International Journal of Innovative Computing, Information and Control
Volume **11**, Number **5**, October **2015**

ICIC International ©2015 ISSN 1349-4198
pp. 1655–1666

AN IMPROVED NEURAL NETWORK ADAPTIVE SLIDING MODE CONTROL USED IN ROBOT TRAJECTORY TRACKING CONTROL

Tianhua Liu and Shoulin Yin

Software College
Shenyang Normal University
No. 253, North Huanghe Street, Huanggu District, Shenyang 110034, P. R. China
liutianhua@sina.com; 352720214@qq.com

Received January 2015; revised May 2015

Abstract. *This paper proposes an improved neural network adaptive sliding mode control method based on the neural network sliding mode control to improve the performance of the robot trajectory tracking control. This new scheme regards neural network as a controller and adopts robust control law to eliminate the approximation error which uses its nonlinear mapping ability to approximate various unknown nonlinear systems. Hidden layer units and network structure parameter have an effect on neural network mapping. So we will decrease chattering. We conduct experiments with three joints robot and give an example to verify this paper's scheme under the MATLAB platform. The results of experiments show that the new neural network adaptive sliding mode control has a good control accuracy and robustness than other control methods. The new scheme reduces the chattering effectively and also decreases the effect of hidden layer units and network structure parameter on neural network mapping. It will be a good choice for the robot trajectory tracking control.*

Keywords: Robot tracking, Neural network, Neural network adaptive sliding mode control, Particle swarm optimization algorithm, Chattering

1. **Introduction.** As we all know, robot system [1] is a programmable controller. It can do some specific function with drive circuit. As we all know, appropriate control strategies [2-4] have been used for getting rapid and precise tracking control.

Sliding mode can be designed as required. The sliding mode of system and parameter variation of controlled member have nothing to do with the outside interference of system. Therefore, the robustness of sliding mode variable structure control system is stronger than the conventional continuous system. Discontinuous switching characteristics of sliding mode variable structure control will cause system chattering in essence. Especially for the discrete sliding mode variable structure control system, it will add a zigzag path on the smooth sliding mode. In fact, there must exist chattering. And it eliminates the chattering and also eliminates the perturbation resistance of variable structure control and the ability of resisting disturbance. It can lose effectiveness [5,6] of system. So we can only weaken chattering. To solve chattering problem in sliding mode control, Shi et al. [7] proposed a more realistic and accurate measurement mode to compensate for the negative influence of both missing data and different time delays in a random way. Nazari et al. [8] proposed a parallel fuzzy logic theory used to compensate the system dynamic uncertainty. At the same time, it combined fuzzy logic methodology and sliding mode. Fuzzy logic methodology could compensate the control error and sliding mode could reduce and remain the fuzzy inference system's error. Finally, it made the tracking error of robot asymptotically stable. Nguyen et al. [9] developed an adaptive sliding mode controller for tracking control. An adaptive sliding mode controller adopted a fuzzy

neural network to estimate the unknown nonlinear models for constructing the sliding mode controller and a compensational controller adaptively compensated estimation errors. At the end, it used Lyapunov synthesis approach to ensure the stability of controlled systems. Lian [10] represented an adaptive self-organizing fuzzy sliding-mode radial basis function neural-network controller which used a radial basis function neural-network to regulate the parameters of the self-organizing fuzzy controller. This method solved the problem in determining the stability of the robot system control. Wu et al. [11] presented a robust adaptive sliding-mode control scheme for a class of condenser-cleaning mobile manipulator in the presence of parametric uncertainties and external disturbances.

This paper proposes an improved neural network adaptive sliding mode control method based on neural network sliding mode control which in robot systems considers radial basis function neural network as a controller and utilizes its nonlinear mapping ability to approximate various uncertain items. Robust control law is used to remove the approximation error. Moreover, it compares and analyzes the performance of several neural network structure for the effect of hidden layer units number and network structure parameter on the effectiveness of neural network mapping. Once determining the neural network's structure model, it uses particle swarm optimization to solve center position and base width parameter of neural network which is difficult to be calculated. We apply the new method into three-joint robot trajectory tracking control in this paper. Finally, under the MATLAB simulation environment, we compare wide range disturbance with general disturbance and achieve the different control effects with different sliding mode control methods. This paper's structure is as follows. Section 2 introduces some necessary models. Improved RBF neural network adaptive sliding model control is described in detail in Section 3. Numerical examples and some comparison results are given in Section 4. The conclusions are drawn in Section 5.

2. **Problem Statement.** We must take friction, non-modeling state and external disturbance into consideration for n-joint robot. Using Lagrange method can obtain robot [12] kinetic equation.

$$M(c)c_2 + C(c_1, c_2) + G(c) + F(c_1) + \upsilon_d = \upsilon \tag{1}$$

where c_1 and c_2 are velocity vector and acceleration vector respectively, and $c \in R^n$ is joint angle displacement. $M(c)$ is the inertial matrix of robot. $C(c_1, c_2)$ is centrifugal force and Coriolis matrix. $G(c)$ is the gravity vector acting on the joints. $F(c_1)$ is the vector consisting of frictional force. υ_d is the vector which composes of model error, parameter variation and external disturbance uncertain factors. υ is composed of control moment.

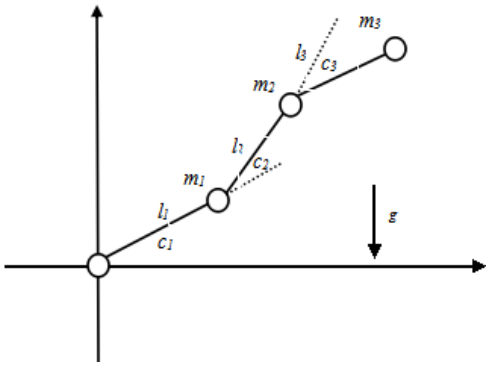

FIGURE 1. Three-joint robot structure

This paper selects a three-joint robot to make research. Its joint structure is as Figure 1.

