generations by 200, respectively. Moreover, the search interval for each particle is given by $[\theta_{\min}, \theta_{\max}] = [-5.0, 5.0]$. In order for evaluating the robustness of the proposed algorithm, four optimization cases with different random sets of initial populations are considered for each function. Simulation results are displayed in Figs. 1–3, respectively, and all of function solutions and optimal function values can be correctly solved.

3. Nonlinear PID control system design using the improved PSO algorithm

The PID control law is generally of the form

$$u(t) = K_p \cdot e(t) + K_i \int_0^t e(\tau)\,d\tau + K_d \frac{d}{dt} e(t), \tag{4}$$

where e is the system error between the desired and actual outputs, u is the control force, K_p is the proportional gain, K_i is the integral gain, and K_d is the derivative gain. We wish to properly design a set of PID gains such that the system output response satisfies certain specifications. In the improved PSO-based PID control system design, let $\Theta = [\theta_1, \theta_2, \theta_3] = [K_p, K_i, K_d]$ be a parameter vector or a particle. Moreover, in this study the controlled plant is a nonlinear dynamic system, formulated by

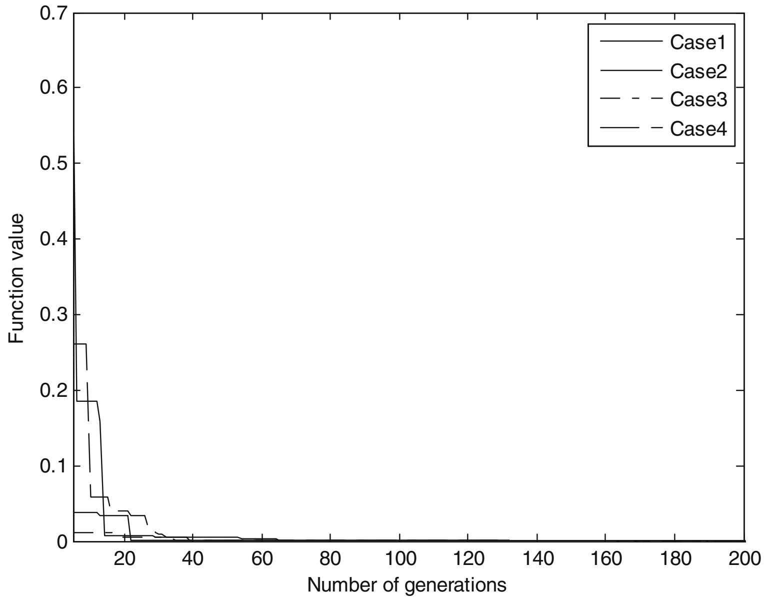

Fig. 1. Function 1 (For clarity, it is shown after fifth generation).

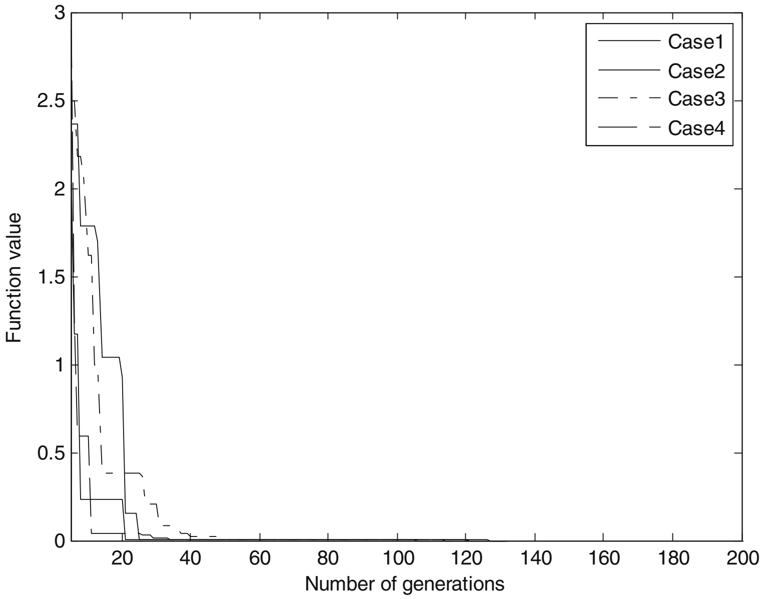

Fig. 2. Function 2 (For clarity, it is shown after fifth generation).

