Int. J. Mach. Learn. & Cyber. (2011) 2:225–234
DOI 10.1007/s13042-011-0021-4

ORIGINAL ARTICLE

A robust self-learning PID control system design for nonlinear systems using a particle swarm optimization algorithm

Chih-Min Lin · Ming-Chia Li · Ang-Bung Ting · Ming-Hung Lin

Received: 23 February 2010 / Accepted: 18 May 2011 / Published online: 22 June 2011

Abstract This study presents a robust self-learning proportional-integral-derivative (RSPID) control system design for nonlinear systems. This RSPID control system comprises a self-learning PID (SPID) controller and a robust controller. The gradient descent method is utilized to derive the on-line tuning laws of SPID controller; and the H_∞ control technique is applied for the robust controller design so as to achieve robust tracking performance. Moreover, in order to achieve fast learning of PID controller, a particle swarm optimization (PSO) algorithm is adopted to search the optimal learning-rates of PID adaptive gains. Finally, two nonlinear systems, a two-link manipulator and a chaotic system are examined to illustrate the effectiveness of the proposed control algorithm. Simulation results show that the proposed control system can achieve favorable control performance for these nonlinear systems.

Keywords PID control · Particle swarm optimization (PSO) · H_∞ control

C.-M. Lin (✉) · M.-C. Li · A.-B. Ting · M.-H. Lin
Department of Electrical Engineering, Yuan Ze University, No. 135, Yuan-Tung Road, Chung-Li, Tao-Yuan 320, Taiwan, ROC
e-mail: cml@saturn.yzu.edu.tw

M.-C. Li
e-mail: s929101@mail.yzu.edu.tw

A.-B. Ting
e-mail: s968501@mail.yzu.edu.tw

M.-H. Lin
e-mail: s958505@mail.yzu.edu.tw

1 Introduction

The proportional-integral-derivative (PID) controller has been practically applied in industries for 60 years due to its simple architecture and easy design properties. Until today, the PID controller is still used in different control applications, even though lots of new control techniques have been proposed. However, the traditional PID controller needs some manual tuning before it is used to practical application in industries. When the PID controllers are applied to complicated systems such as nonlinear systems, different tuning algorithms of PID controllers have been proposed [1–5]. In recent decade, several intelligent tuning algorithms have been applied to tune the PID controllers. The PID controller automatic tuning methods have been proposed by using a genetic algorithm [6], an immune algorithm [7] and a fuzzy-genetic algorithm [8]. However, the learning speed of these algorithms is slow; thus, they are not suitable for real time control systems. Harinath and Mann [9] proposed a fuzzy PID controller for multivariable process systems. However, it takes a two-level tuning algorithm; thus, it also takes too much computation time. Therefore, in this paper, another simple optimization algorithm called particle swam optimization (PSO) algorithm will be used for the optimal parameter search of the PID controller.

The PSO algorithm, a new evolutionary computation technique, is proposed by Kennedy and Eberhart [10]. It was developed by the research of the social behavior of animals, e.g., bird flocking. Unlike a typical GA, PSO algorithms have memorial capability without complicated evolutionary process such as crossover and mutation in GA, and each particle can memorize its best solution. In addition, if another particle discovers a better solution, it will be shared among other particles. The best solution is

thus memorized and every particle will move toward this best solution. In past decade, PSO algorithms have been widely used to solve the modeling and control problems of application systems [11–14]. Moreover, the PID control based on PSO algorithms have been proposed in [15, 16]. However, it is difficult to solve the inequality for the optimal solution in [15]; and it has not given the stability analysis in [16].

In this paper, a robust self-learning PID (RSPID) control system is proposed for nonlinear systems. This RSPID control system comprises a self-learning PID (SPID) controller and a robust controller. The SPID controller is utilized to approximate an ideal controller, and the robust controller is designed to recover the residual approximation error between the ideal controller and SPID controller. The gradient descent method and H_∞ control technique are utilized to derive the on-line tuning laws of SPID controller and robust controller, so that the robust stability of the system can be obtained. Furthermore, the PSO algorithm is adopted to auto-search the optimal learning-rates of PID controller to increase the learning speed. Finally, two nonlinear systems are presented to support the validity of the proposed control method.

This study is organized as follows. Problem formulation is described in Sect. 2. The PSO algorithm is briefly reviewed in Sect. 3. The design procedure of the proposed RSPID control system is constructed in Sect. 4. In Sect. 5, simulations are performed to verify the effectiveness of the proposed control method. Finally, conclusions are drawn in Sect. 6.

2 Problem formulation

Consider a class of nth-order multi-input multi-output (MIMO) nonlinear systems expressed in the following form:

$$\begin{aligned} \boldsymbol{x}^{(n)}(t) &= \boldsymbol{f}(\underline{\boldsymbol{x}}(t)) + \boldsymbol{G}(\underline{\boldsymbol{x}})(t)\boldsymbol{u}(t) + \boldsymbol{d}(t) \\ \boldsymbol{y} &= \boldsymbol{x}(t) \end{aligned} \tag{1}$$

where

$\boldsymbol{u}(t) = [u_1(t), u_2(t), \ldots, u_m(t)]^T \in \Re^m$ the control input vector of the system

$\boldsymbol{y} = \boldsymbol{x}(t) = [x_1(t), x_2(t), \ldots, x_m(t)]^T \in \Re^m$ the system output vector

$\underline{\boldsymbol{x}}(t) = [\boldsymbol{x}^T(t), \dot{\boldsymbol{x}}^T(t), \ldots, \boldsymbol{x}^{(n-1)T}(t)]^T \in \Re^{mn}$ the state vector of the system

$\boldsymbol{f}(\underline{\boldsymbol{x}}(t)) \in \Re^m$ unknown bounded nonlinear function

$\boldsymbol{G}(\underline{\boldsymbol{x}}(t)) \in \Re^{m\times m}$ unknown bounded nonlinear matrix

$\boldsymbol{d}(t) = [d_1(t), d_2(t), \ldots, d_m(t)]^T \in \Re^m$ unknown bounded external disturbance.

When neglecting the modeling uncertainty and external disturbance, the nominal system of Eq. 1 can be obtained as

$$\boldsymbol{x}^{(n)}(t) = \boldsymbol{f}_n(\underline{\boldsymbol{x}}(t)) + \boldsymbol{G}_n\boldsymbol{u}(t) \tag{2}$$

where $\boldsymbol{f}_n(\underline{\boldsymbol{x}}(t)) \in \Re^{\mathrm{m}}$ is the nominal function of $\boldsymbol{f}(\underline{\boldsymbol{x}}(t))$, and the constant matrix $\boldsymbol{G}_n = diag(g_{n1}, g_{n2}, \ldots, g_{nm}) \in \Re^{m\times m}$ is the nominal functions of $\boldsymbol{G}(\underline{\boldsymbol{x}}(t))$. Without losing generality, assume the constant $g_{ni} \geq 0$ for $i = 1\ldots m$. Assume that the nonlinear system of Eq. 2 is controllable and $\boldsymbol{G}_n^{-1}$ exists for all $\underline{\boldsymbol{x}}(t)$. If the external disturbance and modeling uncertainty exist, the MIMO nonlinear systems Eq. 1 can be reformulated as

$$\boldsymbol{x}^{(n)}(t) = \boldsymbol{f}_n(\underline{\boldsymbol{x}}(t)) + \boldsymbol{G}_n\boldsymbol{u}(t) + \boldsymbol{l}(\underline{\boldsymbol{x}}(t), t) \tag{3}$$

where $\boldsymbol{l}(\underline{\boldsymbol{x}}(t), t)$ is referred to as the lumped uncertainty, including system's uncertainty and external disturbance.

The control problem is to find a suitable controller for the MIMO nonlinear systems Eq. 1 so that the system output vector $\boldsymbol{x}(t)$ can track desired reference trajectory vector $\boldsymbol{x}_d(t) = [x_{d1}(t), x_{d2}(t), \ldots, x_{dm}(t)]^T \in \Re^m$ closely.

