International Journal of Innovative Computing, Information and Control
Volume **7**, Number **12**, December **2011**

ICIC International ©2011 ISSN 1349-4198
pp. 7091–7103

DESIGN OF PSO FUZZY NEURAL NETWORK CONTROL FOR BALL AND PLATE SYSTEM

Xiucheng Dong[1], Yunyuan Zhao[1], Yunyun Xu[1], Zhang Zhang[1]
and Peng Shi[2,3]

[1]Provincial Key Lab on Signal and Information Processing
Xihua University
Chengdu 610039, P. R. China
dxc136@163.com

[2]Department of Computing and Mathematical Sciences
University of Glamorgan
Pontypridd CF37 1DL, UK

[3]School of Engineering and Mathematics
Victoria University
Melbourne, VIC 8001, Australia
pshi@glam.ac.uk

Received November 2010; revised March 2011

Abstract. *The ball and plate system is a typical multi-variable plant, which is the extension of the traditional ball and beam problems. Particle swarm optimization algorithm fuzzy neural network control (PSO-FNNC) scheme is introduced for the ball and plate system. The fuzzy neural network (FNNC) is optimized by the offline particle swarm optimization (PSO) of global searching ability, and the online radius basis function (RBF) algorithm ability of local searching. Then, the optimized fuzzy RBF neural network (FRBF) tuned PID controller. The simulation results demonstrate the potential of the proposed technique, especially tracking speed, tracking accuracy and robustness, is improved obviously, which embodies the nice characters of the PSO-FNNC scheme.*

Keywords: Ball and plate, Fuzzy neural network, PSO algorithm, PID

1. Introduction.

The ball and plate system is an extension of the classical ball and beam system [1,2]. The system consists of a plate pivoted at its centre such that the slope of the plate can be manipulated in two perpendicular directions. A servo system consists of motor controller card and two servo motors to title the plate. Intelligent vision system is used for measurement of a ball position from a CCD camera. The problem of the motion control of this system is to control the position of a ball on a plate for both static positions and desired paths. The slope of the plate can be manipulated in two perpendicular directions, so that the tilting of the plate will make the ball move on the plate.

The ball on the hard plate is an unconstrained object which is able to move freely and which has no ability to recognize the environment. So, the unconstrained object cannot control its behavior by itself. This makes motion control of ball and plate system difficult.

Stabilizing control of the ball and plate system is to hold the ball in a specific position on the plate. Trajectory tracking control demands the ball to follow the given position reference. Stabilizing and trajectory tracking control of this system have been studied separately.

Since the system is inherently nonlinear, the mathematical model is difficult to be derived. A practical result on static position control as well as smooth paths tracking was given in [3]. Ball position was measured by a touch pad rather than a camera. A linear quadratic state feedback regulator was designed after the system was linearized around a few operating points. The accuracy of stabilization control was 5(mm). Steady error in average was 18(mm) in circle while given tracking velocity was less than 4.2(mm/s). A real time toolbox was developed on the ball and plate device CE 151 [4]. A fuzzy controller with online learning capability was reported also on CE 151 [5]. A hierarchical fuzzy control scheme with a T-S type fuzzy tracking controller was proposed in [6]. The fuzzy controller was built based on local linearized models in different state space regions. Instead of fuzzy logic, a sliding mode controller was utilized to cope with system uncertainty [7]. A trajectory tracking experiment system was constructed based on single input rule modules (SIRMs) fuzzy logic [8]. Maximum tracking error of circles was less than 50(mm). On the ball and plate experimental device BPVS-JLUI, Tian proposed a nonlinear switching control scheme for the tracking problem [9].

In this paper, the fuzzy neural network is applied in the ball and plate control. The fuzzy neural network has the fuzzy logic reasoning ability and the neural network learning ability. The physics meaning of the fuzzy neural network is clear and intelligible. In addition, PSO is introduced to neural fuzzy system to enhance its learning ability and optimize its structure. However, similar with other global optimization algorithms, such as genetic algorithm (GA), PSO also exists in the phenomenon of premature convergence, especially in the complex multi-peak searching problems. And if PSO is adopted to optimize the whole model, the computing time will be enormously long. Therefore, the PSO combining with genetic algorithm is adopted and we divide the learning of FNNC into two phases. In the first learning, PSO is used to train the FNNC and an optimized model will be constructed based on the training data. In the second learning, a phase of deep and careful updating and RBF algorithm is chosen to update the system parameters on the basis of first learning stage to improve its precision and reduce the computation time.

RBF neural network has good performance. It has proved that RBF network can approximate any arbitrary nonlinear curve at arbitrary precision. And RBF has global approximation ability, and solves local optimal problem of the BP network fundamentally. In addition, its topology is very compact, structure parameters can be achieved, and separate learning and convergence speed are quick. The biggest advantage of adopting RBF network to design fuzzy neural network is that RBF network has equivalence to fuzzy inference process. The equivalence unifies these two different structural systems in the function, so that operators and parameters of the network will have a clear physical meaning, and at the same time, a suitable algorithm is found for the fuzzy adaptive control.

Theorem 1.1. *Normalized RBF network and fuzzy inference system have the function equivalence when they met the following conditions: 1) The number of center value of RBF network is equal to the number of "IF THEN" rules of fuzzy inference system; 2) Membership functions of every fuzzy subset in the fuzzy inference system select Gaussian function; 3) Fuzzy inference adopts T-norm; 4) RBF network and fuzzy inference system adopt the same total output form.*

The proposed FRBF based on the PSO algorithm possesses a good learning capacity, a shorter learning time than pure PSO, and less sensitivity to the problem of local extremes than neural network.