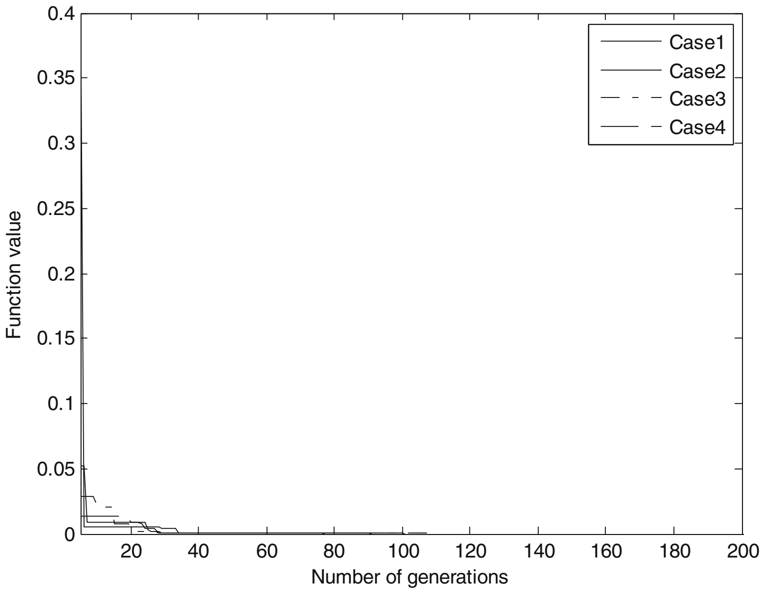

Fig. 3. Function 3 (For clarity, it is shown after fifth generation).

$$\begin{aligned}&\dot{x}_1(t) = x_2(t), \quad \dot{x}_2(t) = x_3(t), \ldots, \dot{x}_{n-1}(t) = x_n(t),\\&\dot{x}_n = f(x_1, x_2, \ldots, x_n) + g(x_1, x_2, \ldots, x_n)u(t),\\&y(t) = x_1(t),\end{aligned} \tag{5}$$

or equivalently by

$$\begin{aligned}&x^{(n)} = f(x, \dot{x}, \ldots, x^{(n-1)}) + g(x, \dot{x}, \ldots, x^{(n-1)})u(t),\\&y(t) = x(t),\end{aligned} \tag{6}$$

where $f,\ g : \Re^n \to \Re$ are two nonlinear functions, u is the control input and here we utilize the PID control law as described by Eq. (4), and y is the actual output of the nonlinear system. The overall block diagram of the nonlinear PID control system design combined with the improved PSO algorithm is shown in Fig. 4, where y_d is the desired output.

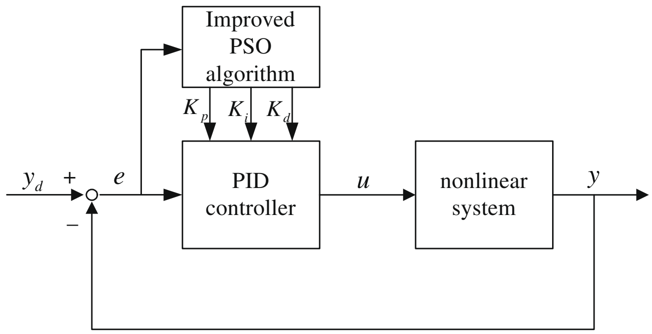

Fig. 4. Block diagram of nonlinear PID control system design combined with the improved PSO algorithm.