A lot of control techniques have been presented to achieve reference trajectory tracking. However, in this paper, a simple adaptive PID control scheme will be proposed to deal with the uncertain MIMO nonlinear system. Moreover, the H_∞ control technique will be used to guarantee the robust tracking performance.

Define the tracking error as

$$\boldsymbol{e}(t) \underline{\Delta} \boldsymbol{x}_d(t) - \boldsymbol{x}(t) \in \Re^m \tag{4}$$

then the system tracking error vector is defined as

$$\underline{\boldsymbol{e}} \underline{\Delta} [\boldsymbol{e}^T(t), \dot{\boldsymbol{e}}^T(t), \ldots, \boldsymbol{e}^{(n-1)T}(t)]^T \in \Re^{mn} \tag{5}$$

If the system dynamics $\boldsymbol{f}_n(\underline{\boldsymbol{x}}(t))$ and $\boldsymbol{G}_n$, and the lumped uncertainty $\boldsymbol{l}(\underline{\boldsymbol{x}}, t)$ are exactly known, an ideal controller can be designed as

$$\boldsymbol{u}^*(t) = \boldsymbol{G}_n^{-1}[\boldsymbol{x}_d^{(n)}(t) - \boldsymbol{f}_n(\underline{\boldsymbol{x}}(t)) - \boldsymbol{l}(\underline{\boldsymbol{x}}(t), t) + \underline{\boldsymbol{H}}^T\underline{\boldsymbol{e}}(t)] \tag{6}$$

where $\underline{\boldsymbol{H}} = [\boldsymbol{H}_n, \ldots, \boldsymbol{H}_2, \boldsymbol{H}_1]^T \in \Re^{mn\times m}$ is the feedback gain matrix which contains real numbers. $\boldsymbol{H}_i = diag(h_{i1}, h_{i2}, \ldots, h_{im}) \in \Re^{m\times m}$ is a nonzero positive constant diagonal matrix. Substituting the ideal controller Eq. 6 into Eq. 3, gives the error dynamic equation

$$e^{(n)} + \underline{\boldsymbol{H}}^T \underline{\boldsymbol{e}} = 0 \tag{7}$$

In Eq. 7, if $\underline{\boldsymbol{H}}$ is chosen to correspond to the coefficients of a Hurwitz polynomial, it implies $\lim_{t\to\infty} ||\underline{\boldsymbol{e}}(t)|| = 0$. However, in practical application, the system uncertainties and external disturbance of nonlinear systems are generally unknown, so that the idea controller $\boldsymbol{u}^*$ in Eq. 6 is always unobtainable. Thus, a SPID controller is designed to mimic the idea controller. And then, based on the H_∞ control technique, the robust controller is developed to attenuate the effect of the approximation error between SPID controller and the ideal controller so that the robust tracking performance can be achieved.

3 Robust self-learning PID (RSPID) control system design

The block diagram of the nonlinear control system is shown in Fig. 1. The RSPID control system is assumed to take the following form:

$$\boldsymbol{u}_{RSPID}(t) = \boldsymbol{u}_{SPID}(t) + \boldsymbol{u}_R(t) \tag{8}$$

where $\boldsymbol{u}_{SPID}(t)$ is a self-learning PID controller utilized to approximate the ideal controller $\boldsymbol{u}^*$, and $\boldsymbol{u}_R(t)$ is the robust controller designed to suppress the influence of residual approximation error between the ideal controller and SPID controller.

3.1 SPID controller design

The SPID controller can be described as

$$\boldsymbol{u}_{SPID}(t) = \hat{\boldsymbol{K}}_P \boldsymbol{e}(t) + \hat{\boldsymbol{K}}_I \int_0^t \boldsymbol{e}(\tau)d\tau + \hat{\boldsymbol{K}}_D \frac{d\boldsymbol{e}(t)}{dt} \tag{9}$$

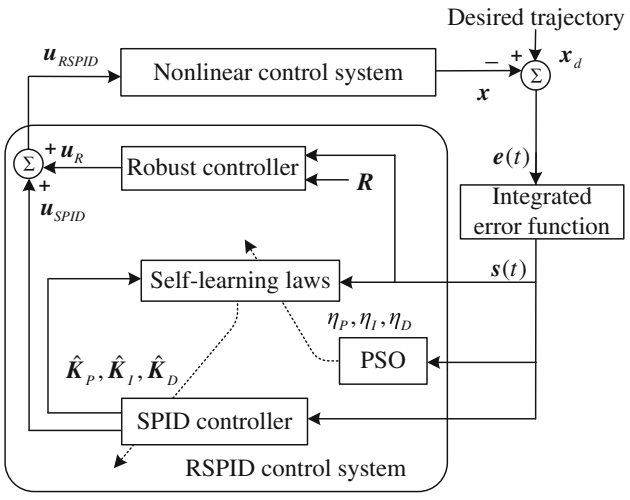

Fig. 1 The block diagram of RSPID control system

where $\hat{\boldsymbol{K}}_P$, $\hat{\boldsymbol{K}}_I$ and $\hat{\boldsymbol{K}}_D$ are the adaptive parameters of proportional gain, integral gain and derivative gain matrices, respectively; and $\hat{\boldsymbol{K}}_P = diag(\hat{k}_{P1}, \hat{k}_{P2}, \ldots, \hat{k}_{Pm}) \in \Re^{m\times m}$, $\hat{\boldsymbol{K}}_I = diag(\hat{k}_{I1}, \hat{k}_{I2}, \ldots, \hat{k}_{Im}) \in \Re^{m\times m}$, $\hat{\boldsymbol{K}}_D = diag(\hat{k}_{D1}, \hat{k}_{D2}, \ldots \hat{k}_{Dm}) \in \Re^{m\times m}$.

An integrated error function is defined as

$$\boldsymbol{s}(\underline{\boldsymbol{e}}, t) \equiv \boldsymbol{e}^{(n-1)} + \boldsymbol{H}_1 \boldsymbol{e}^{(n-2)} + \cdots + \boldsymbol{H}_n \int_0^t \boldsymbol{e}(\tau)d\tau \tag{10}$$

where $\boldsymbol{s}(\underline{\boldsymbol{e}}, t) = [s_1(t), s_2(t), \ldots, s_m(t)]^T$. From Eq. 9, the control law Eq. 8 can be rewritten as

$$\boldsymbol{u}_{RSPID}(t) = \boldsymbol{u}_{SPID}(\hat{\boldsymbol{K}}_P, \hat{\boldsymbol{K}}_I, \hat{\boldsymbol{K}}_D, t) + \boldsymbol{u}_R(t) \tag{11}$$

Taking the time derivative of both sides of Eq. 10 and using Eq. 3, it can be obtained that

$$\begin{aligned}\dot{\boldsymbol{s}}(\underline{\boldsymbol{e}}, t) &= \boldsymbol{e}^{(n)} + \underline{\boldsymbol{H}}^T \underline{\boldsymbol{e}} \\ &= -\boldsymbol{f}_n(\underline{\boldsymbol{x}}(t)) - \boldsymbol{G}_n \boldsymbol{u}(t) + \boldsymbol{x}_d^{(n)} - \boldsymbol{l}(\underline{\boldsymbol{x}}(t), t) + \underline{\boldsymbol{H}}^T \underline{\boldsymbol{e}}\end{aligned} \tag{12}$$

Substituting Eq. 9 into Eq. 12 and multiplying both sides by $\boldsymbol{s}^T(\underline{\boldsymbol{e}}, t)$, yields

$$\begin{aligned}\boldsymbol{s}^T(\underline{\boldsymbol{e}}, t)\dot{\boldsymbol{s}}(\underline{\boldsymbol{e}}, t) = &-\boldsymbol{s}^T(\underline{\boldsymbol{e}}, t)\boldsymbol{f}_n(\underline{\boldsymbol{x}}(t)) - \boldsymbol{s}^T(\underline{\boldsymbol{e}}, t)\boldsymbol{G}_n \\ &[\boldsymbol{u}_{SPID}(\hat{\boldsymbol{K}}_P, \hat{\boldsymbol{K}}_I, \hat{\boldsymbol{K}}_D, t) + \boldsymbol{u}_R(t)] \\ &+ \boldsymbol{s}^T(\underline{\boldsymbol{e}}, t)\ (\boldsymbol{x}_d^{(n)} - \boldsymbol{l}(\underline{\boldsymbol{x}}(t), t) + \underline{\boldsymbol{H}}^T \underline{\boldsymbol{e}})\end{aligned} \tag{13}$$