In Figure 1,

$h_{11} = (m_1 + m_2)l_1^2 + m_2 l_2^2 + 2m_2 l_1 l_2 \cos c_2' + J_1$. $h_{12} = h_{21} = m_2 l_2^2 + m_2 l_1 l_2 \cos c_2'$.

$h_{22} = (m_2 + m_3)l_2^2 + m_3 l_3^2 + 2m_3 l_2 l_3 \cos c_3' + J_2$. $h_{23} = h_{32} = m_3 l_3^2 + m_3 l_2 l_3 \cos c_3'$.

$h_{33} = m_3 l_3^2 + J_3$. $C(c) = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix}$. $G(c) = \begin{bmatrix} G_1 \\ G_2 \\ G_3 \end{bmatrix}$. $M(c) = \begin{bmatrix} 1\,0\,0 \\ 0\,0\,0 \\ 0\,0\,0 \end{bmatrix}$.

$F(c_1) = \begin{pmatrix} F_1 \\ F_2 \\ F_3 \end{pmatrix} = \begin{pmatrix} 10c_1' + 3\mathrm{sgn}(c_1') \\ 10c_2' + 3\mathrm{sgn}(c_2') \\ 10c_3' + 3\mathrm{sgn}(c_3') \end{pmatrix}$. $c_{11} = -m_2 l_1 l_2 \sin c_2 c_2'$.

$c_{12} = -m_2 l_1 l_2 \sin c_2 (c_2' + c_1')$. $c_{21} = m_2 l_1 l_2 \sin c_2 c_1'$. $c_{22} = -m_3 l_2 l_3 \sin c_3 c_3'$.

$c_{23} = -m_3 l_2 l_3 \sin c_3 (c_3' + c_2')$. $c_{32} = m_3 l_2 l_3 \sin c_3 c_2'$. $c_{33} = 0$.

$G_1 = (m_1 + m_2) g l_1 \cos c_1 + m_2 g l_2 \cos(c_1 + c_2)$.

$G_2 = (m_2 + m_3) g l_2 \cos c_2 + m_3 g l_3 \cos(c_2 + c_3)$. $G_3 = m_3 g l_3 \cos(c_2 + c_3)$.

where c_1 is angular displacement of rod1. c_2 is angular displacement of rod2. c_3 is angular displacement of rod3. m_1, m_2 and m_3 are the mass of rod1, rod2 and rod3 respectively. They are expressed by the point mass of connecting rod end. l_1, l_2 and l_3 are the length of rod1, rod2 and rod3 respectively. g is gravity acceleration. J_1, J_2 and J_3 are the rotational inertia of m_1, m_2 and m_3 respectively. h_{11} represents the total moment of inertia with taking original as center. h_{12} denotes the rotational inertia of m_1 with taking m_2 as center. h_{21} denotes the rotational inertia of m_2 with taking m_1 as center. h_{23} denotes the rotational inertia of m_2 with taking m_3 as center. h_{22} and h_{33} represent the total moment of inertia with taking original as center. c_1', c_2' and c_3' is the velocity vector of rod1, rod2 and rod3 respectively. G_1, G_2, G_3 is the gravity vector of rod1, rod2 and rod3 acting on the joints respectively.

3. **The Proposed Improved RBF Neural Network.**

3.1. **Sliding mode variable structure control for the new method.** In the practical engineering, designing of sliding mode variable structure control is to let the true trajectory c trace the desired trajectory c_d preferably. So we define tracing error as:

$$\varepsilon = c_d - c,\ \dot{\varepsilon} = c_d' - c'. \tag{2}$$

The sliding mode function r is defined as:

$$r = \dot{\varepsilon} + \Im\varepsilon \tag{3}$$

where $c_d = (c_{d1}\ \ c_{d2}\ \ c_{d3})^T$, $\Im = \Im^T > 0$, $\Im = diag(\lambda_1, \lambda_2, \lambda_3)$ is positive diagonal matrix. $\lambda_1, \lambda_2, \lambda_3 > 0$, thus we obtain:

$$\begin{aligned} M\dot{r} &= M(c_d'' - c'' + \Im\dot{\varepsilon}) = M(c_d'' + \Im\dot{\varepsilon}) - Mc'' \\ M\dot{r} &= M(c_d'' + \Im\dot{\varepsilon}) + Cc' + G + F + \upsilon_d - \upsilon \\ M\dot{r} &= M(c_d'' + \Im\dot{\varepsilon}) - Cr + C(c_d' + \Im\varepsilon) + G + F + \upsilon_d - \upsilon \\ M\dot{r} &= -Cr - \upsilon + f + \upsilon_d \end{aligned} \tag{4}$$

where $c' = c_d' - \dot{\varepsilon} = c_d' - r + \Im\varepsilon$, $f = M(c_d' + \Im\varepsilon) + C(c_d + \Im\varepsilon) + G + F$.

Lyapunov stability requirements are very important for control system stability conditions. According to (3), we can develop variable structure controller with standard sliding mode.

3.2. **The new RBF neural network adaptive sliding mode controller.** RBF neural network adaptive sliding mode is a three-layer forward network. The mapping of input or output is nonlinear. However, it is linear from hidden layer space to output space which accelerates learning speed and avoids local minimum. RBF neural network structure is simple. It has a simple training and fast convergence. It can approximate any nonlinear function [13]. Function approximation can approximate any continuous function by any arbitrary precision function. General functions can be expressed as a set of linear combinations of base functions. RBF network is equivalent to using the output of the hidden layer units to constitute a group of base functions. Then it uses the output layer to conduct linear combination and completes the approximation work. The uncertainty item f can be approximated by RBF neural network. Network input is: $x = \begin{pmatrix} \varepsilon^T & \dot{\varepsilon}^T & c_d^T & c_d'^T & c_d''^T \end{pmatrix}$.