The convergence of the PSO algorithm toward the global optimal solution is guided by the objective function. Hence it should be properly defined before the improved PSO algorithm is executed. In the present study, the objective function is defined by the integral of the squared error (ISE) as

$$\mathrm{ISE} = \int_0^{T_i} [y_d(t) - y(t)]^2\, dt = \int_0^{T_i} e^2(t)\, dt, \tag{7}$$

where T_i is the time of integration. This function will be minimized by using the improved PSO algorithm to obtain the optimal PID gains.

The complete steps for designing the improved PSO-based nonlinear PID control system are summarized as follows:

Data: Nonlinear systems of Eqs. (5) or (6), time of integration T_i in Eq. (7), parameter search interval $[\theta_{\min}, \theta_{\max}]$, population size H, inertia weight w and positive constants $c_1,\ c_2$, and c_3 in Eq. (3), number of sub-populations, and number of iterations (generations) N.

Goal: Search for a set of optimal PID control gains for a class of nonlinear systems of Eqs. (5) or (6) via the improved PSO algorithm such that the objective function ISE of Eq. (7) is minimized.

Step 1. Randomly generate an initial population that contains H particles (population size) from certain interval.
Step 2. If a prescribed number of iterations (generations) N is achieved, then stop the algorithm.
Step 3. Evaluate the corresponding objective function ISE of every particle and record each particle's best previous position (*pbest*), sub-best position (*sbest*), and global best position (*gbest*).
Step 4. Perform the improved velocity updating of Eq. (3) and the position updating of Eq. (2) for each particle.
Step 5. Check each particle's position using the following formula:

$$\theta_n = \begin{cases} \theta_{\min} & \text{if } \theta_n < \theta_{\min} \\ \theta_n & \text{if } \theta_{\min} \leqslant \theta_n \leqslant \theta_{\max} \\ \theta_{\max} & \text{if } \theta_n > \theta_{\max} \end{cases}, \text{for} \quad n = 1, 2, 3.$$

Step 6. Go back to Step 2.

4. Tracking control of the inverted pendulum system

The inverted pendulum system is rather nonlinear and complicated, and it is a typical example of nonlinear system control. This section demonstrates the feasibility of the improved PSO-based PID controller design on such nonlinear systems. Its dynamic differential equations can be described by [12]

$$\begin{aligned} \dot{x}_1(t) &= x_2(t), \\ \dot{x}_2(t) &= \frac{g \sin x_1(t) - \frac{m l x_2^2(t) \cos x_1(t) \sin x_1(t)}{m_c + m}}{l\left(\frac{4}{3} - \frac{m \cos^2 x_1(t)}{m_c + m}\right)} + \frac{\frac{\cos x_1(t)}{m_c + m}}{l\left(\frac{4}{3} - \frac{m \cos^2 x_1(t)}{m_c + m}\right)} u(t), \\ y(t) &= x_1(t), \end{aligned} \tag{8}$$

where $x_1 (= y)$ is the angle of the pole in radian with respect to the vertical axis, x_2 is the angular velocity of the pole, $g = 9.8\ \mathrm{m/s^2}$ is the acceleration due to gravity, $m_c = 1$ kg is the mass of cart, $m = 0.1$ kg is the mass of pole, $l = 0.5$ m is the half length of the pole, and u is the control input in Newton and here the PID controller is adopted.

Table 1
Variables of the improved PSO algorithm used in the nonlinear inverted pendulum control system.