2. **Mathematical Model of Ball and Plate System.** The structure of the ball and plate system is shown in Figure 1. The model has four degrees of freedom, two at ball moving and two at plate inclination. With derivation of the Euler-Lagrange's equation we get the system of nonlinear differential equations who are mathematically describing system dynamic:

$$\left(m+\frac{I_b}{r^2}\right)\ddot{x}-mx\dot{\alpha}^2-my\dot{\alpha}\dot{\beta}+mg\sin\alpha=0 \tag{1}$$

$$\left(m+\frac{I_b}{r^2}\right)\ddot{y}-my\dot{\beta}^2-mx\dot{\alpha}\dot{\beta}+mg\sin\beta=0 \tag{2}$$

$$\left(I_p+I_b+mx^2\right)\ddot{\alpha}+2mx\dot{x}\dot{\alpha}+mxy\ddot{\beta}+m\dot{x}y\dot{\beta}+mx\dot{y}\dot{\beta}+mgx\cos\alpha=\tau_x \tag{3}$$

$$\left(I_p+I_b+mx^2\right)\ddot{\beta}+2my\dot{y}\dot{\beta}+mxy\ddot{\alpha}+m\dot{x}y\dot{\alpha}+mx\dot{y}\dot{\alpha}+mgy\cos\beta=\tau_y \tag{4}$$

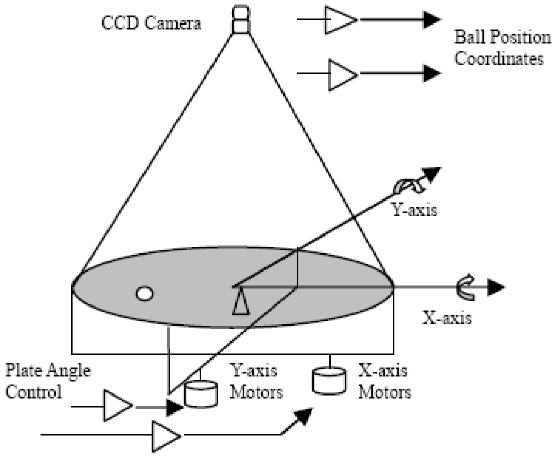


FIGURE 1. The structure of the ball and plate system

System variables are selected as following: x is the displacement of the ball along the x-axis, y is the displacement of the ball along the y-axis, α is the angle between x-axis of the plate and horizontal plane, β is the angle between y-axis of the plate and horizontal plane, τ_x is the torque exerted on the plate in x-axis, τ_y is the torque exerted on the plate in the y-axis. m is the mass of the ball, r is the radius of the ball, I_b is the ball inertia and I_p is the plate inertia. Usually, $|\alpha| < 5^\circ$, $|\beta| < 5^\circ$. Parameters of the ball and plate system are shown in Table 1.

Define $B = m/(m + I_b/r^2)$. Define two new inputs u_x and u_y using the invertible nonlinear transformations (5) and (6):

$$\left(I_p+I_b+mx^2\right)u_x+2mx\dot{x}\dot{\alpha}+mxy\ddot{\beta}+m\dot{x}y\dot{\beta}+mx\dot{y}\dot{\beta}+mgx\cos\alpha=\tau_x \tag{5}$$

$$\left(I_p+I_b+mx^2\right)u_y+2my\dot{y}\dot{\beta}+mxy\ddot{\alpha}+m\dot{x}y\dot{\alpha}+mx\dot{y}\dot{\alpha}+mgy\cos\beta=\tau_y \tag{6}$$

Then, the system can be modeled in the state-space form as follows:

$$\dot{X}=\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\\ \dot{x}_4\\ \dot{x}_5\\ \dot{x}_6\\ \dot{x}_7\\ \dot{x}_8\end{bmatrix}=\begin{bmatrix}x_2\\ B\left(x_1x_4^2+x_4x_5x_8-g\sin x_3\right)\\ x_4\\ 0\\ x_6\\ B\left(x_5x_8^2+x_1x_4x_8-g\sin x_7\right)\\ x_8\\ 0\end{bmatrix}+\begin{bmatrix}0&0\\0&0\\0&0\\1&0\\0&0\\0&0\\0&0\\0&1\end{bmatrix}\begin{bmatrix}u_x\\ u_y\end{bmatrix} \tag{7}$$

Table 1. Parameters of the ball and plate system

Parameters	*Description*	*Numerical Values*
m	*mass of the ball*	*0.038kg*
r	*radius of the ball*	*0.015m*
I_P	*Plate inertia*	*1.6kgm^2*
I_b	*Ball inertia*	$4.2 * 10^{-6} kgm^2$
g	*gravity acceleration*	$9.8m/s^2$
x	*position of the ball in the x-axis*	m
y	*position of the ball in the y-axis*	m
$\dot{x}$	*velocity of the ball in the x-axis*	m/s
$\dot{y}$	*velocity of the ball in the y-axis*	m/s
α	*angle of the plate from the x-axis*	*Arc*
β	*angle of the plate from the y-axis*	*Arc*

where $X = (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8)^T = \left(x, \dot{x}, \alpha, \dot{\alpha}, y, \dot{y}, \beta, \dot{\beta}\right)^T$, $Y = (x_1, x_5)^T = (x, y)^T$.

3. Control System Structure and Design.

In the ball and plate system, it is supposed that the ball remains in contact with the plate and the rolling occurs without slipping, which imposes a constraint on the rotation acceleration of the plate. Because of the low velocity and acceleration of the plate rotation, the mutual interactions of both coordinates can be negligible. So, motion of ball on the plate can be divided into motion along the x-axis and motion along the y-axis. The two motions are regulated respectively:

$$\dot{X} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} x_2 \\ B\left(x_1 x_4^2 - g \sin x_3\right) \\ x_4 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u_x \tag{8}$$

$$\dot{Y} = \begin{bmatrix} \dot{x}_5 \\ \dot{x}_6 \\ \dot{x}_7 \\ \dot{x}_8 \end{bmatrix} = \begin{bmatrix} x_6 \\ B\left(x_5 x_8^2 - g \sin x_7\right) \\ x_8 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u_y \tag{9}$$

Namely, the ball-plate system can be regarded as two individual sub-systems and we can control both coordinates independently. Due to the symmetry of x direction and y direction, hereafter we will discuss x direction only.

The fuzzy neural network control system is shown in Figure 2.

3.1. Design of FRBF-PID.

The inputs including error e and error change rate e_c, need to be fuzzified according to fuzzy control rules. $e(k) = r(k) - y(k)$, $e_c(k) = e(k) - e(k-1)$. For the FRBF PID controller, universe of discourse for plate angle error e is chosen to be $[-140, 140]$. The universe of discourse for the plate angle error derivative e_c is $[-60, 60]$. The plate angle error e and the plate angle error derivative e_c is divided into 7 fuzzy subsets {NB(very small), NM(small), NS(slightly small), Z(zero), PS(slightly large), PM(large), PB(rather large)}. The outputs is the angle between x-axis of the plate and horizontal plane or between y-axis of the plate and horizontal plane, the universe of discourse for the expect angle is $[-6, 6]$.

