Sliding Mode Controller with RBF Neural Network for Manipulator Trajectory Tracking

Haitao Zhang, Mengmeng Du and Wenshao Bu

***Abstract*—In order to decrease the chattering generated by the sliding mode controller (SMC) in the manipulator trajectory tracking, we present a new adaptive sliding mode controller based on radial basis function (RBF) neural network. The sliding mode variable structure control is used for resisting disturbance and guaranteeing the system stability, and the RBF neural network is introduced to reduce the switching gain through self-learning ability. The input of RBF neural network is the sliding mode function, and its output is the switching gain which can be adjusted adaptively. Under the condition of existing model errors and external disturbances, the simulation studies on the multi-joint rigid manipulator show that the proposed algorithm can obtain good performance both in tracking the trajectory and reducing the chattering.**

***Index Terms*—manipulator, RBF neural network, sliding mode variable structure control, trajectory tracking**

I. Introduction

The manipulator system is a time-varying, strong coupling and nonlinear system. The manipulator trajectory tracking have attracted considerable attentions of scholars [1]-[3].

Sliding-mode control is a robust design methodology using a systematic scheme based on a sliding surface and Lyapunov's stability theorem. The main advantage of SMC is that the system uncertainties and external disturbances can be accommodated because of the invariance characteristics of system sliding conditions. However, the switching control in SMC results in control gain chattering. When the model of the manipulator is precisely known, Zhai et al. [4] proposed a new improved dual power reaching law based on the traditional sliding mode control. The adaptive items were added based on the dual power reaching law, the variables γ and k were adjusted to effectively improve the approaching speed. The simulation result shows that the method has less chattering for the manipulator trajectory tracking. However, manipulator system is a complex nonlinear system, whose dynamic parameters are difficult to be forecasted precisely. In fact, it is almost impossible to obtain exact dynamic models because of such uncertainties as nonlinear frictions and flexibilities of the joints and links of manipulator. In the literature [5], the genetic algorithm was used to optimize the switching function, the size of the chattering as the index of fitness function was optimized, and a switching function with relative minimum chattering was constructed. Feiy et al. [6] presented a sliding mode control based on disturbance observer for robot, the coupling system was decoupled by the feedback joint angle; it achieved high precision, but the simple control structure demanded that the control parameters must be known. In literature [7], the sliding mode control was applied to deal with the robust control of space robot in capturing operation of the target and controlling the spacecraft motion under unknown parameters. The saturation function was introduced in order to avoid the chattering phenomena, the simulation results proved the feasibility of the algorithm.

The self-learning characteristics and high parallel computing characteristic of neural network are very powerful, it can approximate the nonlinear systems with arbitrary accuracy, and it also has a strong robustness. In addition, aiming at the approximation errors of the neural network, most of the researches get the result that the tracking errors can be uniformly ultimately bounded or can be kept arbitrarily small if some gain parameters are sufficiently large. Lin et al. [8] combined sliding mode control and RBF neural network to design the controller. The output of RBF neural network was regarded as the input of sliding mode controller. It achieved a certain effect on eliminating the chattering, but it used the objective function to estimate the network weights, so it can not realize the adaptive weight updated online. The controller of literature [9] was composed of SMC, RBF neural network, and fuzzy control. A Lyapunov function was selected for the design of the SMC, and RBF neural network was proposed to compute the equivalent control. The weights of the RBF neural network were adjusted according to an adaptive algorithm. Fuzzy logic was used to adjust the gain of the corrective control of the SMC. The real time implementations indicated that the proposed method can be applied to manipulator trajectory control. In this paper, the RBF neural network and sliding mode controller is designed serially, RBF neural network approximate the switching gain, and the robust term is used to eliminate the neural network error. The simulation results demonstrate that the chattering and the steady state errors are eliminated and satisfactory trajectory tracking is achieved.