Variables	Values
Integration time T_i	2
Search interval $[\theta_{\min}, \theta_{\max}]$	[0, 500]
Population size H	50
Number of sub-populations	5
Inertia weight w	0.8
Constant c_1	1.0
Constant c_2	1.0
Constant c_3	1.0
Number of iterations N	200

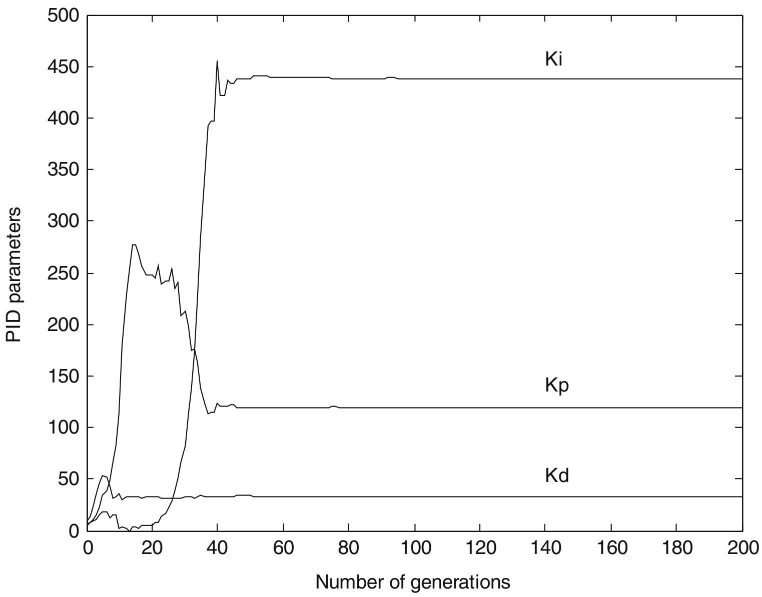

Fig. 5. Convergence curves of PID control gains for Case 1.

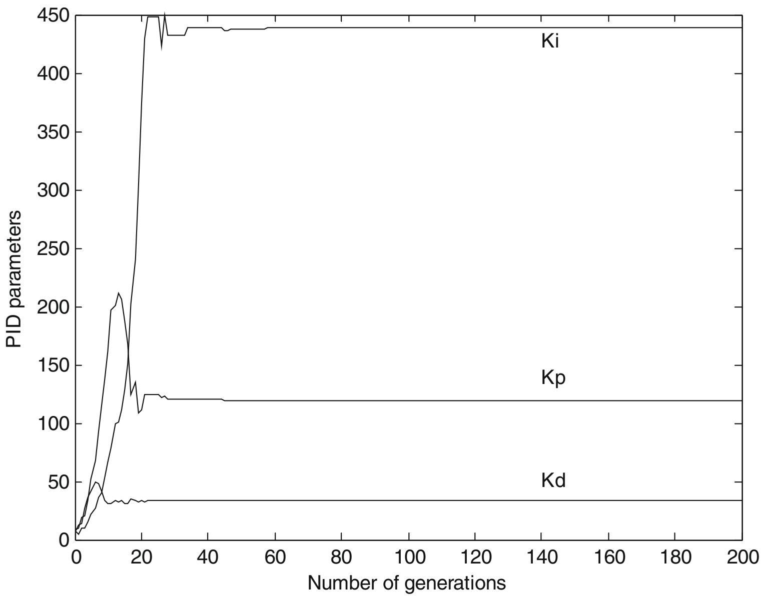

Fig. 6. Convergence curves of PID control gains for Case 2.

In the following simulation, the packet software of Borland C++ 5.02 is utilized to implement the above improved PSO algorithm. The sampling time is set to 0.01 s for simulating the nonlinear differential equations. For solving the optimal PID controller to the inverted pendulum control system, the values assigned to the related variables of the improved PSO algorithm are listed in Table 1. The control purpose is to make the pole angle y of the inverted pendulum track the desired output $y_d = \pi \sin(t)/30$ with initial states $(x_1(0), x_2(0)) = (0.1, 0)$.