Gaussian function is the basic function in this paper. Network method based on RBF neural network can be set as:

$$a_{ij} = e^{\left(-||x_i - c_{ij}||^2 / b_{ij}^2\right)} \tag{5}$$

$$f_i = w_i^T h_i + \tau_i \tag{6}$$

where $i = 1, 2, \ldots, n$, n is the number of joint. $x_i = \begin{pmatrix} \varepsilon_i^T & \dot{\varepsilon}_i^T & c_{di}^T & c_{di}'^T & c_{di}''^T \end{pmatrix}$ is the neural network input of i-th joint. $j = 1, 2, \ldots, m$, m is the number of hidden layer neurons. $a_i = (a_{i1}\ a_{i2}\ \ldots\ a_{im})$ is radial basis vector. c_i is central position of Gaussian function. b_i is base width parameter. τ_i is neural network approximation error of i-th joint. w_i is the inertia weight of i-th joint. f_i is the network output error of i-th joint.

Approximated network output $\hat{f}$ can be expressed as follows which adopts RBF neural network to approximate f based on the expression of f_i:

$$\begin{aligned} \hat{f}(x) &= \begin{pmatrix} \hat{f}_1 & \hat{f}_2 & \ldots & \hat{f}_n \end{pmatrix}^T \\ \hat{f}(x) &= \begin{bmatrix} w_1^T a_1 + \tau_1 & w_2^T a_2 + \tau_2 & \ldots & w_n^T a_n + \tau_n \end{bmatrix}^T \\ \hat{f}(x) &= \begin{bmatrix} w_1^T a_1 & w_2^T a_2 & \ldots & w_n^T a_n \end{bmatrix}^T + \tau \\ \hat{f}(x) &= \hat{W}^T h(x) + \tau \end{aligned} \tag{7}$$

where $\tau = (\tau_1 \quad \tau_2 \quad \ldots \quad \tau_n)^T$. Set $\tilde{W} = W - \hat{W}$.

Then the control law σ is:

$$\sigma = \hat{W}^T h(x) + K_v r - v. \tag{8}$$

The adaptive law of neural network is:

$$\dot{\hat{W}} = F h r^T. \tag{9}$$

The aim of setting adaptive law is to adjust system and make the state of RBF network match the output of network model which can enhance the stability of system. Here $\hat{W}$ is adaptive law, and v is robust item for overcoming neural network approximation error. The aim of setting v is to enhance the robustness of the control. Putting (8) into (4) can get the following equation:

$$M\dot{r} = -(K_v + C) + \tilde{W}^T a(x) + (\tau + \sigma_d) + v. \tag{10}$$

Theorem 3.1. *Modeling error σ_d and function approximation error must be taken into consideration for the neural network when proving the stability of RBF neural network adaptive sliding mode control. And $||\tau|| \le \tau_N$, $||\sigma_d|| \le b_d$. Let v be $v = -(\tau_N + b_d)\mathrm{sgn}(r)$. We define the new Lyapunov function $\dot{L} \le 0$.*

Proof: Redefining the Lyapunov function:

$$L = \frac{1}{2} r^T M r + \frac{1}{2} tr\left(\tilde{W}^T F^{-1} \tilde{W}\right). \tag{11}$$

Lyapunov stability theory has an important effect on random perturbation nonlinear systems. Using control law expression (8) and adaptive law expression (9) can get:

$$\dot{L} = r^T M \dot{r} + \frac{1}{2} r^T \dot{M} r + tr\left(\tilde{W}^T F^{-1} \dot{\tilde{W}}\right). \tag{12}$$

Formula (10) is plugged into Formula (12).

$$\dot{L} = -r^T K_v \dot{r} + r^T \left(\varepsilon + \tau_d + v\right). \tag{13}$$

Since

$$\begin{aligned} r^T(\tau + \sigma_d + v) &= r^T(\tau + \sigma_d) + r^T v \\ &= r^T(\tau + \sigma_d) - ||r||(\tau_N + b_d) \leq 0 \end{aligned} \tag{14}$$

it can get $\dot{L} \leq 0$.

Remark 3.1. *Based on the Lyapunov stability discriminant theorem, if $r \neq 0$, then $\dot{L}(r) < 0$. If $r = 0$, then $\dot{L}(r) = 0$. $\dot{L}(r)$ is negative definite. It can conclude that r and e are uniform boundedness based on Lyapunov stability theorem and $\dot{L} \leq 0$. $\dot{\varepsilon}$ is uniform boundedness from $r = \dot{\varepsilon} + \Lambda\varepsilon$. According to Babalat theorem, if $t \to \infty$, then $\dot{L} = 0$ (i.e., $t \to 0$). So it gets $\varepsilon \to 0$ and $\dot{\varepsilon} \to 0$. Also the stability of the system has been demonstrated.*

By the above theorem, we are ready to present the numerical results on neural network adaptive sliding mode control of robot trajectory tracking control.

4. Numerical Example with Robot Tracking Systems.

4.1. Practical systems analysis.

This paper proposes improved neural network adaptive sliding mode control which can be used in real systems applications, such as automatic operating systems, robot operating systems, communication systems, neural network control systems and production systems. Also they can be built as hybrid systems. In this paper, we take robot tracking system as an example. The numerical example reflects the proposed new neural network adaptive sliding mode control with the real systems. Our aim is to demonstrate and illustrate the significance and novelty of the work.