The paper is organized as follows. In section 2, the related knowledge of the manipulator is introduced. In section 3, the slide model control method is described, and its disadvantages are discussed; the sliding mode controller with

Manuscript received April 17, 2015; revised July 5, 2015. This work was supported in part by the key scientific research project of universities and colleges of Henan Province under Grant 15A413003.

Haitao Zhang is with the Information Engineering College, Henan University of Science and Technology, Luoyang, CO 471023 China (corresponding author: e-mail: zhang_haitao@163.com).

Mengmeng Du is with the Information Engineering College, Henan University of Science and Technology, Luoyang, CO 471023 China.

Wenshao Bu is with the Information Engineering College, Henan University of Science and Technology, Luoyang, CO 471023 China.

RBF neural network is designed and the stability proof is presented. Section 4 gives an example of a two degrees of freedom robot arm to show the effect of the proposed method. In Section 5 the concluding remarks are discussed.

II. Preliminaries

A. Model of Manipulator

A standard method for deriving the dynamics equations of a mechanical system is via the Euler-Lagrange equations. Using this method, the dynamics equations of a n degree of freedom rigid manipulator can be described in the following general form [10]:

$$D(q)\ddot{q}+C(q,\dot{q})\dot{q}+G(q)+\tau_d=\tau \qquad (1)$$

Where $D(q)$ is an $n\times n$ inertia matrix, which is a positive definite matrix. $C(q,\dot{q})$ is an $n\times n$ matrix containing the centrifugal and Coriolis forces. $G(q)$ is an $n\times 1$ vector containing gravity torques. q is joint position, $\dot{q}$ is joint velocity, $\ddot{q}$ is joint acceleration. τ is joint drive torque. τ_d denotes the external disturbance and $\int_0^\infty \tau_d^T\tau_d dt$ is bounded.

The dynamic characteristics of manipulator system are as follows:

Property 1: Inertia matrix $D(q)$ is a symmetric positive definite matrix bounded by $d_1\|x\|^2\le x^TD(q)x\le d_2\|x\|^2$, where d_1 and d_2 are known positive constants.

Property 2: The matrix $D(q)-2C(q,\dot{q})$ is skew-symmetric, so it satisfies $x^T\dot{D}(q)x=2x^TC(q,\dot{q})x$, $x\in R^n$.

Property 3: The unknown disturbance satisfies $\|\tau_d\|_\infty\le b_d$, where b_d is a positive constant.

Our objective is to design a trajectory tracking controller whose output is the control torque τ, which can make the position of joints track the desired trajectory accurately.

B. RBF Neural Network

RBF neural network was proposed by J. Moody and C. Darken in the 1980s, and the corresponding theory was developed by Powell. It is a powerful feed forward neural network architecture [11]. This type of network was applied to the real multivariable interpolation problem and was first formulated as neural network by Broomhead and Lowe. In the control engineering, the RBF neural network is usually used as a tool for modeling nonlinear function up to a small error tolerance because of its good capabilities in function approximation.

The structure of typical RBF neural network is made up of a collection of parallel processing units called nodes as shown in Fig. 1:

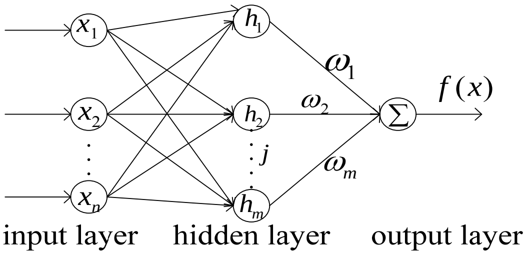

Fig. 1. The structure of RBF neural network

The RBF neural network has a feed forward architecture with an input layer, a hidden layer, and an output layer. The hidden layer is responsible for nonlinear transformation from the input space to the hidden space. The output layer is linear, which is designed to provide the response to the input signal.