By defining $\frac{1}{2}\boldsymbol{s}^T(\underline{\boldsymbol{e}}, t)\boldsymbol{s}(\underline{\boldsymbol{e}}, t)$ as a cost function, then its derivative is $\boldsymbol{s}^T(\underline{\boldsymbol{e}}, t)\dot{\boldsymbol{s}}(\underline{\boldsymbol{e}}, t)$. According to the gradient descent method, the gains of $\hat{\boldsymbol{K}}_P$, $\hat{\boldsymbol{K}}_I$ and $\hat{\boldsymbol{K}}_D$ are updated by the following tuning laws

$$\dot{\hat{k}}_{Pi} = -\eta_p \frac{\partial \boldsymbol{s}^T(\underline{\boldsymbol{e}}, t)\dot{\boldsymbol{s}}(\underline{\boldsymbol{e}}, t)}{\partial u_{SPID_i}(t)} \frac{\partial u_{SPID_i}(t)}{\partial \hat{k}_{Pi}} = \eta_P s_i(t) g_{ni} e_i(t) \tag{14}$$

$$\dot{\hat{k}}_{Ii} = -\eta_I \frac{\partial \boldsymbol{s}^T(\underline{\boldsymbol{e}}, t)\dot{\boldsymbol{s}}(\underline{\boldsymbol{e}}, t)}{\partial u_{SPID_i}(t)} \frac{\partial u_{SPID_i}(t)}{\partial \hat{k}_{Ii}} = \eta_I s_i(t) g_{ni} \int_0^t e_i(t)d\tau \tag{15}$$

$$\dot{\hat{k}}_{Di} = -\eta_D \frac{\partial \boldsymbol{s}^T(\underline{\boldsymbol{e}}, t)\dot{\boldsymbol{s}}(\underline{\boldsymbol{e}}, t)}{\partial u_{SPID_i}(t)} \frac{\partial u_{SPID_i}(t)}{\partial \hat{k}_{Di}} = \eta_D s_i(t) g_{ni} \frac{de_i(t)}{dt} \tag{16}$$

where u_{SPID_i} is the ith element of $\boldsymbol{u}_{SPID}$; η_P, η_I and η_D are the learning-rates, which will be auto-searched by PSO algorithm.

3.2 Robust controller design

In case of the existence of an approximation error, the ideal controller can be reformulated as the summation of SPID controller and the approximation error:

$$\boldsymbol{u}^*(t) = \boldsymbol{u}_{SPID}(\hat{\boldsymbol{K}}_P, \hat{\boldsymbol{K}}_I, \hat{\boldsymbol{K}}_D, t) + \varepsilon(t) \tag{17}$$

where $\varepsilon(t) = [\varepsilon_1(t), \varepsilon_2(t), \ldots, \varepsilon_m(t)]^T \in \Re^m$ denotes the approximation error.

Substituting Eq. 8 into Eq. 3, yields

$$\boldsymbol{x}^{(n)}(t) = \boldsymbol{f}_n(\underline{\boldsymbol{x}}(t)) + \boldsymbol{G}_n[\boldsymbol{u}_{SPID}(t) + \boldsymbol{u}_R(t)] + \boldsymbol{l}(\underline{\boldsymbol{x}}(t), t) \tag{18}$$

From Eqs. 6, 10, 18 and after some straightforward manipulation, it can be obtained that

$$\boldsymbol{e}^{(n)} + \underline{\boldsymbol{H}}^T\underline{\boldsymbol{e}} = \boldsymbol{G}_n[\boldsymbol{u}^*(t) - \boldsymbol{u}_{SPID}(t) - \boldsymbol{u}_R(t)] = \dot{\boldsymbol{s}}(\underline{\boldsymbol{e}}, t) \tag{19}$$

Now, the robust controller can be developed to attenuate the effect of the approximation error between the ideal controller and SPID controller so that the H_∞ tracking performance can be achieved. In case of the existence of $\varepsilon(t)$, consider a specified H_∞ tracking performance [17]

$$\sum_{i=1}^{m}\int_0^T s_i^2(t)dt \le \sum_{i=1}^{m}[s_i^2(0)/g_{ni}] + \sum_{i=1}^{m} r_i^2 \int_0^T \varepsilon_i^2(t)dt \tag{20}$$

where r_i is a prescribed attenuation constant. The robust controller is designed as

$$\boldsymbol{u}_R(t) = (2\boldsymbol{R}^2)^{-1}(\boldsymbol{R}^2 + \boldsymbol{I})\boldsymbol{s}(\underline{\boldsymbol{e}}, t) \tag{21}$$

where $\boldsymbol{R} = diag(r_1, r_2, \ldots, r_m) \in \Re^{m\times m}$. Then the following theorem can be stated and proven.

Theorem 1: *Consider the nth-order MIMO nonlinear systems represented by Eq. 1. The RSPID control law is designed as Eq. 11, where $\boldsymbol{u}_{SPID}(t)$ is given in Eq. 9 with the on-line parameter tuning algorithms given as Eqs. 14–16, and the robust controller is designed as Eq. 21. Then the desired H_∞ tracking performance in Eq. 20 can be achieved for the specified attenuation levels $r_i, i = 1, 2, \ldots, m$.*

Proof: The Lyapunov function is given by

$$V(\boldsymbol{s}(\underline{\boldsymbol{e}}, t)) = \frac{1}{2}\boldsymbol{s}^T(\underline{\boldsymbol{e}}, t)\boldsymbol{s}(\underline{\boldsymbol{e}}, t) \tag{22}$$

Taking the derivative of the Lyapunov function and using Eqs. 17, 19 and 21, yields

$$\begin{aligned}
\dot{V}(\boldsymbol{s}(\underline{\boldsymbol{e}}, t)) &= \boldsymbol{s}^T(\underline{\boldsymbol{e}}, t)\dot{\boldsymbol{s}}(\underline{\boldsymbol{e}}, t) \\
&= \boldsymbol{s}^T(\underline{\boldsymbol{e}}, t)\boldsymbol{G}_n[\varepsilon(t) - (2\boldsymbol{R}^2)^{-1}(\boldsymbol{R}^2 + \boldsymbol{I})\boldsymbol{s}(\underline{\boldsymbol{e}}, t)] \\
&= \sum_{i=1}^{m} g_{ni}\left[s_i(t)\varepsilon_i(t) - s_i^2(t)\frac{r_i^2 + 1}{2r_i^2}\right] \\
&= \sum_{i=1}^{m} g_{ni}\left[s_i(t)\varepsilon_i(t) - \frac{s_i^2(t)}{2} - \frac{s_i^2(t)}{2r_i^2}\right] \\
&= \sum_{i=1}^{m} g_{ni}\left[-\frac{s_i^2(t)}{2} - \frac{1}{2}\left(\frac{s_i(t)}{r_i} - r_i\varepsilon_i(t)\right)^2 + \frac{r_i^2\varepsilon_i^2(t)}{2}\right] \\
&\le \sum_{i=1}^{m} g_{ni}\left[-\frac{s_i^2(t)}{2} + \frac{r_i^2\varepsilon_i^2(t)}{2}\right]
\end{aligned} \tag{23}$$

Assuming $\varepsilon_i(t) \in L_2[0, T], \forall T \in [0, \infty)$, integrating the above equation from $t = 0$ to $t = T$, yields

$$V(T) - V(0) \le \sum_{i=1}^{m} g_{ni}\left[-\frac{1}{2}\int_0^T s_i^2(t)(t)\,dt + \frac{r_i^2}{2}\int_0^T \varepsilon_i^2(t)dt\right] \tag{24}$$

Since $V(T) \ge 0$, the above inequality implies the following inequality

$$\frac{1}{2}\sum_{i=1}^{m} g_{ni}\int_0^T s_i^2(t)dt \le V(0) + \frac{1}{2}\sum_{i=1}^{m} g_{ni}r_i^2\int_0^T \varepsilon_i^2(t)dt \tag{25}$$

Using Eq. 22, the above inequality is equivalent to the following

$$\sum_{i=1}^{m}\int_0^T s_i^2(t)dt \le \sum_{i=1}^{m}[s_i^2(0)/g_{ni}] + \sum_{i=1}^{m} r_i^2\int_0^T \varepsilon_i^2(t)dt \tag{26}$$

Thus the proof is completed.