For the fuzzy neutral network controller, the scaling factors were also obtained as $K_e = n/e = 0.05$, $K_{ec} = n/e_c = 0.1$, $K_u = U/n = 0.15$.

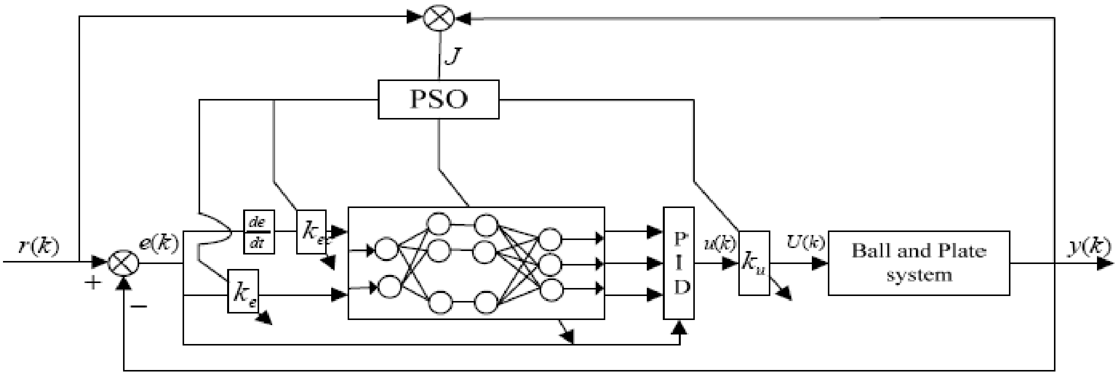


Figure 2. Fuzzy neural network PID controller based on PSO

Structure of FRBF neural network PID is shown in Figure 3. The designed network contains input layer, fuzzification layer, fuzzy inference layer and output layer. The relation between input and output of all networks is given as follows.

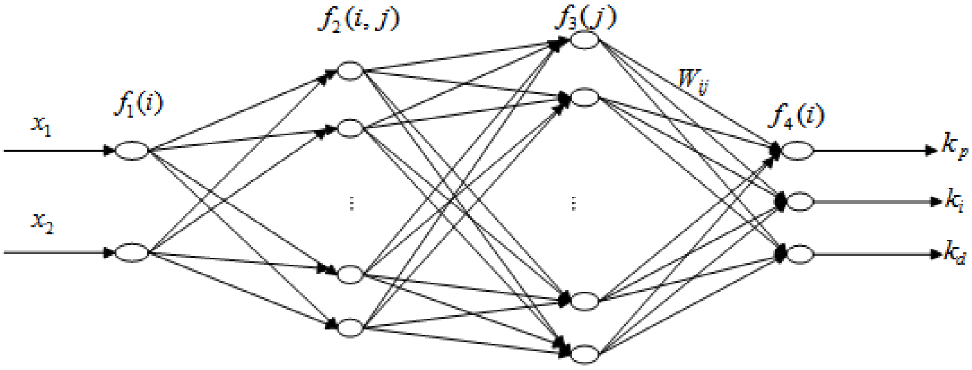


Figure 3. Structure of FRBF-PID

Layer 1: Input layer

Nodes at Layer 1 are input nodes that represent input linguistic variables. Here, error e and error change rate e_c are used as input variables. The relation between input and output are given as follows:

$$f_1(i) = x_i \quad i = 1, 2, \ldots, m, \quad m = 2 \tag{10}$$

where $x_1 = k_e \cdot e$, $x_2 = k_{ec} \cdot e_c$.

Layer 2: Fuzzification layer

Nodes at Layer 2 are term nodes that act as membership functions (MF) to represent the terms of the respective linguistic variable. Each input has 7 grades fuzzy linguistic variable, namely $n = 7$. Under the action of Gaussian MF, the operations performed in this layer are given as follows:

$$f_2(i,j) = \exp\left(net_j^2\right), \quad i = 1, 2, \ldots, m, \quad m = 2 \quad j = 1, 2, \ldots, n \tag{11}$$

$$net_j^2 = -\frac{(f_1(i) - c_{ij})^2}{(b_{ij})^2} \tag{12}$$

where c_{ij} and b_{ij} are the center and the width of the Gaussian function respectively.

Initialize membership of e and e_c as follows:

NB: $net_j^2 = -\left(x_1 - 9\right)^2/0.6^2$ $\quad net_j^2 = -\left(x_1 - 4.5\right)^2/0.6^2$
NM: $net_j^2 = -\left(x_1 - 6\right)^2/0.6^2$ $\quad net_j^2 = -\left(x_1 - 3\right)^2/0.6^2$
NS: $net_j^2 = -\left(x_1 - 3\right)^2/0.6^2$ $\quad net_j^2 = -\left(x_1 - 1.5\right)^2/0.6^2$
Z: $net_j^2 = -\left(x_1 + 0\right)^2/0.6^2$ $\quad net_j^2 = -\left(x_1 + 0\right)^2/0.6^2$
PS: $net_j^2 = -\left(x_1 + 3\right)^2/0.6^2$ $\quad net_j^2 = -\left(x_1 + 1.5\right)^2/0.6^2$
PM: $net_j^2 = -\left(x_1 + 2\right)^2/0.6^2$ $\quad net_j^2 = -\left(x_1 + 3\right)^2/0.6^2$
PB: $net_j^2 = -\left(x_1 + 3\right)^2/0.6^2$ $\quad net_j^2 = -\left(x_1 + 4.5\right)^2/0.6^2$

Layer 3: Fuzzy inference layer

Each node at Layer 3 represents one fuzzy logic rule. The various nodes achieve fuzzy operation; the output for each node is the product of the input signal of all nodes.

$$f_3(j) = \prod_{j=1}^{N} f_2(i,j) \quad i = 2 \tag{13}$$

$$N = \prod_{i=1}^{n} N_i \quad n = 7 \tag{14}$$

where N_i is the number of the ith input membership functions, that is, the number of nodes at fuzzification layer.