Considering space three-joint robot's physical parameters are set as: $m_1 = 2\text{kg}$, $m_2 = 3\text{kg}$, $m_3 = 4\text{kg}$, $l_1 = 1.22\text{m}$, $l_2 = 0.78\text{m}$, $l_3 = 0.64\text{m}$, $g = 9.8\text{m/s}^2$, $J_1 = J_2 = J_3 = 4\text{kg·m}$. The desired tracking trajectory is $c_{d1} = c_{d2} = c_{d3} = 0.2\sin t$. $c(0) = (0 \;\; 0 \;\; 0)^T$, $c'(0) = (0 \;\; 0 \;\; 0)^T$ are initial conditions. The parameters of sliding mode control are set $K_v = diag(50, 50, 50)$, $\Im = diag(5, 5, 5)$. We select $\varepsilon_N = 0.3$ and $b_d = 0.1$. The function of uncertain factors can be denoted by $\sigma_d = (\sigma_{d1}, \sigma_{d2}, \sigma_{d3})$, $\sigma_{d1} = \alpha_1 \sin(\beta_1 \pi t)$, $\sigma_{d2} = \alpha_2 \sin(\beta_2 \pi t)$, $\sigma_{d3} = \alpha_3 \sin(\beta_3 \pi t)$. α_i and β_i can confirm disturbance. The disturbance can increase with the increasing of α_i and β_i. The different three-joint disturbances have an effect on the control effectiveness. So we set general disturbance and big disturbance in experiment. The general disturbance parameters can be set $\alpha_1 = 0.8$, $\alpha_2 = \alpha_3 = 4.5$, $\beta_i = 1$. The big disturbance parameters are $\alpha_1 = 37$, $\alpha_2 = \alpha_3 = 118$, $\beta_i = 1$.

We make three-joint robot trajectory tracking experiments by the MATLAB platform. Table 1 shows that the number of hidden layer neurons has an effect on the performance

of RBF neural network. It takes the average of correlation values R after system program running 9 times as the evaluation index. The correlation value R can be described as:

$$R = \frac{(o - o')(\hat{o} - \hat{o}')^T}{\sqrt{(o - o')(o - o')^T}\sqrt{(\hat{o} - \hat{o}')(\hat{o} - \hat{o}')^T}} \tag{15}$$

where o' and $\hat{o}'$ represent the average value of actual output and RBF neural network output respectively. The correlation value R ranges from $[0, 1]$. The more correlation value R is close to 1, the more network model has a better performance. Otherwise, the performance is worse.

Table 1 shows that when the number of hidden layer neurons is 9, the average of correlation values R is maximum. And it uses particle swarm optimization algorithm to optimize the central position c and base width parameter b. The optimized central position c and base width parameter b are close to the optimal value.

And we conduct experiments between the displacement of new scheme and the real value for robot tracking system. Figure 2 is the result. The left figure shows that the robot tracking value with the proposed RBF neural network adaptive sliding mode control is close to the real value within 20s. The right figure represents that with the time increasing, the rate of convergence with experiment value achieves the optimal effect. When $10\text{s} \leq t \leq 14\text{s}$ and $25\text{s} \leq t \leq 37\text{s}$, the robot re-sets moving direction.

4.2. **Robot systems comparison with Joint-1, Joint-2 and Joint-3.** We use fuzzy sliding mode control method, RBF neural network sliding mode control method and the

TABLE 1. The average of correlation values R with running 9 times

The number of hidden layer neurons	*Correlation values R*		
	The minimum value	*The maximum value*	*The average value*
4	*0.3123*	*0.4112*	*0.3843*
5	*0.3672*	*0.4252*	*0.4062*
6	*0.3872*	*0.5261*	*0.4614*
7	*0.7147*	*0.7752*	*0.7498*
8	*0.5723*	*0.8292*	*0.6837*
9	*0.8262*	*0.8883*	*0.8523*
10	*0.7605*	*0.8461*	*0.8233*
11	*0.7317*	*0.7773*	*0.7545*
12	*0.7115*	*0.7423*	*0.7134*

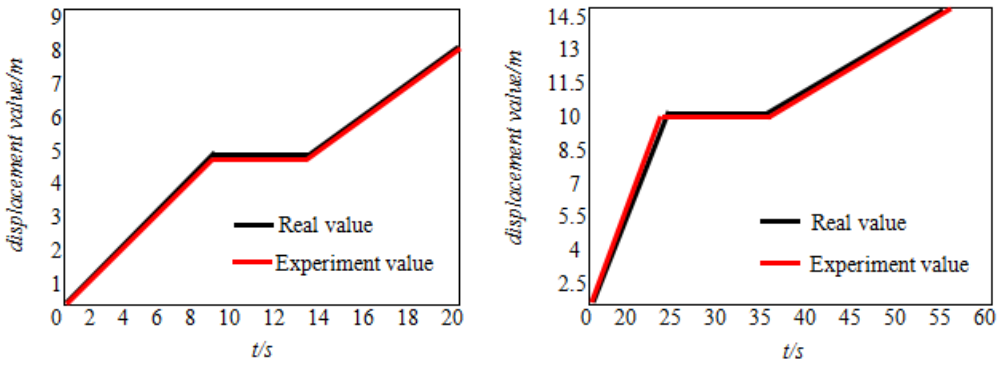

FIGURE 2. Robot tracking value with different time

improved RBF neural network adaptive sliding mode control to compare in this paper. It analyzes the three methods' effect on the performance of robot trajectory tracking. Figures 3-8 are the position tracking and control input of fuzzy sliding mode control method. It shows that though it can track the robot position with fuzzy sliding mode control, the error of system is larger. Chattering interference is very obvious. Figures 9-14 are the simulation results of RBF neural network sliding mode control method. The control input of RBF neural network sliding mode control is superior to fuzzy sliding mode. Each Joint position tracking can reach the desired position. Figures 15-20 are the simulation results of improved RBF neural network adaptive sliding mode control method with the optimized network structure and network parameters. The tracking effect of improved RBF is the best. It also reduces the chattering effectively which can reach the ideal tracking effect. Figures 21-26 are the simulation results contrast figures