To examine the robustness and effectiveness of the proposed PSO algorithm, four simulations cases with different random sets of initial populations are also considered. All of particles in the population are randomly generated from the interval $[0, 10]$ in the beginning. Simulation results reveal that three PID control gains are finally solved to the same values as $K_p = 119.70,\ K_i = 438.42$, and $K_d = 33.56$, and the corresponding value of ISE all converges to 0.0002747, regardless of any cases. This confirms that the solution quality of the improved PSO algorithm does not heavily rely on initial populations. From any starting points, the proposed algorithm still ensures the convergence to the optimal solution. Figs. 5–8 then dem-

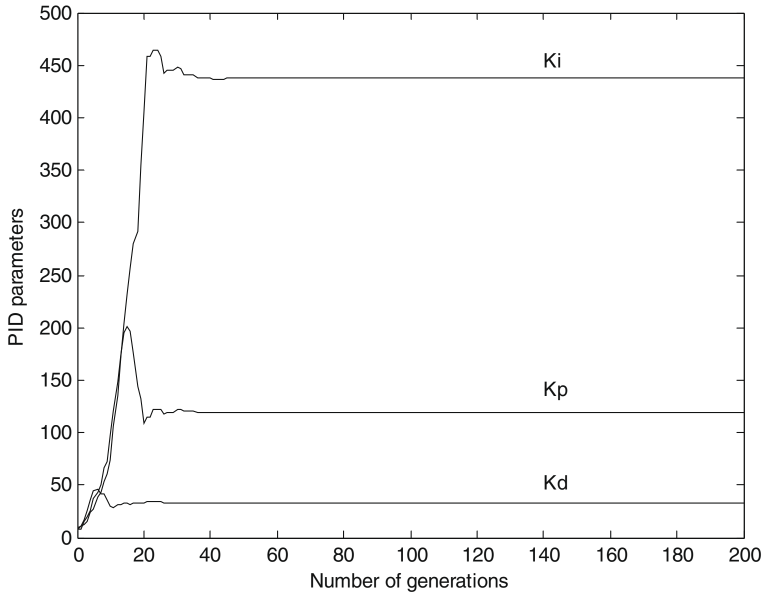

Fig. 7. Convergence curves of PID control gains for Case 3.

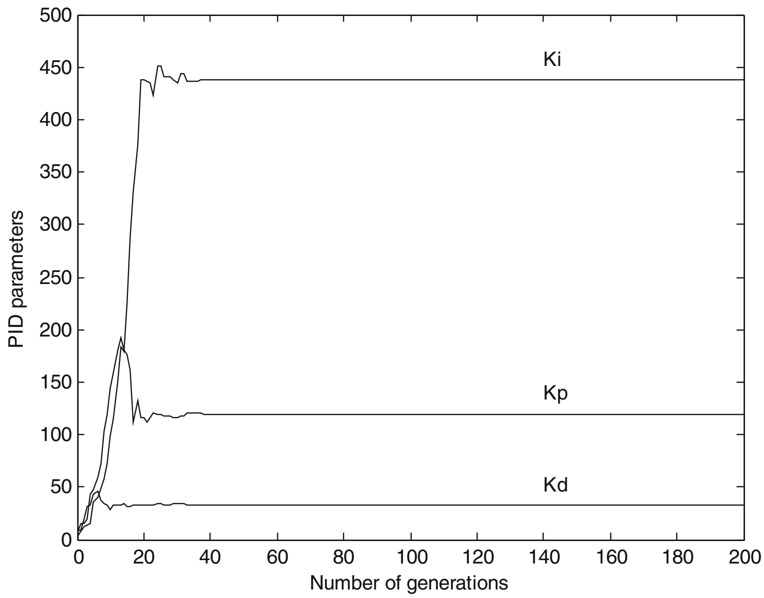

Fig. 8. Convergence curves of PID control gains for Case 4.