Layer 4: Output layer

The output $f_4(i)$ is three parameters of the PID controller.

$$f_4(i) = \sum_{j=1}^{N} W_{ij} \cdot f_3(j) \quad i = 1, 2, 3 \quad j = 1, 2, \ldots, N \tag{15}$$

where W_{ij} is the connection weight matrix between output nodes and nodes of Layer 3.

The controller of this system is designed as follows:

$$\Delta u(k) = k_p e(k) + k_i(e(k) - e(k-1)) + k_d(e(k) - 2e(k-1) + e(k-2)) \tag{16}$$

where $k_p = f_4(1)$, $k_i = f_4(2)$, $k_d = f_4(3)$.

The incremental PID control algorithm is adopted

$$U(k) = k_u(u(k-1) + \Delta u(k)) \tag{17}$$

3.2. **PSO optimization.** PSO is an evolutionary computing technology developed by Eberhart and Kennedy in 1995. PSO generally adopts the real-number encoding. This algorithm begins with initializing a group of random particles, and then finds the optimal solution through iteration. At each iteration, particles track two extreme values to update their own, one is the optimal solution called the individual extreme value p_b that particles themselves find, the other is the present optimal solution called the global extreme value g_b that the particle swarm finds. After finding these two extreme values above, the following two formulas are utilized to update their own speed and position.

$$V_{new} = w \cdot V_{cur} + c_1 \cdot rad \cdot (p_b - P_{cur}) + c_2 \cdot rad \cdot (g_b - P_{cur}) \tag{18}$$

$$P_{new} = P_{cur} + V_{new} \tag{19}$$

where V_{new} is particles' new speed, P_{new} is particles' new position; V_{cur} is particles' current speed, P_{cur} is particles' current position. rad is random number randomly generated within [0, 1], c_1 and c_2 are acceleration constants, w represents the inertia weight.

Similar with other global optimization algorithms, such as genetic algorithm (GA), PSO also exists the phenomenon of premature convergence, especially in the complex multi-peak searching problems. In order to overcome those disadvantages that PSO algorithm is easily trapped into local extreme value, convergence is slower at the later stage of evolution and others, many improved PSO algorithms were proposed. Mutation operator of GA is introduced into PSO to generate mutation when particles are trapped in local optima. Thus, change particles' speed and the way forward and help get rid of the bound that have been trapped into the local extreme value point at the later stage. So, particles can go into other areas to search, and so on. This improved algorithm can greatly improve the convergence speed and accuracy. At the beginning of PSO, individual's extreme value p_b of the entire particle swarm is evaluated; the average individual extreme value $\overline{p_b}$ is calculated as follows:

$$\overline{p_b} = \frac{1}{D}\sum_{i=1}^{D} p_b(i) \tag{20}$$

Mutation operation will be completed when p_b is worse than $\overline{p_b}$.

3.3. **The RBF algorithm online adjustment.** Objective function is defined as follows:

$$G = \frac{1}{2}(y_d - y_m)^2 \tag{21}$$

where y_m is the actual output, y_d is the desired output.

If the actual output y_m is desired to approach the desired output y_d, G must be the smallest. $\frac{\partial G}{\partial W_{ij}}$, $\frac{\partial G}{\partial c_{ij}}$ and $\frac{\partial G}{\partial b_{ij}}$ could be calculated through the error back propagation algorithm as follows:

$$\frac{\partial G}{\partial W_{ij}} = \frac{\partial G}{\partial y_m}\cdot\frac{\partial y_m}{\partial \Delta u}\frac{\partial \Delta u}{\partial f_4}\frac{\partial f_4}{\partial W_{ij}} = -(y_d - y_m)\cdot\frac{\partial y_m}{\partial \Delta u}\cdot\phi\cdot f_3 \tag{22}$$

where $\phi(1) = e(k)$, $\phi(2) = e(k) - e(k-1)$, $\phi(3) = e(k) - 2e(k-1) + e(k-2)$. $\frac{\partial y_m}{\partial \Delta u}$ cannot be calculated directly, it could be replaced approximately by the function of relative change.

$$P = \frac{\partial y_m}{\partial \Delta u} = \operatorname{sgn}\left(\frac{y_m(k) - y_m(k-1)}{\Delta u(k) - \Delta u(k-1)}\right) \tag{23}$$

$$\frac{\partial G}{\partial W_{ij}} = \frac{\partial G}{\partial y_m}\cdot\frac{\partial y_m}{\partial \Delta u}\frac{\partial \Delta u}{\partial f_4}\frac{\partial f_4}{\partial W_{ij}} = -(y_d - y_m)\cdot P\cdot\phi\cdot f_3 \tag{24}$$

Similarly Equations (25) and (26) could be achieved.

$$\frac{\partial G}{\partial c_{ij}} = \frac{\partial G}{\partial y_m}\cdot\frac{\partial y_m}{\partial \Delta u}\cdot\frac{\partial \Delta u}{\partial f_4}\cdot\frac{\partial f_4}{\partial f_3}\cdot\frac{\partial f_3}{\partial f_2}\cdot\frac{\partial f_2}{\partial f_1} = -(y_d - y_m)\cdot P\cdot\phi\cdot f_3\cdot W_{ij}\frac{2(f_1(i) - c_{ij})}{b_{ij}^2} \tag{25}$$

$$\frac{\partial G}{\partial b_{ij}} = \frac{\partial G}{\partial y_m}\cdot\frac{\partial y_m}{\partial \Delta u}\cdot\frac{\partial \Delta u}{\partial f_4}\cdot\frac{\partial f_4}{\partial f_3}\cdot\frac{\partial f_3}{\partial f_2}\cdot\frac{\partial f_2}{\partial f_1} = -(y_d - y_m)\cdot P\cdot\phi\cdot f_3\cdot W_{ij}\frac{2(f_1(i) - c_{ij})}{b_{ij}^3} \tag{26}$$

W_{ij}, c_{ij} and b_{ij} can be adjusted by gradient optimization algorithm.

$$W_{ij}(k) = W_{ij}(k-1) - \eta\frac{\partial G}{\partial W_{ij}} + \beta(W_{ij}(k-1) - W_{ij}(k-2)) \tag{27}$$

$$c_{ij}(k) = c_{ij}(k-1) - \eta\frac{\partial G}{\partial c_{ij}} + \beta(c_{ij}(k-1) - c_{ij}(k-2)) \tag{28}$$

$$b_{ij}(k) = b_{ij}(k-1) - \eta\frac{\partial G}{\partial b_{ij}} + \beta(b_{ij}(k-1) - b_{ij}(k-2)) \tag{29}$$

where η is the learning rate; β is inertia factor.