4 Particle swarm optimization (PSO) algorithm

The learning-rates of the tuning laws in SPID controller are usually selected by trial-and-error process. In order to achieve the best learning speed, the PSO algorithm is adopted to search the optimal learning-rates η_P, η_I and η_D in the SPID controller.

In 1995, Kennedy and Eberhart [10] initially proposed the particle swarm concept and PSO algorithm; this algorithm is one of optimization methods. It has been proven to be efficient in solving optimization problem. In the PSO algorithm, each particle represents a candidate solution to the optimization problem. The particle keeps track of its coordinates in the problem space which are associated with the personal best solution. Another is the global best value that is tracked by the global version of the particle swarm optimizer. At each time step, the PSO algorithm consists of changing the velocity that accelerates each particle toward its personal best and global best locations. Acceleration is weighted by a random term with separate random numbers being generated for acceleration toward personal best and global best locations, respectively [11]. From then on, several PSO algorithms have been proposed with slightly different versions [12–16].

The flowchart of the utilized PSO algorithm is drawn in Fig. 2.

4.1 Fitness function

In order to maintain the control characteristic of SPID controller, a fitness function is chosen as

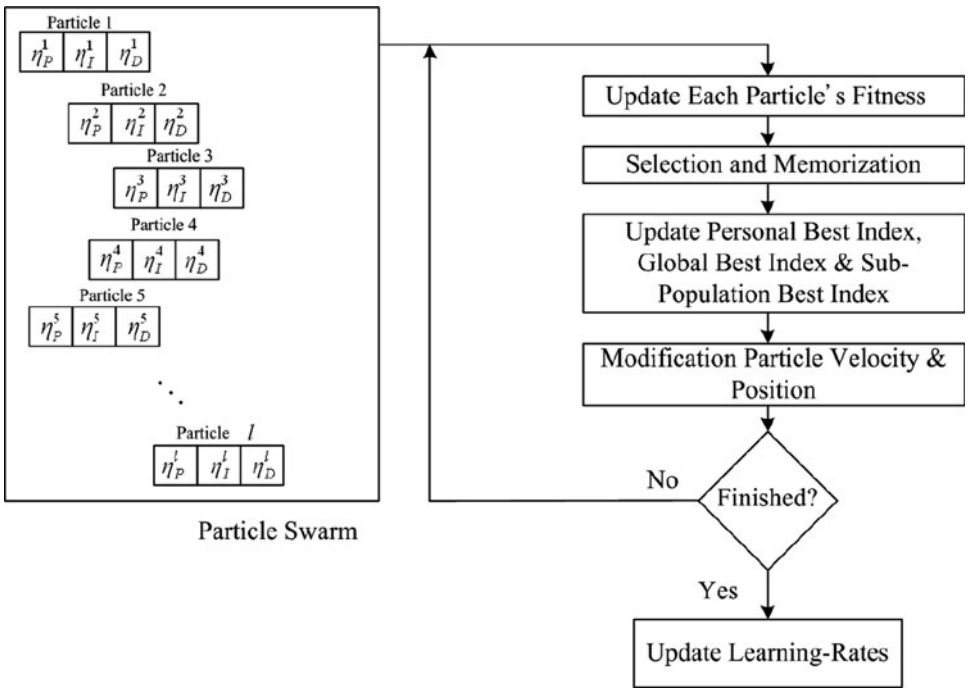

Fig. 2 The flowchart of PSO algorithm

$$fit = \frac{1}{0.1 + ||\underline{\boldsymbol{e}}(t)||} \tag{27}$$

It means that if the error states $\underline{\boldsymbol{e}}(t)$ is forced to zero then the expected value of fitness will be $fit = 10$.

4.2 Velocity and position update law

In PSO algorithm, a population of particles is randomly generated initially; each particle adjusts self-position with velocity according to its own experience and the experiences of other particles. The particle velocity and position update law is adopted as [16]

$$\begin{aligned} v_q^l(n+1) = {} & v_q^l(n) \times iw + \xi_1 \times rand_1(\cdot) \times [L_{best_q}^l - p_q^l(n)] \\ & + \xi_2 \times rand_2(\cdot) \times [G_{best_q}^l - p_q^l(n)] \\ & + \xi_3 \times rand_3(\cdot) \times [S_{best_q}^l - p_q^l(n)] \end{aligned} \tag{28}$$

$$p_q^l(n+1) = p_q^l(n) + v_q^l(n+1)$$

where iw is called the inertia weight which balances the global and local search, $v_q^l(n)$ and $p_q^l(n)$ denote current velocity and current position, respectively; $rand_1(\cdot)$, $rand_2(\cdot)$ and $rand_3(\cdot)$ denote random variables between 0 and 1; ξ_1, ξ_2, and ξ_3 denote acceleration factor_1, acceleration factor_2 and acceleration factor_3, respectively; $L_{best_q}^l$, $G_{best_q}^l$, and $S_{best_q}^l$ are the personal best index, the global best index, and the sub-population best index of the qth particle, respectively. Additionally, $q = 1, 2, \ldots, n_p$, in which n_p is the population size and $\ell = 1, 2, \ldots, n_d$, in which n_d is the dimension of each particle and iw is given by

$$iw = iw_{\max} - \frac{iw_{\max} - iw_{\min}}{N_{\max}} \times N_n \tag{29}$$

where $iw_{\max}$, $iw_{\min}$, $N_{\max}$ and N_n are iteration maximum value, iteration minimum value, total iteration number and current iteration number of inertia weight.

This PSO algorithm is used to on-line tune the learning-rates η_P, η_I and η_D in Eqs. 14–16 to achieve fast learning speed of PID gains.

5 Simulation results

Two uncertain nonlinear systems, a two-link manipulator control system and a unified chaotic system are examined to illustrate the effectiveness of the proposed design method. The parameters of PSO are set as: the population size $z = 20$, the dimension of the particle $h = 2$, the acceleration factors $\xi_1 = 0.75$, $\xi_2 = 3.25$, $\xi_3 = 0.1$, the total iteration number $N_{\max} = 10$, the iteration maximum value $iw_{\max} = 0.9$, the iteration minimum value $iw_{\min} = 0.4$, the initial states of velocity $v_q^l(n)$ and position $p_q^l(n)$ of each particle are randomly generated.