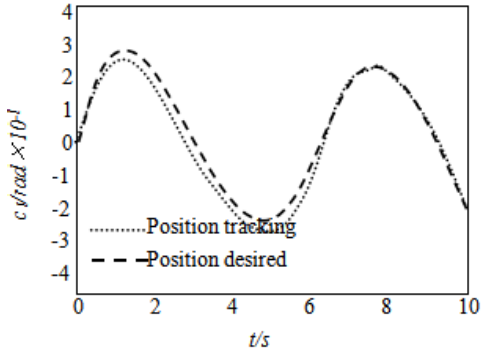

FIGURE 3. Joint-1 position tracking

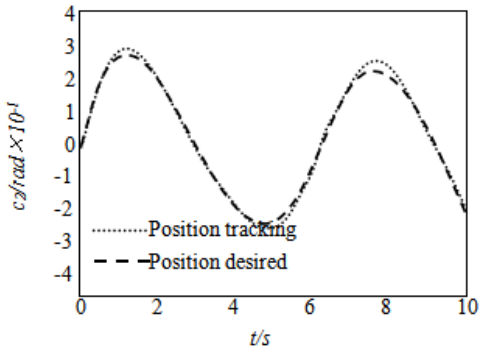

FIGURE 4. Joint-2 position tracking

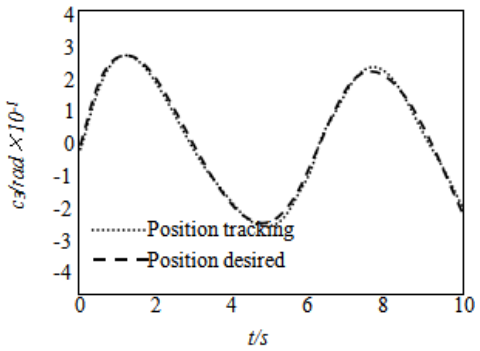

FIGURE 5. Joint-3 position tracking

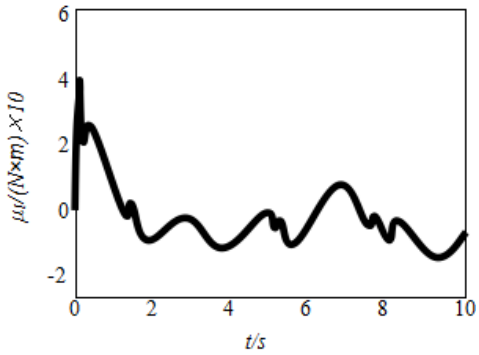

FIGURE 6. Joint-1 control input

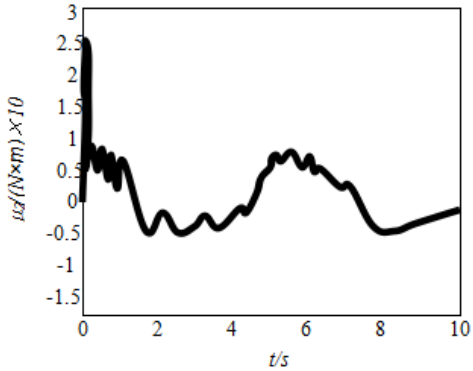

FIGURE 7. Joint-2 control input

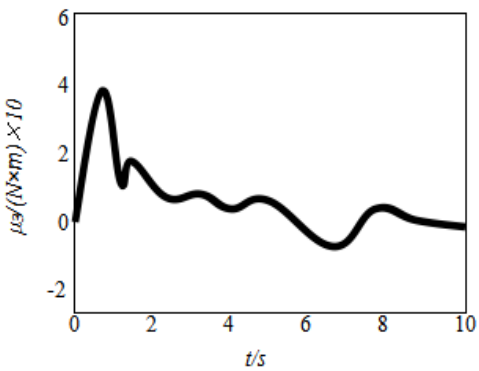

FIGURE 8. Joint-3 control input

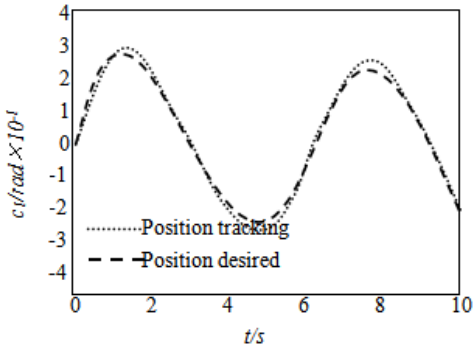

FIGURE 9. Joint-1 position tracking

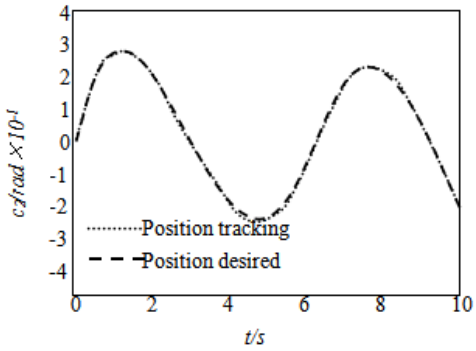

FIGURE 10. Joint-2 position tracking

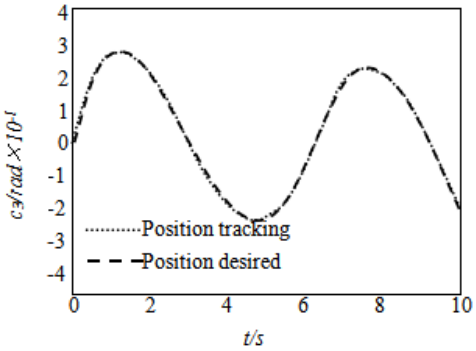

FIGURE 11. Joint-3 position tracking

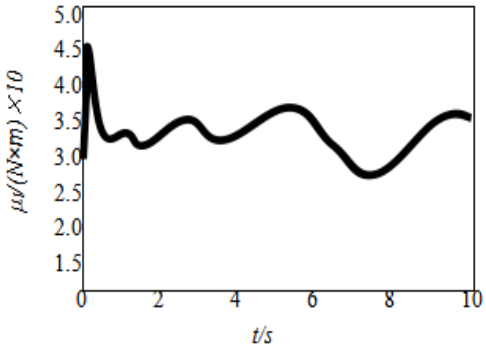

FIGURE 12. Joint-1 control input

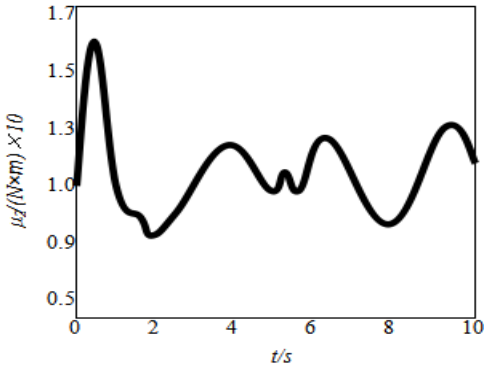

FIGURE 13. Joint-2 control input

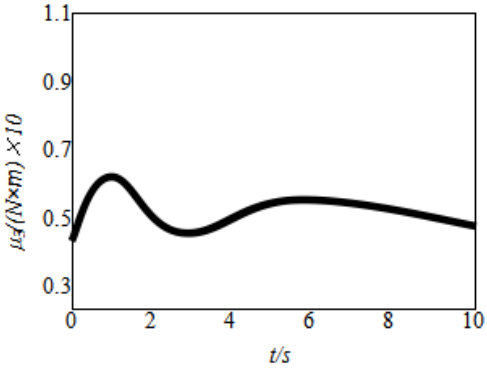

FIGURE 14. Joint-3 control input

with different methods under large disturbance. Figures 27-29 are the control input simulation results contrast figures with different methods under large disturbance. They all represent that the improved method has the smallest error and strong robustness.

4.3. **A comparison with reference [14].** Reference [14] designs a state estimator to estimate network states by available output measurements and stochastic Lyapunov-Krasovskii functional approach. Under uncertain discrete-time neural networks conditions, it provides robust stochastic finite-time state estimation sufficient conditions with time-varying delays and Markovian jumps. We use the example-1 to make comparison between reference [14] and this paper under the same MATLAB environment. The physical parameters are set to the same. $[A_1, A_2]^T$, $[B_1, B_2]^T$, $[C_1, C_2]^T$, $[U_1, U_2]^T$, $[H_1, H_2]^T$ and $[V_1, V_2]^T$ are as example-1. Correlation values $R = 0.7$. The optimal bound ε of robot trajectory tracking is as the following Figure 30 through experiments.