4. Simulation Results and Analysis.

4.1. **Result of PSO optimizing FRBF offline.** PSO is adopted to optimize offline FRBF, the error function is defined as follows:

$$J = \sum |y_d - y_m| \tag{30}$$

where y_m is the actual output, y_d is the desired output.

For ball and plate's control, structure of FRBF adopts 2-14-49-3. A total of 168 parameters, which consist of the values of center c_{ij}, widths of membership function b_{ij} at Layer 2 of FRBF, and weights of RBF need to be optimized. Select 100 as the number of initial population based on experience, 168 as the individuals' length, 260 as the maximum number of iterations, $c_1 = c_2 = 2$, $\eta = 0.30$, $\beta = 0.05$. Change curve of error function for ball and plate system after PSO offline training FRBF is shown in Figure 4.

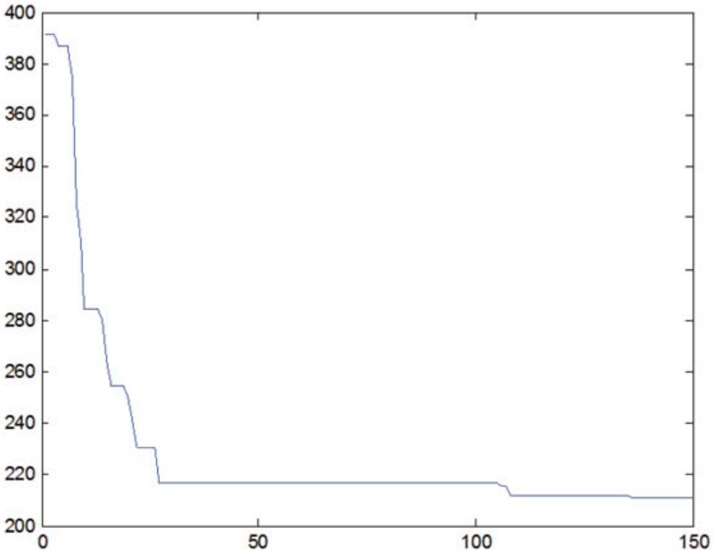


Figure 4. The result of the fitness function

After enough training, the best parameters of fuzzy neural network controller are achieved; the membership functions of fuzzy neural network after training are shown in Figures 5-8.

4.2. **Stabilizing control result.** When the given inputs of control system are step reference signal, and the system have disturbing signal (random number generated within [0, 0.1]), the simulation results are shown in Figure 9. Furthermore, the PSO optimizing FRBF-PID (PSO-FRBF-PID) simulation results are compared with conventional fuzzy PID (F-PID) control method; the contrast simulation results show that the proposed method has the merits with small overshooting, quick dynamic response and good control quality, etc.

Due to the symmetry of the system, here only the results of x-axis are given. Simulation results show that the method has a faster response time and overshoot is smaller than that before optimization fuzzy PID method. And this dotted line is smoother than the solid line, so control effect is better.

At the beginning of the next set-point simulation, target location coordinates are namely set as $(2, -2)$; $(4, -1)$; $(5, 1.5)$. The trajectory is shown in Figure 10. Tracking trajectory in X-axis and Y-axis are given in Figures 10(a) and 10(b). Then, the total tracking trajectory is shown in Figure 10(c). Simulation results show that PSO-FRBF-PID control system can more quickly reach the specified location and be quickly stabilized, and has small fluctuations.

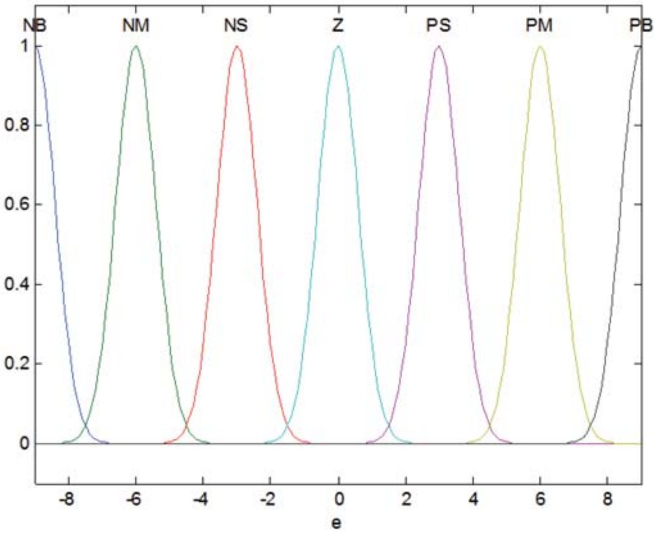


Figure 5. Membership functions of the input variable e

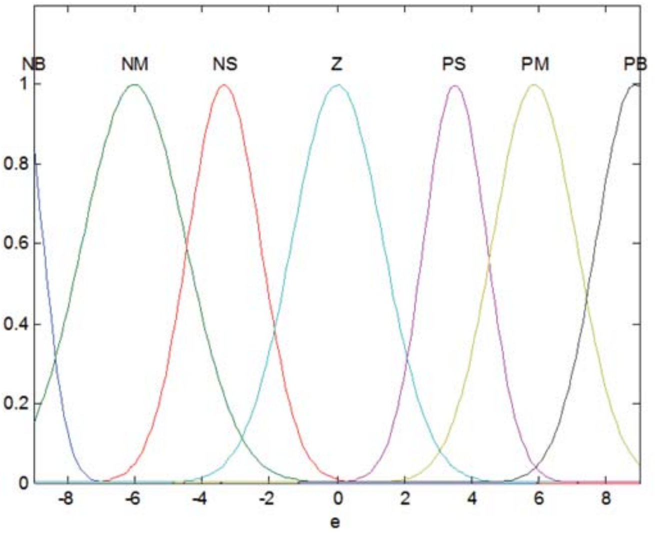


Figure 6. Membership functions of the input variable e after optimization

4.3. **Trajectory tracking control result.** For the set-point problem, it belongs to a regulator, and is easy at some extent. However, the tracking problem is especially challenging. This is because most of these systems are exhibit nonholonomic constraints. Because motion of the ball on rigid plate is subject to environment change, accordingly anterior standpoint, it belongs to a nonholonomic under-actuated dynamics system. When the ball is set a circle trajectory, its velocity need to change at all time. Therefore, this trajectory is the most difficult than others.