Example 1. Two-link manipulator system

In this example, the proposed control system is applied to control a two-link robot manipulator. Figure 3 depicts this two-link manipulator, the angles of the second and the third links were considered to be θ_1 and θ_2, respectively

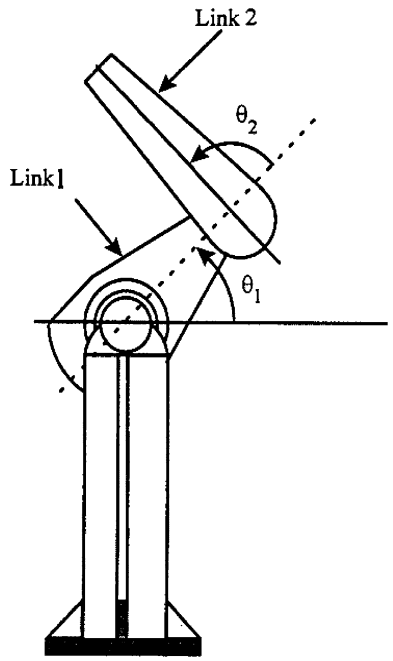

Fig. 3 The two-link manipulator system

[18]. In addition, the numerical values of parameters of the robot model were specified as that in [19]. The dynamic equation is given as follows

$$\boldsymbol{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \boldsymbol{A}_m(\boldsymbol{\theta},\dot{\boldsymbol{\theta}})\dot{\boldsymbol{\theta}} + \boldsymbol{B}(\boldsymbol{\theta}) + \boldsymbol{Z}\dot{\boldsymbol{\theta}} + \tau_d = \tau \tag{30}$$

where $\boldsymbol{\theta} \in \Re^n$ is the joint position vector; $\boldsymbol{M}(\boldsymbol{\theta}) \in \Re^{n\times n}$ is a symmetric positive definite inertia matrix; $\boldsymbol{A}_m(\boldsymbol{\theta},\dot{\boldsymbol{\theta}})$ is a vector of Coriolos and centripetal torques; $\boldsymbol{B}(\boldsymbol{\theta}) \in \Re^n$ representing the gravitational torques; $\boldsymbol{Z} = \boldsymbol{K}_\omega + \boldsymbol{V}_f \in \Re^{n\times n}$ is a diagonal matrix consisting of the back emf coefficient matrix $\boldsymbol{K}_\omega$ and the viscous friction coefficient matrix $\boldsymbol{V}_f$; $\tau_d \in \Re^n$ is the unmodeled disturbances vector; $\tau \in \Re^n$ is the vector of control input torques. By defining the state vector $\underline{x}(t) = [x_1(t), x_2(t)]^T = [\theta_1, \theta_2]^T$, the dynamic Eq. 30 can be expressed as

$$\ddot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t)) + \boldsymbol{G}(\boldsymbol{x}(t))\,\boldsymbol{u}(t) + \boldsymbol{d}(t) \tag{31}$$

where the unknown nonlinear function $\boldsymbol{G}(\boldsymbol{x}(t)) = \boldsymbol{M}^{-1}(\boldsymbol{\theta})$, and $\boldsymbol{f}(\underline{\boldsymbol{x}}(t)) = \boldsymbol{M}^{-1}(\boldsymbol{\theta}) - \left[\boldsymbol{A}_m(\boldsymbol{\theta},\dot{\boldsymbol{\theta}})\dot{\boldsymbol{\theta}} - \boldsymbol{B}(\dot{\boldsymbol{\theta}}) - \boldsymbol{Z}\dot{\boldsymbol{\theta}} - \tau_d\right]$, where $\boldsymbol{A}_m(\boldsymbol{\theta},\dot{\boldsymbol{\theta}}) = \sin(\theta_2)(a_2 + p_2 a_9)\begin{bmatrix} -\dot{\theta}_2 & -(\dot{\theta}_1+\dot{\theta}_2) \\ \dot{\theta}_1 & 0 \end{bmatrix}$, $\boldsymbol{B}(\boldsymbol{\theta}) = \begin{bmatrix} a_2 & 0 \\ 0 & a_8 \end{bmatrix}$, $\boldsymbol{Z} = [a_5\cos(\theta_1) + a_6\cos(\theta_1+\theta_2) + p_4\cos(\theta_1) + p_5\cos(\theta_1+\theta_2) a_9 a_6\cos(\theta_1+\theta_2) + p_5\cos(\theta_1+\theta_2) a_9], \boldsymbol{M} = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}$

where $M_{11} = a_1 + 2a_2\cos(\theta_2) + (p_1 + 2p_2\cos(\theta_2)a_9)$, $M_{12} = (a_3 + a_2\cos(\theta_2)) + (p_3 + p_2\cos(\theta_2)a_9)$, $M_{21} = (a_3 + a_2\cos(\theta_2)) + (p_3 + p_2\cos(\theta_2)a_9)$, $M_{22} = a_1 + p_3 a_9$ [19].

In this example, the initial conditions of the states are given as $a_1 = 6.33, a_2 = 0.14, a_3 = 0.11, a_4 = 27.6, a_5 = 31.9, a_6 = 3.3, a_7 = 0.94, a_8 = 4.54, a_9 = 1.25, p_1 = 0.37, p_2 = 0.18, p_3 = 0.18, p_4 = 4.23, p_5 = 4.15, \theta_1 = 50\pi/180, \theta_2 = 10\pi/180, \dot{\theta}_1 = 0, \dot{\theta}_2 = 0$.

For demonstrating the tracking performance of the proposed control system, the desired trajectories for $x_{d1}(t)$ and $x_{d2}(t)$ are set as

$$\underline{x}_d(t) = \begin{bmatrix} x_{d1}(t) \\ x_{d2}(t) \end{bmatrix} = \begin{bmatrix} 0.5 + 0.2\,(\sin(t) + \sin(2t)) \\ 1.3 - 0.1\,(\sin(t) + \sin(2t)) \end{bmatrix} \tag{32}$$

where $\underline{\boldsymbol{x}}_d(t) = [x_{d1}(t), x_{d2}(t)]^T = [\theta_{d1}, \theta_{d2}]^T$ is the reference trajectory vector.

For the proposed RSPID control system, the feedback gain matrix is designed as $\underline{\boldsymbol{H}} = diag(40, 4)$; the initial conditions of the learning-rates are chosen as $\eta_P = 400$, $\eta_I = 300$ and $\eta_D = 400$, respectively; all the PID gains are set as zero initially; the initial states of two-link manipulator are specified as $x_1(0) = 20\pi/180, x_2(0) = 10\pi/180, \dot{x}_1(0) = 0$ and $\dot{x}_2(0) = 0$, and the initial states of desired trajectories $x_{d1}(0) = 0, x_{d2}(0) = 0, \dot{x}_{d1}(0) = 0$ and $\dot{x}_{d2}(0) = 0$. In order to study the robustness of the proposed control system, assume that the two-link manipulator control system has external disturbance $\tau_d = [40, 25]$ at $t = 5$ s. Besides, the attenuation level is chosen as $\boldsymbol{R} = diag(0.5, 0.5)$

In order to compare the control performance, the adaptive fuzzy control (AFC) presented in [18] is also applied to this manipulator system. Using this control system, the tracking trajectories are shown in Fig. 4a, b, respectively. The simulation results of the proposed RSPID control system for this two-link manipulator control system are shown in Figs. 4, 5, 6, 7. Figure 4c, d represents the tracking responses of $x_1(t)$ and $x_2(t)$ by the proposed RSPID control scheme, respectively. Moreover, the control inputs and tracking errors of $x_1(t)$ and $x_2(t)$ are plotted in Fig. 5a–d, respectively. In addition, the fitness function and learning-rates are shown in Fig. 6a–d, respectively. The PID gains $\boldsymbol{K}_P$, $\boldsymbol{K}_I$ and $\boldsymbol{K}_D$ are shown in Fig. 7a–c, respectively. Comparing Fig. 4c, d with Fig. 4a, b, it can be seen the tracking error have been much reduced by using the RSPID controller. Moreover, from Fig. 6, it is also seen that the learning-rates of PID controller converge after the 5th second. The simulation results also show that the proposed RSPID control system can effectively achieve parameter tuning and favorable control for the two-link manipulator control system.

Example 2. Unified chaotic system

Consider a general master–slave unified chaotic systems; the master system and slave system are given as [20].