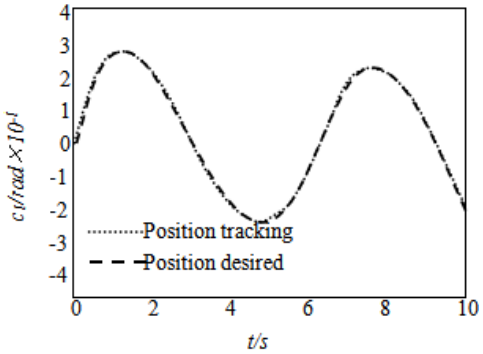

FIGURE 15. Joint-1 position tracking comparison

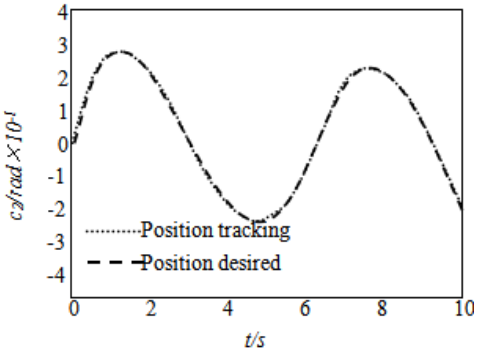

FIGURE 16. Joint-2 position tracking comparison

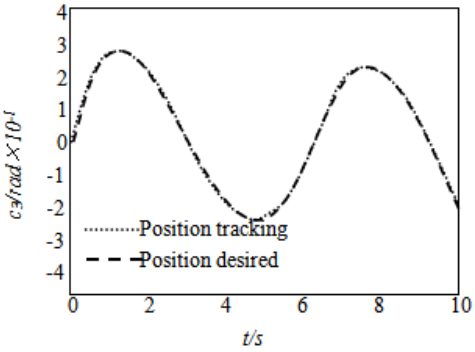

FIGURE 17. Joint-3 position tracking comparison

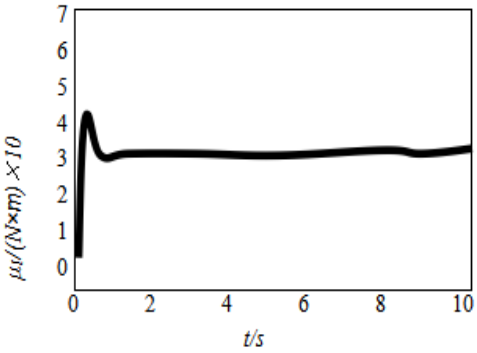

FIGURE 18. Joint-1 control input

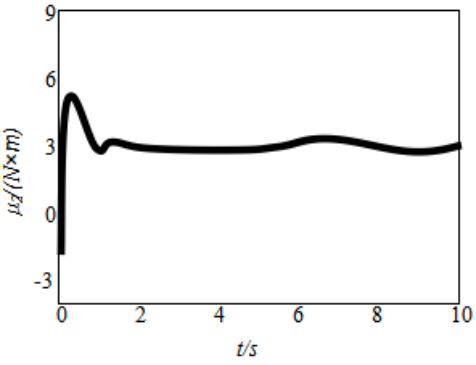

FIGURE 19. Joint-2 control input

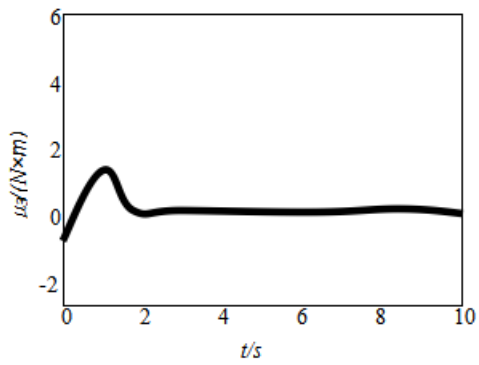

FIGURE 20. Joint-3 control input

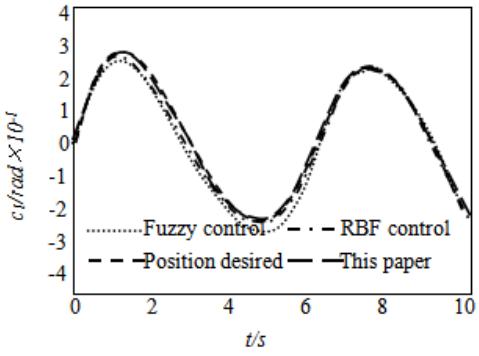

FIGURE 21. Joint-1 position tracking comparison

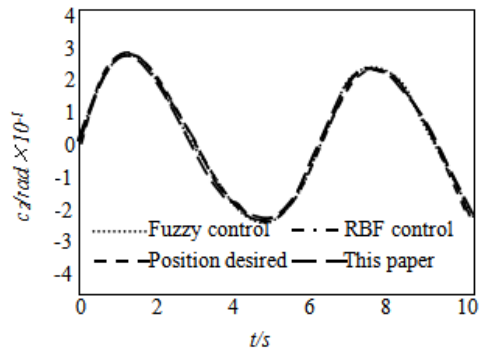

FIGURE 22. Joint-2 position tracking comparison

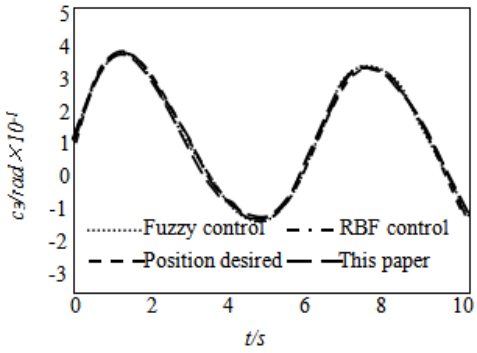

FIGURE 23. Joint-3 position tracking comparison

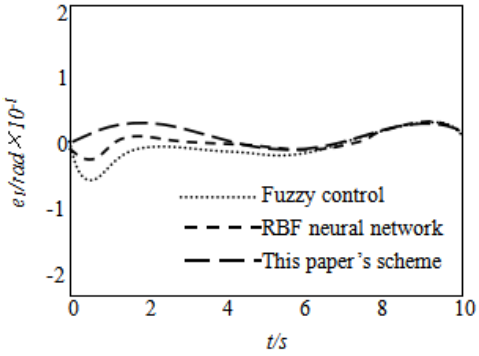

FIGURE 24. Joint-1 position tracking error comparison

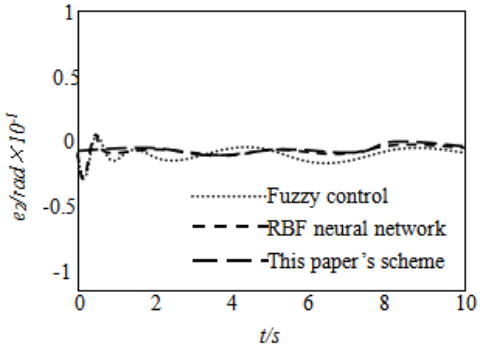

FIGURE 25. Joint-2 position tracking error comparison

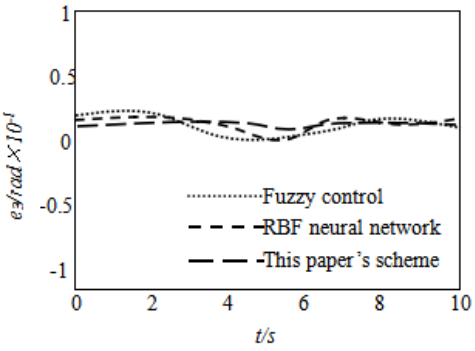

FIGURE 26. Joint-3 position tracking error comparison

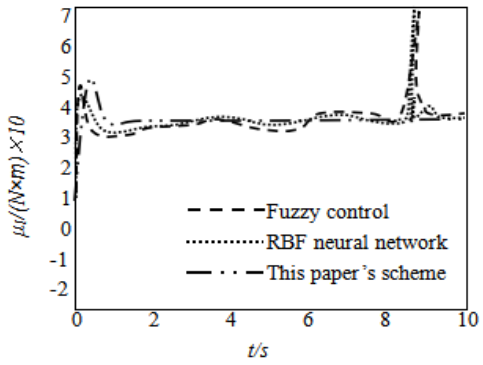

FIGURE 27. Joint-1 control input comparison

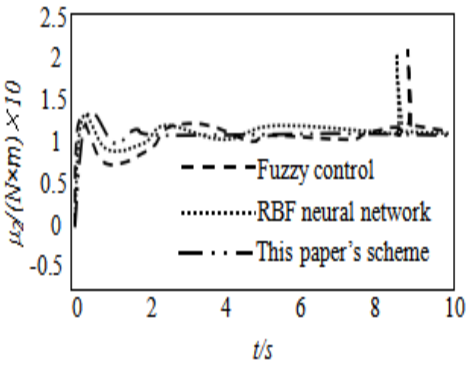

FIGURE 28. Joint-2 control input comparison

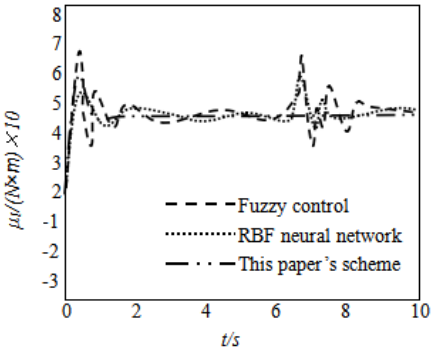

FIGURE 29. Joint-3 control input comparison

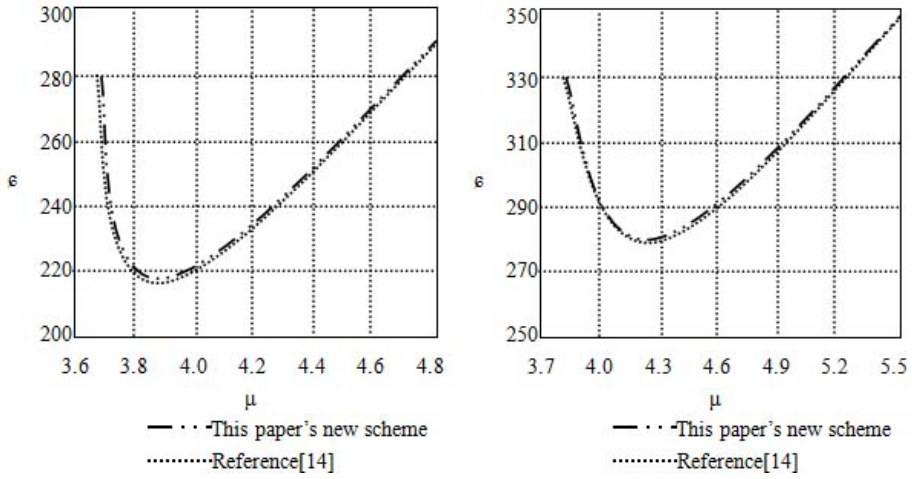

FIGURE 30. Local optimal bound of ε in robot tracking system

In Figure 30, we set parameter $\mu = 3.9$ and parameter $\mu = 4.2$ respectively. In the left figure, $\varepsilon_1 = 218 > \varepsilon_2 = 217$ when $\mu = 3.9$. In the right figure, $\varepsilon_1 = 280 > \varepsilon_2 = 279$ when $\mu = 4.2$ (ε_1 is the optimal bound of this paper. ε_2 is the optimal bound of reference [14]). And with the increase of μ, their convergence curves are close to equal. So we can conclude that ε of the improved scheme is superior to reference [14].