As shown in Figures 11 and 12, a circle with radius 100(mm) is chosen as the desired trajectory. Center of the circle is located in the plate center. Given tracking velocity is constant, and it is 12.56(mm/s). Before PSO optimization, trajectory tracking result is shown in Figure 11, which shows the tracking error is relatively large at the beginning, the average tracking error in X direction is 3.605(mm), average tracking error in Y direction is 6.306(mm), and average tracking error is 9.631(mm). With PSO-FRBF-PID controller, the tracking error reduces obviously as shown in Figure 12, the average tracking error in X direction is 3.105(mm), average tracking error in Y direction is 5.101(mm), and average tracking error is 5.533(mm). Compared Figure 12 with Figure 11, it is shown

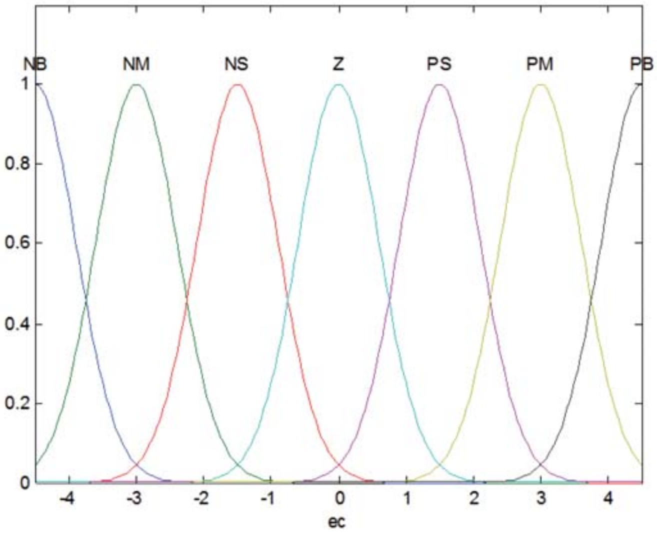


Figure 7. Membership functions of the input variable e_c

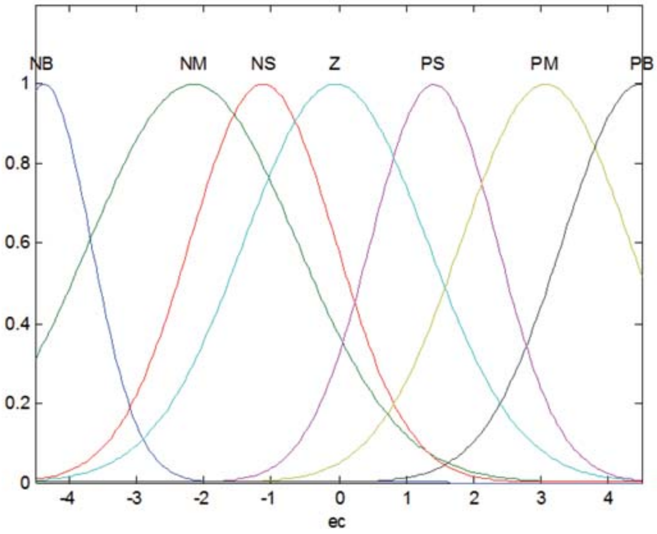


Figure 8. Membership functions of the input variable e_c after optimization

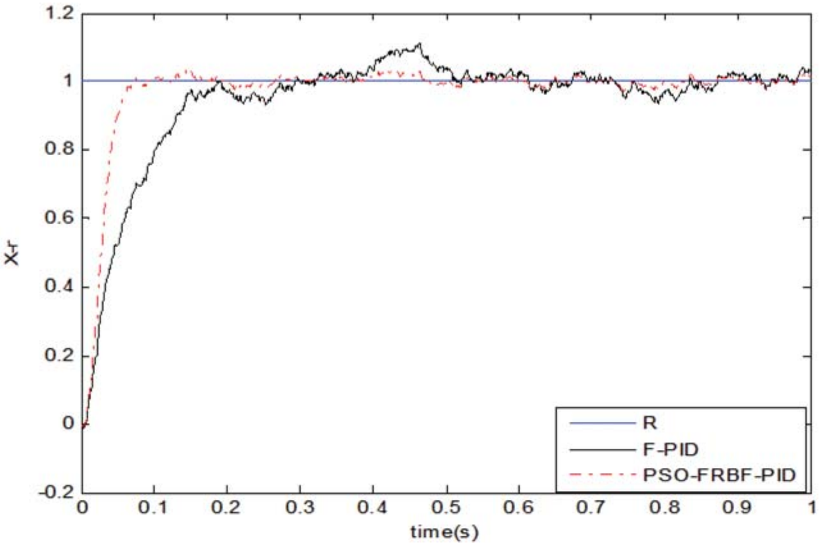
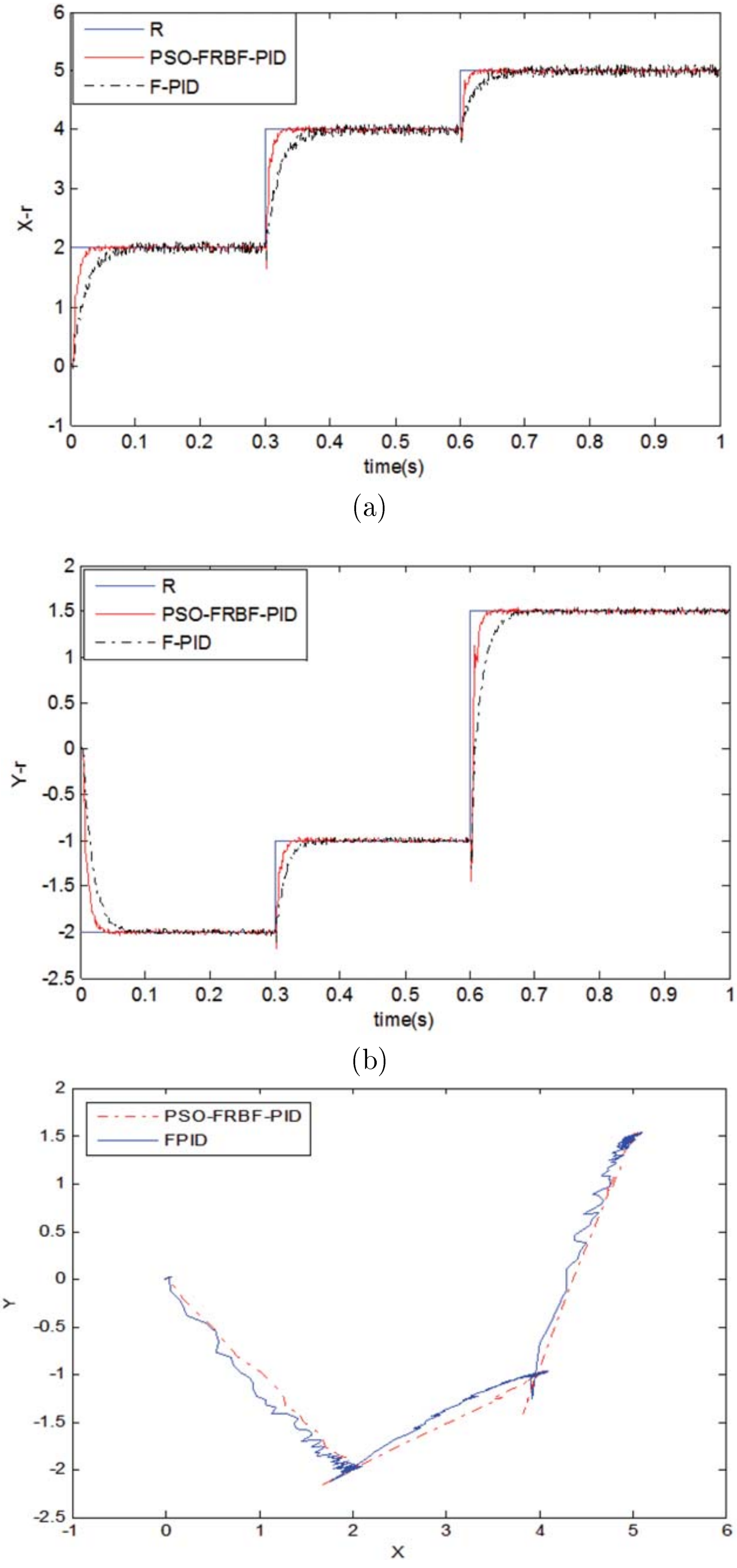


Figure 9. Response on a step reference signal



(a)

(b)

(c)

Figure 10. (a) X axis, (b) Y axis and (c) stabilizing simulation result after optimization tracking error and response velocity are all improved. Because of the presence of disturbance, robustness of this algorithm has proved.