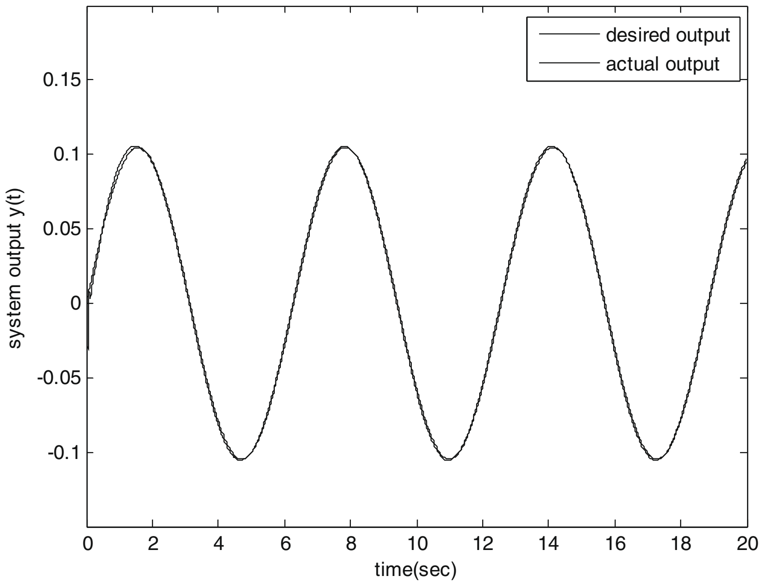

Fig. 9. Tracking control for the inverted pendulum using the derived PID control gains $K_p = 119.70,\ K_i = 438.42$, and $K_d = 33.56$.

onstrate the convergences of PID control gains with respect to the number of generations from different initial populations (Cases 1–4), respectively. These figures reveal that the improved PSO algorithm has stable convergence with good computational efficiency. Based on the obtained PID control gains, the tracking control of the inverted pendulum is demonstrated in Fig. 9, and an excellent tracking performance is also achieved.

5. Conclusions

In this paper, a design method for a class of nonlinear PID control system has been fully presented. We propose a new improved PSO algorithm to enhance the algorithm search ability, which introduces the third kind of best particle into the original velocity updating formula. With minimizing the objective function, three PID control gains can be optimally solved by using the improved PSO algorithm. To verify the feasibility of the proposed method in the nonlinear PID control system design, the tracking control of the nonlinear inverted pendulum is illustrated. Several examination cases with different initial populations are provided to show the stable convergence of the proposed method.

References

[1] Kennedy J, Eberhart R. Particle swarm optimization. In: Proc IEEE int conf neural networks, vol. IV, Perth, Australia; 1995. p. 1942–48.
[2] Shi Y, Eberhart RC. Empirical study of particle swarm optimization. In: Proc IEEE int conf evol comput, Washington, DC; July 1999. p. 1945–50.
[3] Eberhart RC, Shi Y. Particle swarm optimization: development, applications and resources. In: Proc congress on evol comput, Seoul, Korea; 2001. p. 81–6.
[4] Hwang KC. Optimisation of broadband twist reflector for ku-band application. Electron Lett 2008;44(3):210–1.
[5] Lin YL, Chang WD, Hsieh JG. A particle swarm optimization approach to nonlinear rational filter modeling. Expert Syst Appl 2008;34:1194–9.
[6] Abido MA. Optimal design of power-system stabilizers using particle swarm optimization. IEEE Trans Energy Conversion 2002;17:406–13.
[7] Liu B, Wang L, Jin YH, Tang F, Huang DX. Improved particle swarm optimization combined with chaos. Chaos Solitons Fractals 2005;25:1261–71.
[8] Wilkie J, Johnson M, Katebi R. Control engineering. New York: Palgrve; 2002.
[9] Gaing ZL. A particle swarm optimization approach for optimum design of PID controller in AVR system. IEEE Trans Energy Conversion 2004;19:384–91.
[10] Karaboga N, Kalinli A, Karaboga D. Designing digital IIR filters using ant colony optimization algorithm. Eng Appl Artif Intell 2004;17:301–9.
[11] Tutkun N. Optimization of multimodal continuous functions using a new crossover for the real-coded genetic algorithms. Expert Syst Appl 2009;36:8172–7.
[12] Wang LX. Adaptive fuzzy systems and control: design and stability analysis. New Jersey: Prentice-Hall; 1994.