5. **Conclusions.** In order to reduce the effect of system model error and some interference on robot trajectory tracking control, we propose an improved neural network sliding mode control method. The number of hidden layer units and network structure parameters play an important role in the tracking effectiveness. Particle swarm optimization algorithm is adopted to improve network structure's parameters. Thus, we can get optimal results of chattering. It can also increase the error convergence speed. Finally, we conduct experiments with Joint-3 robot. It demonstrates that the new scheme has a good robustness and accuracy control and it also weakens the chattering effectively in sliding mode control. In the future, we will do more works to improve neural network adaptive sliding mode control.

REFERENCES

[1] C. Plociennik, H. W. Schöck and M. Moors, *Robot Interaction System: U.S. Patent 8,700,197[P]*, 2014.

[2] R. S. Muñoz-Aguilar, P. Rodríguez, A. Dòria-Cerezo et al., A sensor-less sliding mode control scheme for a stand-alone wound rotor synchronous generator under unbalanced load conditions, *International Journal of Electrical Power & Energy Systems*, vol.60, pp.275-282, 2014.

[3] P. Shi, Y. Yin, F. Liu et al., Robust control on saturated Markov jump systems with missing information, *Information Sciences*, vol.265, pp.123-138, 2014.

[4] P. Shi, Y. Xia, G. P. Liu et al., On designing of sliding-mode control for stochastic jump systems, *IEEE Transactions on Automatic Control*, vol.51, no.1, pp.97-103, 2006.

[5] M. A. Hussain and P. Y. Ho, Adaptive sliding mode control with neural network based hybrid models, *Journal of Process Control*, vol.14, no.2, pp.157-176, 2004.