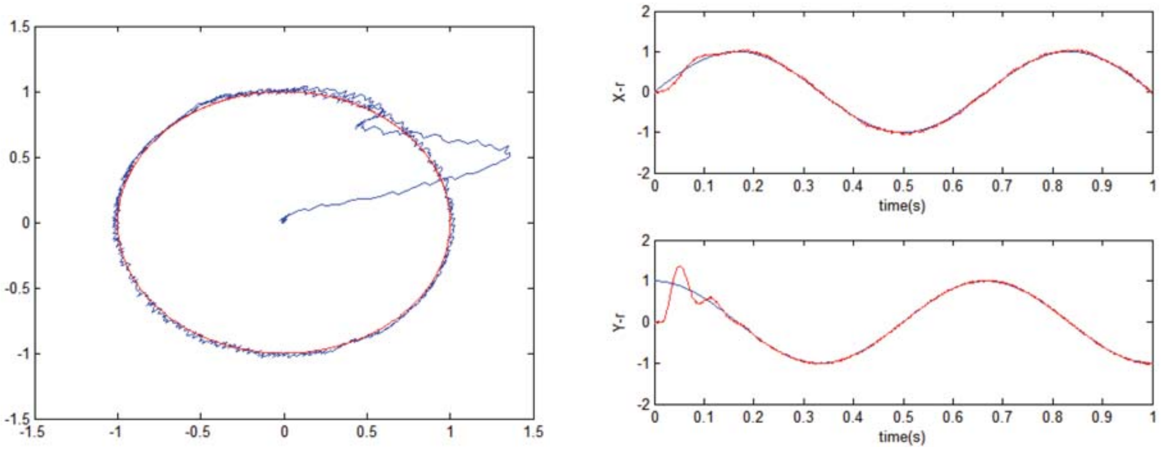


FIGURE 11. Trajectory tracking before optimization

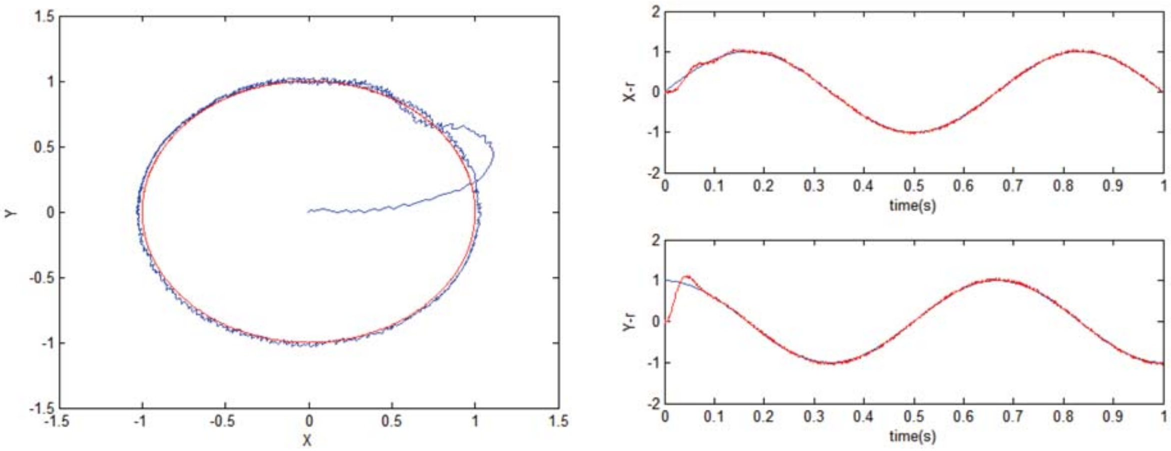


FIGURE 12. Trajectory tracking after optimization

5. **Conclusions.**

In this paper, for the control problem of ball and plate system, a FRBF tuning method optimized by PSO is proposed and tested. Design and simulation with MATLAB, show that multi-layer neural network is suitable to process large amounts of information fuzzy control in real time system. Applying technology of PSO in FRBF-PID system of ball and plate system is helpful to improve the dynamic properties and stability. Based on the fuzzy logic for overcoming the shortcomings in information processing and control, the PSO-FRBF-PID is more quickly, accurately and efficiently in control of the angle in ball and plate system. Given set-point control and trajectory tracking experiments have proved that the PSO-FRBF-PID controller does stabilize the ball and plate system. When given trajectory is a circle with radius 100(mm) and the ball velocity is 12.56(mm/s), tracking error in average is 5.533(mm). When the system work situation changes or disturbing signal is input, the control performance is better than conventional fuzzy PID controller. The parameters and rules optimized by PSO are excellent to adapt to varying system, which makes controlled performance optimal or near optimal.

Acknowledgment. The work was supported by the fund of Sichuan Provincial Key Laboratory of signal and information processing, Xihua University (No. SGXZD0101-10-1), partly by the Research Fund of Sichuan Provincial Key Discipline of Power Electronics and Electric Drive, Xihua University (No. SZD0503-09) and the Key Science and Research Fund of Sichuan Provincial Science and Technology Department (No. 06209084) and the Key Science and Research Fund of Sichuan Provincial Education Department (No. 06209005). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] J. Hauser, S. Sastry and P. Kokotovic, Nonlinear control via approximate input-output linearization: The ball and beam example, *IEEE Transactions on Automatic Control*, vol.37, no.3, pp.392-398, 1992.

[2] K. C. Ng and M. M. Trivedi, Neural integrated fuzzy controller and real-time implementation of a ball balancing beam, *Proc. of the 1996 IEEE Conf. on Robotics and Automation*, Minneapolis, MN, U.S.A., pp.1590-1595, 1996.

[3] S. Awtar, C. Bernard, N. Boklund, A. Master, D. Ueda and K. Craig, Mechatronic design of a ball-on-plate balancing system, *Mechatronics*, vol.12, no.2, pp.217-228, 2002.

[4] Humusoft Ltd, CE 151 ball and plate apparatus users manual, *User's Manual*, 1993.

[5] A. Rad, P. Chan and W. Lo, An online learning fuzzy controller, *IEEE Transactions on Industrial Electronic*, vol.50, no.5, pp.1016-1021, 2003.