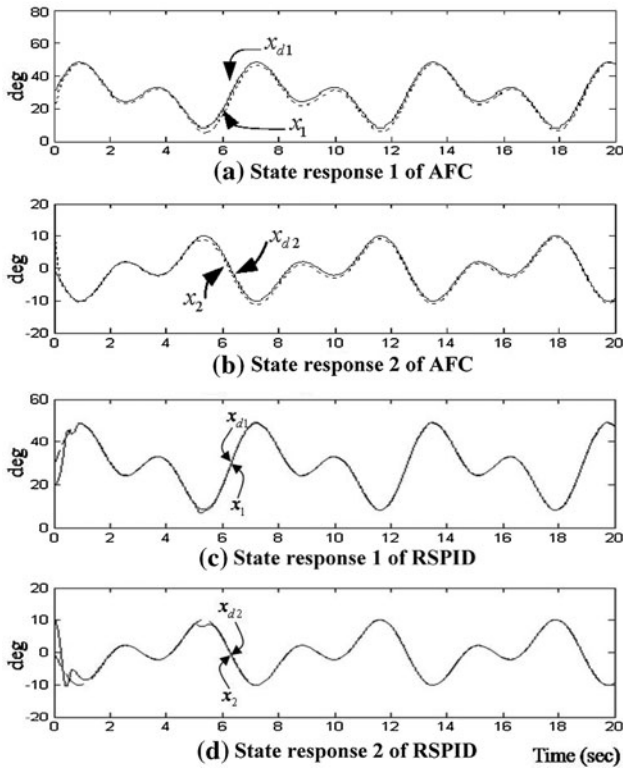

Fig. 4 The state responses of AFC and RSPID. **a** state response 1 of AFC, **b** state response 2 of AFC, **c** state response 1 of RSPID, **d** state response 2 of RSPID (*dashed line* desired trajectory, *continuous line* state trajectory)

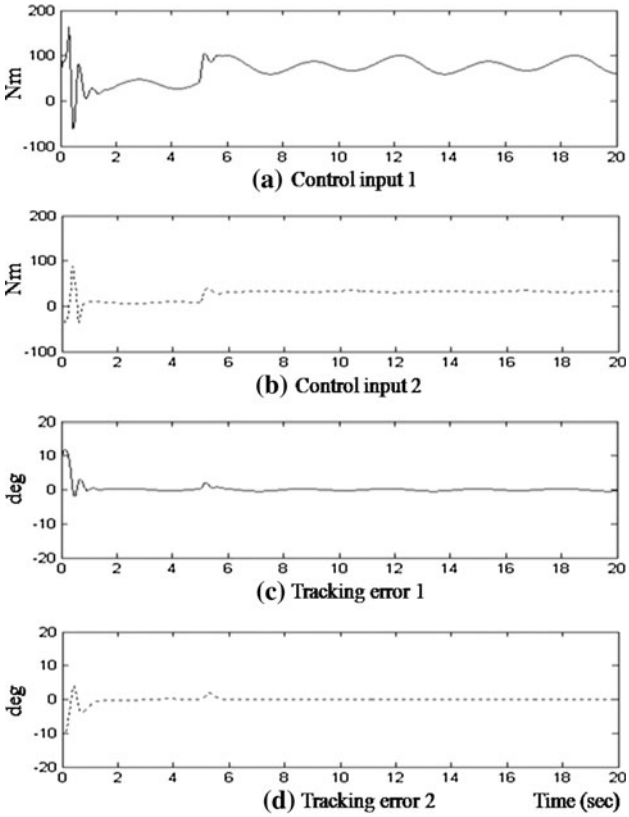

Fig. 5 The control inputs and tracking errors of the two-link manipulator system **a** control input 1, **b** control input 2, **c** tracking error 1, **d** tracking error 2

$$\begin{aligned}
&\dot{x}_{d1}(t) = (25\theta_c + 10)(x_{d2}(t) - x_{d1}(t))\\
&\text{Master } \dot{x}_{d2}(t) = (28 - 35\theta_c)x_{d1}(t)\\
&\qquad\qquad - x_{d1}(t)x_{d3}(t) + (29\theta_c - 1)x_{d2}(t)\\
&\dot{x}_{d3}(t) = x_{d1}(t)x_{d2}(t) - \left(\frac{8+\theta_c}{3}\right)x_{d3}(t)
\end{aligned} \tag{33}$$

$$\begin{aligned}
&\dot{x}_1(t) = (25\theta_c + 10)(x_2(t) - x_1(t)) + d_1(t) + u_1(t)\\
&\text{Slave } \dot{x}_2(t) = (28 - 35\theta_c)x_1(t) - x_1(t)x_3(t)\\
&\qquad\qquad + (29\theta_c - 1)x_2(t) + d_2(t) + u_2(t)\\
&\dot{x}_3(t) = x_1(t)x_2(t) - \left(\frac{8+\theta_c}{3}\right)x_3(t) + d_3(t) + u_3(t)
\end{aligned} \tag{34}$$

where x_{di} (t) and $x_i(t)$ are the system states variables of master system and slave system, respectively; d_i, $i = 1, 2, 3$ denote the disturbances and u_i, $i = 1, 2, 3$ are the control inputs. Assume $\alpha_1 = (25\theta_c + 10), \alpha_2 = (28 - 35\theta_c), \alpha_3 = (29\theta_c - 1)$ and $\alpha_4 = \left(\frac{8+\theta_c}{3}\right)$; then the master system Eq. 33 and the slave system Eq. 34 can be rewritten as

$$\begin{aligned}
\dot{x}_{d1}(t) &= \alpha_1(x_{d2}(t) - x_{d1}(t))\\
\dot{x}_{d2}(t) &= \alpha_2 x_{d1}(t) - x_{d1}(t)x_{d3}(t) + \alpha_3 x_{d2}(t)\\
\dot{x}_{d3}(t) &= x_{d1}(t)x_{d2}(t) - \alpha_4 x_{d3}(t)
\end{aligned} \tag{35}$$

$$\begin{aligned}
\dot{x}_1(t) &= \alpha_1(x_2(t) - x_1(t)) + d_1(t) + u_1(t)\\
\dot{x}_2(t) &= \alpha_2 x_1(t) - x_1(t)x_3(t) + \alpha_3 x_2(t) + d_2(t) + u_2(t)\\
\dot{x}_3(t) &= x_1(t)x_2(t) - \alpha_4 x_3(t) + d_3(t) + u_3(t)
\end{aligned} \tag{36}$$

the master–slave system can be expressed as

$$\dot{\boldsymbol{x}}_d(t) = \boldsymbol{f}(\boldsymbol{x}_d(t)) \tag{37}$$

and

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t)) + \boldsymbol{G}(\boldsymbol{x}(t))\boldsymbol{u}(t) + \boldsymbol{d}(t) \tag{38}$$

where $\boldsymbol{x}_d(t) = [x_{d1}(t), x_{d2}(t), x_{d3}(t)]^T, \boldsymbol{x}(t) = [x_1(t), x_2(t), x_3(t)]^T$, $\boldsymbol{f}(\boldsymbol{x}_d(t)) = [\alpha_1(x_{d2}(t) - x_{d1}(t)), \alpha_2 x_{d1}(t) - x_{d1}(t)x_{d3}(t) + \alpha_3 x_{d2}(t), x_{d1}(t)x_{d2}(t) - \alpha_4 x_{d3}(t)]^T$, $\boldsymbol{f}(\boldsymbol{x}(t)) = [\alpha_1(x_2(t) - x_1(t)), \alpha_2 x_1(t) - x_1(t)x_3(t) + \alpha_3 x_2(t), x_1(t)x_2(t) - \alpha_4 x_3(t)]^T$, $\boldsymbol{G}(\boldsymbol{x}(t)) = diag[1, 1, 1]$, $\boldsymbol{d}(t) = [d_1(t), d_2(t), d_3(t)]^T$ and $\boldsymbol{u}(t) = [u_1(t), u_2(t), u_3(t)]^T$.

The control objective is to find a suitable control law $\boldsymbol{u}(t)$, so that the state trajectories of slave chaotic system $\boldsymbol{x}(t)$ can follow the master chaotic system $\boldsymbol{x}_d(t)$ under different initial conditions and subject to disturbances.