[6] C. H. Tsai, H. Y. Chung and F. M. Yu, Neuro-sliding mode control with its applications to seesaw systems, *IEEE Transactions on Neural Networks*, vol.15, no.1, pp.124-134, 2004.

[7] P. Shi, X. Luan and C. L. Liu, Filtering for discrete-time systems with stochastic incomplete measurement and mixed delays, *IEEE Transactions on Industrial Electronics*, vol.59, no.6, pp.2732-2739, 2012.

[8] I. Nazari, A. Hosainpour, F. Piltan et al., Design sliding mode controller with parallel fuzzy inference system compensator to control of robot manipulator, *I. J. Intelligent Systems and Applications*, pp.63-75, 2014.

[9] T. B. T. Nguyen, T. L. Liao and J. J. Yan, Adaptive sliding mode control of chaos in permanent magnet synchronous motor via fuzzy neural networks, *Mathematical Problems in Engineering*, vol.2014, 2014.

[10] R. J. Lian, Adaptive self-organizing fuzzy sliding-mode radial basis-function neural-network controller for robotic systems, *IEEE Transactions on Industrial Electronics*, vol.61, no.3, pp.1493-1503, 2014.

[11] X. R. Wu, Y. N. Wang and X. J. Dang, Robust adaptive sliding-mode control of condenser-cleaning mobile manipulator using fuzzy wavelet neural network, *Fuzzy Sets and Systems*, vol.235, pp.65-82, 2014.

[12] R. Radharamanan and H. E. Jenkins, Laboratory learning modules on CAD/CAM and robotics in engineering education, *International Journal of Innovative Computing, Information and Control*, vol.4, no.2, pp.433-443, 2008.

[13] L. Lin, H. B. Ren and H. R. Wang, RBFNN-based sliding mode control for robot, *Control Engineering of China*, vol.14, no.2, pp.224-226, 2007.

[14] P. Shi, Y. Zhang and R. K. Agarwal, Stochastic finite-time state estimation for discrete time-delay neural networks with Markovian jumps, *Neurocomputing*, vol.151, pp.168-174, 2015.