[6] X. Fan, N. Zhang and S. Teng, Trajectory planning and tracking of ball and plate system using hierarchical fuzzy control scheme, *Fuzzy Sets and Systems*, vol.14, no.2, pp.297-312, 2003.

[7] J. Park and Y. Lee, Robust visual servoing for motion control of the ball on a plate, *Mechatronics*, vol.13, no.7, pp.723-738, 2003.

[8] N. Yubazaki, J. Yi, M. Otani, N. Unemura and K. Himota, Trajectory tracking control of unconstrained objects based on the SIRMs dynamically connected fuzzy inference model, *Proc. of IEEE International Conference on Fuzzy Systems*, vol.2, pp.609-614, 1997.

[9] Y. Tian, M. Bai and J. Su, A non-linear switching controller for ball and plate system, *Int. Journal of Modeling, Identification and Control*, vol.1, no.3, pp.177-182, 2006.

[10] K. J. Hunt, Extending the functional equivalence of radial basis function network and fuzzy inference system, *IEEE Trans. on Neural Networks*, vol.7, no.3, pp.776-781, 1996.

[11] M. Zhang and S. Hu, Fuzzy control of nonlinear systems with input saturation using multiple model structure, *International Journal of Innovative Computing, Information and Control*, vol.2, no.2, pp.131-136, 2008.

[12] X. Dong and Z. Zhang, Applying genetic algorithm to on-line updated PID neural network controllers for ball and plate system, *The 4th International Conference on Innovative Computing, Information and Control*, Kaohsiung, Taiwan, pp.751-755, 2009.

[13] C.-J. Lin, C. Chen and C. Lee, Classification and medical diagnosis using wavelet-based fuzzy neural networks, *International Journal of Innovative Computing, Information and Control*, vol.4, no.3, pp.735-748, 2008.

[14] W. Xue and Y. Guo, Fuzzy neural network control in main steam temperature system, *ICIC Express Letters*, vol.3, no.3(A), pp.409-414, 2009.

[15] X. Chen, C. Shen and C. Xu, Application of fuzzy neural network for fog zero point drift modeling, *ICIC Express Letters*, vol.3, no.3(B), pp.847-852, 2009.

[16] J. Li and Z. S. Sun, The study of simulation of fuzzy control in the ball and plate control system, *Electric Machines and Control*, vol.5, no.4, pp.270-273, 2001.

[17] G. Wang and Z. S. Sun, Algorithmic research on PD direct fuzzy control based on ball and plate apparatus, *Electric Drive*, vol.4, pp.23-25, 2004.

[18] S. Teng, N. Zhang and X. Fan, T-S model based fuzzy multi-variable control scheme for ball and plate system, *Information and Control*, vol.31, no.3, pp.268-271, 2002.

[19] F.-J. Lin and P.-H. Shen, Robust fuzzy neural network sliding-mode control for two-axis motion control system, *IEEE Transactions on Industrial Electronics*, vol.53, no.4, pp.1209-1225, 2006.

[20] F. Xue and Y. Tang, The design and simulation of neural fuzzy controller based on GA, *Journal of System Simulation*, vol.13, no.5, pp.573-575, 2001.