There is a layer of processing units called hidden units between the inputs and outputs. Each of them is implemented by a radial basis function. The input layer of the network has n units for an n dimensional input vector. The input units are fully connected to the hidden layer units, which are in turn fully connected to the output layer units. In this paper, the number of input layer is 2, the number of hidden layer is 5, the number of output layer is 1.

The input-output mapping relationship is:

$$f(x)=\sum_{j=1}^{m}\omega^T h(x) \qquad (2)$$

Where x is the input signal of neural network, $f(x)$ is the output signal, $h(x)$ is the Gaussian basis function, and ω denotes neural network weights.

The activation function of the hidden layer is generally a Gaussian function which is expressed as:

$$h(x)=\exp(-\|x-c_j\|^2/b_j{}^2) \qquad (3)$$

Where c_j are the RBF centers in the input vector space. b_j denote the width of the node, $j=1,2,\cdots,m$.

Under the condition of the following assumptions, RBF neural network can approximate continuous functions with any degree of accuracy in a compact set [12].

1. The neural network output $\hat{f}(x,\hat{\omega})$ is continuous.

2. The ideal approximation of the neural network output is $\hat{f}(x,\omega^*)$, for a very small positive number ε_0, which has:

$$\max\|f(x)-\hat{f}(x,\omega^*)\|\le\varepsilon_0 \qquad (4)$$

Where ω^* is an $n\times n$ matrix, and denotes the best approximation of neural network weights.

Define the approximation error of ideal neural network, namely:

$$\eta=f(x)-\hat{f}(x,\omega^*) \qquad (5)$$

By the approximation capability of RBF neural network, the modeling error η is bounded, we assume that it is η_0.

$$\eta_0 = \sup \left\| f(x) - \hat{f}(x, \omega^*) \right\| \tag{6}$$

Where:

$$\hat{f}(x, \omega^*) = \omega^{*T} h(x) \tag{7}$$

III. The design of the controller

A. The Design of the Sliding Mode Controller

Sliding mode variable structure is a class of nonlinear control. The characteristic of this kind of control is that it has no fixed structure, and can change movement way according to the specific circumstances so as to achieve the presupposition state. Because the given trajectory has no association with the system state and the external disturbance, so the control system has strong robustness, and operation method of this control is simple.

The control objective is to drive the joint position q to the desired position q_d. Define the tracking error:

$$e = q - q_d \tag{8}$$

Define the sliding surface:

$$s = \dot{e} + \lambda e \tag{9}$$

Where $\lambda = diag[\lambda_1 \cdots \lambda_i \cdots \lambda_n], \lambda_i > 0$.

Define the reference state [13]:

$$\dot{q}_r = \dot{q} - s = \dot{q}_d - \lambda e \tag{10}$$

$$\ddot{q}_r = \ddot{q} - \dot{s} = \ddot{q}_d - \lambda \dot{e} \tag{11}$$

Assuming that there is no external interference τ_d, we choose the control input τ:

$$\tau = \hat{\tau} - K \operatorname{sgn} s \tag{12}$$

$$\hat{\tau} = \hat{D}\ddot{q}_r + \hat{C}\dot{q}_r + \hat{G} - As \tag{13}$$

Where $\hat{D}$, $\hat{C}$, $\hat{G}$ are the estimated values of D, C, G respectively, and $K = diag[K_{11} \cdots K_{ii} \cdots K_{nn}]$ is a diagonal positive definite matrix in which K_{ii} is a positive constant and $A = diag[a_1 \cdots a_i \cdots a_n]$ is also a diagonal positive definite matrix in whichis a_i a positive constant. $\operatorname{sgn}(\cdot)$ is a sign function, it is given as follows:

$$\operatorname{sgn}(s) = \begin{cases} 1 & , s > 0 \\ 0 & , s = 0 \\ -1 & , s < 0 \end{cases} \tag{14}$$

Substituting Eq. (12) and (13) into Eq. (1), we can get:

$$D\dot{s} + (C + A)s = \Delta f - K \operatorname{sgn} s \tag{15}$$

Where $\Delta f = \Delta D\ddot{q}_r + \Delta C\dot{q}_r + \Delta G$, $\Delta D = \hat{D} - D$, $\Delta C = \hat{C} - C$, $\Delta G = \hat{G} - G$.