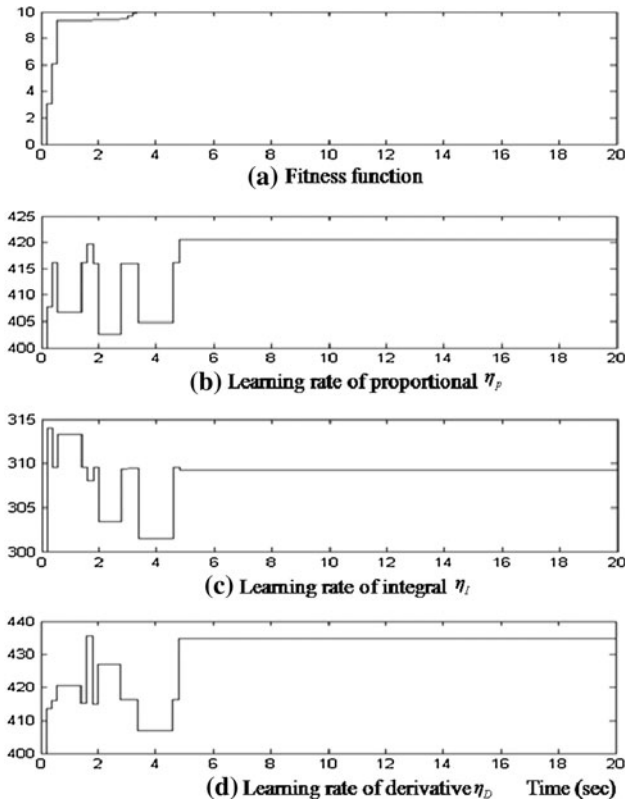

Fig. 6 The fitness function and learning-rates of SPID **a** fitness function, **b** learning rate η_P, **c** learning rate η_I, **d** learning rate η_D

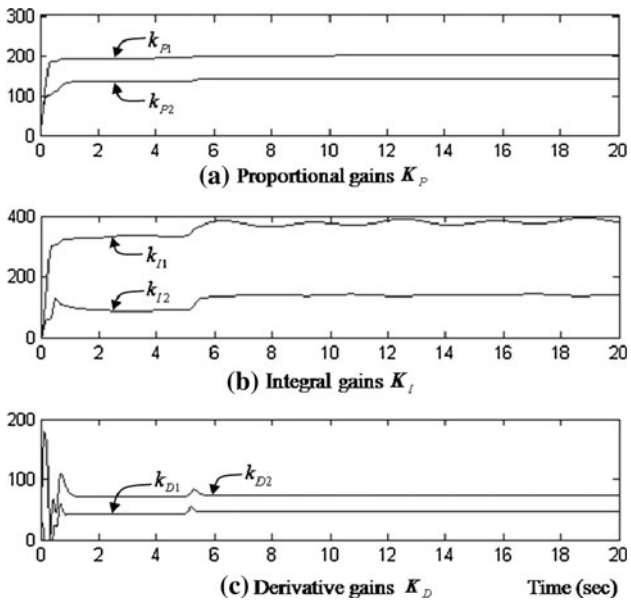

Fig. 7 $\boldsymbol{K}_P$, $\boldsymbol{K}_I$ and $\boldsymbol{K}_D$ gains of SPID controller for the two-link manipulator system

When $\theta_c = [0 \sim 0.8)$ the system is known to be the generalized Lorenz system; when $\theta_c = 0.8$ the system is called the Lu system; and when $\theta_c = (0.8 \sim 1]$ the system is called the Chen system. It is supposed that the initial

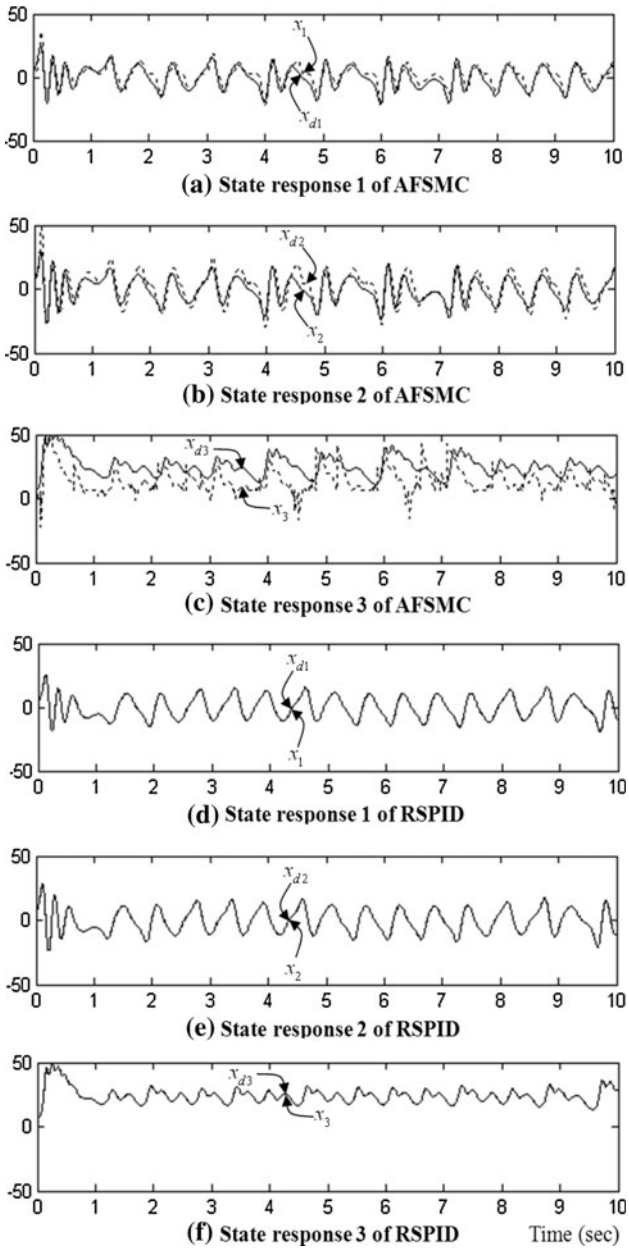

Fig. 8 The synchronization responses of a chaotic system. (*dashed line* drive system trajectory, *continuous line* response system trajectory)

conditions of the states are $x_{d1}(0) = 3$, $x_{d2}(0) = 5$, $x_{d3}(0) = 7$ for the master system, and $x_1(0) = -2$, $x_2(0) = 2$, $x_2(0) = 3$ for the salve system. In addition, $\theta_c = 1$and the external disturbance $\boldsymbol{d}(t) = [0.2\cos(\pi t), 0.1\cos(t), 0.3\cos(2t)]^T$ are used. The attenuation level is chosen as $\boldsymbol{R} = diag(0.6, 0.6, 0.6)$ and the gains of integrated error function are selected as $\underline{\boldsymbol{H}} = diag(0.1, 0.1, 0.1)$. In addition, the initial values of the learning-rates of PID gains are set as $\eta_P = 80, \eta_I = 20$, and $\eta_D = 0$, respectively; all the PID gains are set as zero initially.

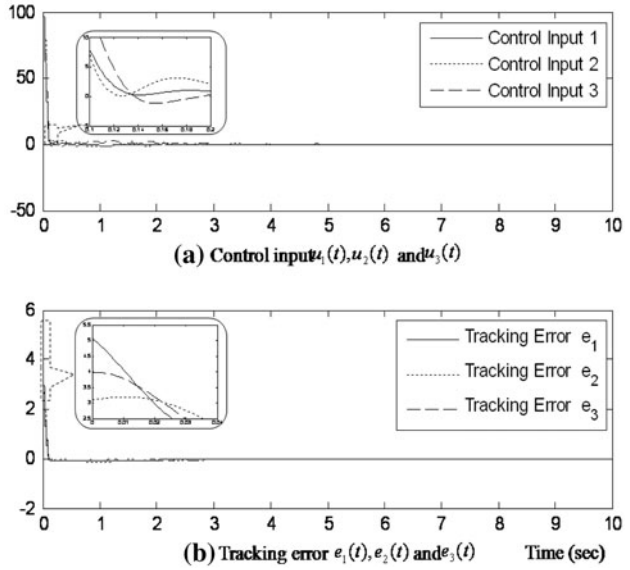

Fig. 9 The synchronization errors and control efforts of unified chaotic system

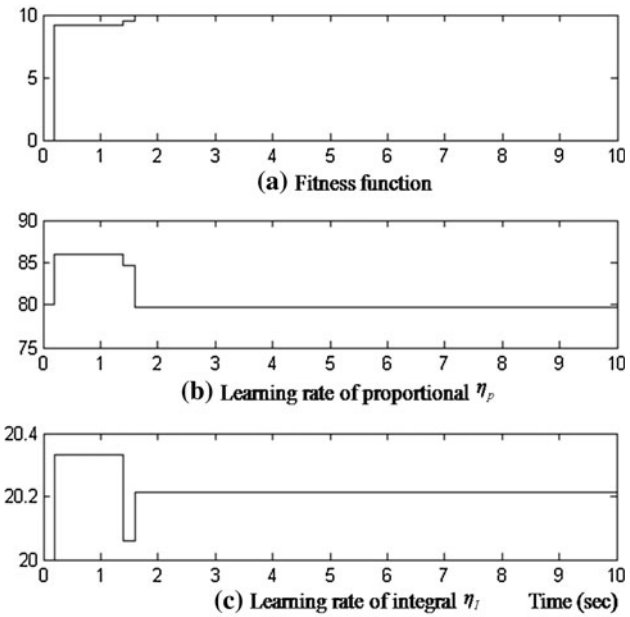

Fig. 10 The fitness function and learning-rates of SPID

For comparison, an adaptive fuzzy sliding model control (AFSMC) [21] is also applied to this system, the tracking trajectories are shown in Fig. 8a–c, respectively. The simulation results of the proposed RSPID control are shown in Figs. 8, 9, 10, 11. Synchronization responses of $x_{d1}(t), x_1(t), x_{d2}(t), x_2(t), x_{d3}(t)$ and $x_3(t)$ are depicted in Figs. 8d–f, respectively. Figure 9 shows the synchronization errors and control efforts of chaotic system. The fitness function and learning-rates are shown in Fig. 10, it is shown when error state $\underline{\boldsymbol{e}}(t)$ converges to zero, the fitness

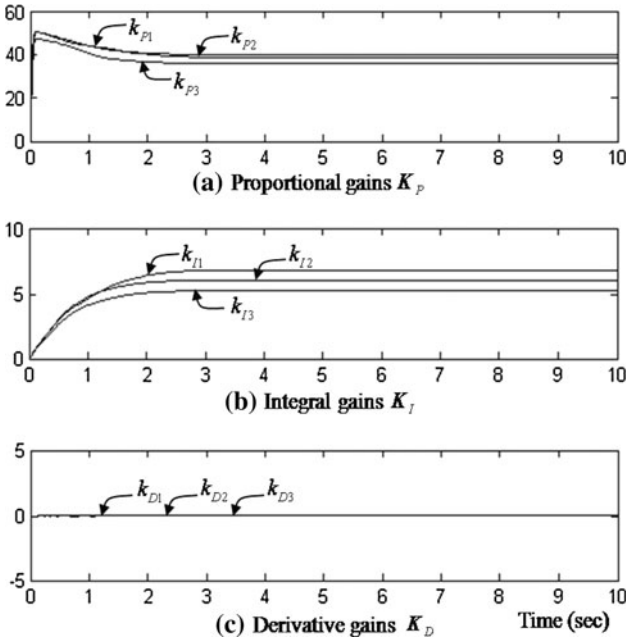

Fig. 11 $\boldsymbol{K}_P$, $\boldsymbol{K}_I$ and $\boldsymbol{K}_D$ gains of SPID controller for unified chaotic system

function also approximates to $fit = 10$. Figure 11 illustrates the $\boldsymbol{K}_P$, $\boldsymbol{K}_I$ and $\boldsymbol{K}_D$ gains. Since this system is a first order system (i.e. PI controller is used); thus, $\boldsymbol{K}_D$ gains approach to zero.

From Figs. 4 and 8, it's easy to find that the proposed control system can achieve control performance better than AFSMC scheme, even under disturbance. By using the proposed design method, the tracking error can converge to zero quickly even in the presence of disturbance for different initial conditions.

6 Conclusion

This paper proposes a nonlinear system control scheme by using a robust self-learning PID (RSPID) control system. In the proposed control system, the self-learning PID (SPID) controller is the main controller used to mimic an ideal controller, and the robust controller is designed to compensate for the difference between the ideal controller and the SPID controller based on H_∞ control technology. In addition, a PSO algorithm has been adopted to search the optimal learning-rates of the tuning law in SPID controller. Simulation results have demonstrated that the proposed RSPID control system can achieve favorable control performance for the nonlinear systems. The major contributions of this study are: (1) The successful development of a self-learning PID controller. (2) The SPID controller with PSO algorithm and the robust controller are successful combined for MIMO nonlinear systems.

References

1. Astrom KJ, Hagglund T (1988) Automatic tuning of PID controllers. Instrument Society of America, Research Triangle Park
2. Ho WK, Hang CC, Zhou JH (1997) Self-tuning PID control of a plant with under-damped response with specifications on gain and phase margins. IEEE Trans Control Syst Technol 5(4):446–452
3. Ren TJ, Chen TC, Chen CJ (2008) Motion control for a two-wheeled vehicle using a self-tuning PID controller. Control Eng Pract 16:365–375
4. Vagia M, Tzes A (2008) Robust PID control design for an electrostatic micromechanical actuator with structured uncertainty. IET Control Theory Appl 2(5):365–373
5. Yu DL, Chang TK, Yu DW (2005) Fault tolerant control of multivariable processes using auto-tuning PID controller. IEEE Trans Syst Man Cybern B 35(1):32–43
6. Juang JG, Huang MT, Lin WK (2008) PID control using presearched genetic algorithms for a MIMO system. IEEE Trans Syst Man Cybern C 38(5):716–727
7. Wang J, Zhang C, Jing Y (2008) Fuzzy immune self-tuning PID control of HVAC system. IEEE Int Conf Mech Autom 678–683
8. Bandyopadhyay R, Chakraborty UK, Patranabis D (2001) Auto-tuning a PID controller: a fuzzy-genetic approach. J Syst Archit 47:663–673
9. Harinath E, Mann GKI (2008) Design and tuning of standard additive model based fuzzy PID controllers for multivariable process systems. IEEE Trans Syst Man Cybern B 38(3):667–674
10. Kennedy J, Eberhart RC (1995) Particle swarm optimization. IEEE Int Conf Neural Netw Perth, Australia 4:1942–1948
11. Eberhart RC, Shi Y (2001) Particle swarm optimization: developments, applications and resources. In: Proceedings of congress on evolution computation, Seoul, pp 81–75
12. Juang CF (2004) A hybrid of genetic algorithm and particle swarm optimization for recurrent network design. IEEE Trans Syst Man Cybern B Cybern 34(2):997–1006
13. Heo JS, Wang K, Lee Y, Garduno-Ramirez R (2006) Multiobjective control of power plants using particle swarm optimization techniques. IEEE Trans Energy Convers 21(2):552–561
14. Lin FJ, Teng LT, Chu H (2008) A robust recurrent wavelet neural network controller with improved particle swarm optimization for linear synchronous motor drive. IEEE Trans Power Electron 23(6):3067–3078
15. Kim TH, Maruta I, Sugie T (2008) Robust PID controller tuning based on the constrained particle swarm optimization. Automatica 44:1104–1110
16. Chang WD, shih SP (2010) PID controller design of nonlinear systems using an improved particle swarm optimization approach. Commun Nonlinear Sci Numer Simul 15:3632–3639
17. Chen BS, Lee CH (1996) H^{∞} tracking design of uncertain nonlinear SISO systems: adaptive fuzzy approach. IEEE Trans Fuzzy Syst 4(1):32–43
18. Wang SD, Lin CK (2000) Adaptive tuning of the fuzzy controller for robots. Fuzzy Sets Syst 110:351–363
19. Erlic M, Lu WS (1995) A reduced-order adaptive velocity observer for manipulator control. IEEE Trans Robot Autom 11(2):293–303
20. Lu J, Chen G, Cheng D (2002) Bridge the gap between the Lorenz system and the Chen system. Int J Bifurcat Chaos 12:2917–2926
21. Lin CM, Hsu CF (2004) Adaptive fuzzy sliding-mode control for induction servomotor systems. IEEE Trans Energy Convers 19(2):362–368