Assuming that $|\Delta f_i| < |\Delta f_i|_{bound}$, where $|\Delta f_i|_{bound}$ is the boundary of $|\Delta f_i|$, we choose K_{ii} such that:

$$K_{ii} \geq |\Delta f_i|_{bound} \tag{16}$$

Define the Lyapunov function candidate:

$$L = s^T D(q) s / 2 \tag{17}$$

Differentiating (17) and considering Eq. (12), (13) and (15), we get [13]:

$$\begin{aligned} \dot{L} &= s^T D\dot{s} + s^T \dot{D} s / 2 \\ &= s^T[-(C + A)s + \Delta f - K \operatorname{sgn}(s) + Cs] \\ &= s^T[-As + \Delta f - K \operatorname{sgn}(s)] \\ &= s^T[\Delta f - K \operatorname{sgn}(s) - s^T As] \\ &= \sum_{i=1}^{n} [s_i(\Delta f_i - K_{ii} \operatorname{sgn}(s_i))] - s^T As \\ &\leq -s^T As \leq 0 \end{aligned} \tag{18}$$

The control law of (12) and (13) may be sensitive to uncertainties in the process of tracking and may lead to chattering. In addition, there is a difficulty to determine the switching gain of a sliding mode controller to achieve desired performance. At present, a trial and error procedure is commonly used to tune the parameters of the sliding mode controller. Thus the problem considered in our work is to propose a method to tune adaptively the switching gain K of the SMC in order to achieve accurate and robust tracking of the manipulator with minimum chattering phenomena.

B. The Design of the RBF Neural Network Sliding Mode Controller (RBFNNSMC)

The design of the structure is shown as Fig. 2:

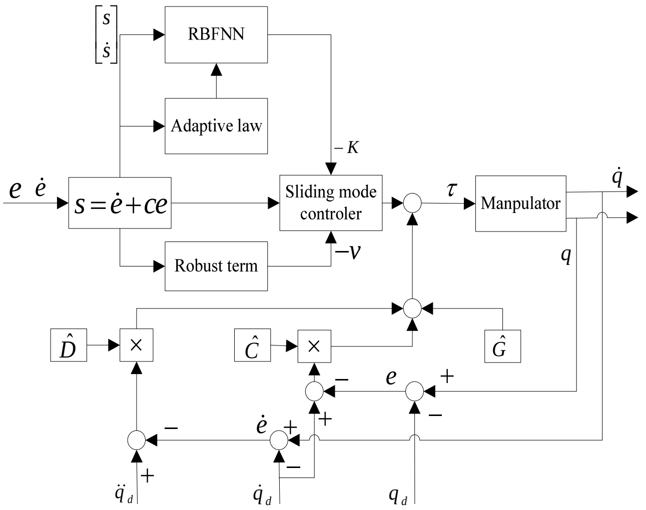

Fig. 2. The structure of trajectory tracking control with RBFNNSMC

Manipulator equation is shown as Eq. (1), the new control law is as follows:

$$\tau = \hat{D}\ddot{q}_r + \hat{C}\dot{q}_r + \hat{G} - As - K - v \tag{19}$$

Where $K = [k_1, k_2]$ is the output of RBF neural network of two joint manipulator; v is the robust term, $v = (\eta_0 + b_d)\,\mathrm{sgn}(s)$, it is used for overcoming the neural network approximation error η and external interference τ_d.

Substituting Eq. (19) into Equation (1), yield:

$$D\dot{s} = -(C + A)s + \Delta f - K - v \tag{20}$$

The adaptive law is designed as:

$$\dot{\tilde{\omega}}_i = P s_i h(s_i) \tag{21}$$

Where $i = 1,2$ represents the joint 1 and joint 2 of manipulator. P is a symmetric positive definite matrix, and its inverse matrix is existing.

Let us introduce the candidate Lyapunov function:

$$L = s^T D s / 2 + \sum_{i=1}^{2} (\tilde{\omega}_i^T P^{-1} \tilde{\omega}_i) / 2 \tag{22}$$

Where

$$\tilde{\omega}_i = \hat{\omega}_i - \omega_i^* \tag{23}$$

Differentiating (22), we get:

$$\begin{aligned}
\dot{L} &= (\dot{s}^T D s + s^T \dot{D} s + s^T D \dot{s}) / 2 + \sum_{i=1}^{2} (\dot{\tilde{\omega}}_i^T P^{-1} \tilde{\omega}_i + \tilde{\omega}_i^T P^{-1} \dot{\tilde{\omega}}_i) / 2 \\
&= (2 s^T D \dot{s} + s^T \dot{D} s) / 2 + \sum_{i=1}^{2} (\tilde{\omega}_i^T P^{-1} \dot{\tilde{\omega}}_i) \\
&= s^T (D\dot{s} + Cs) + \sum_{i=1}^{2} (\tilde{\omega}_i^T P^{-1} \dot{\tilde{\omega}}_i) \\
&= s^T [-(C + A)s + \Delta f - K - v + Cs] + \sum_{i=1}^{2} (\tilde{\omega}_i^T P^{-1} \dot{\tilde{\omega}}_i) \\
&= -s^T A s + s^T (\Delta f - K) + \sum_{i=1}^{2} (\tilde{\omega}_i^T P^{-1} \dot{\tilde{\omega}}_i) - s^T v \\
&= -s^T A s + \sum_{i=1}^{2} s_i (\Delta f_i - k_i) + \sum_{i=1}^{2} (\tilde{\omega}_i^T P^{-1} \dot{\tilde{\omega}}_i) - s^T v
\end{aligned} \tag{24}$$

Because of $k_i = \tilde{\omega}_i^T h(s_i) + \omega_i^{*T} h(s_i)$, we obtain:

$$\begin{aligned}
\dot{L} &= -s^T A s + \sum_{i=1}^{2} s_i (\Delta f_i - \tilde{\omega}_i^T h - \omega_i^{*T} h) + \sum_{i=1}^{2} (\tilde{\omega}_i^T P^{-1} \dot{\tilde{\omega}}_i) - s^T v \\
&= -s^T A s + \sum_{i=1}^{2} s_i (\Delta f_i - \omega_i^{*T} h) + \sum_{i=1}^{2} (-s_i \tilde{\omega}_i^T h + \tilde{\omega}_i^T P^{-1} \dot{\tilde{\omega}}_i) - s^T v \\
&= -s^T A s + \sum_{i=1}^{2} s_i (\Delta f_i - \omega_i^{*T} h) + \sum_{i=1}^{2} \tilde{\omega}_i^T (-s_i h + P^{-1} \dot{\tilde{\omega}}_i) - s^T v
\end{aligned} \tag{25}$$

Substituting Eq. (21) into Eq. (25), yield:

$$\dot{L} = -s^T A s + \sum_{i=1}^{2} s_i [\Delta f_i - \omega_i^{*T} h(s_i)] - s^T v \tag{26}$$

The existence of a very small positive real number γ_i, which makes the Eq. (26) satisfy:

$$\left| \Delta f_i - \omega_i^{*T} h(s_i) \right| \le \eta \le \gamma_i \left| s_i \right|, 0 < \gamma_i < 1 \tag{27}$$

Then :

$$s_i \left| \Delta f_i - \omega_i^{*T} h(s_i) \right| \le \gamma_i \left| s_i \right|^2 = \gamma_i s_i^2 \tag{28}$$

Therefore:

$$\begin{aligned}
\dot{L} &\le -s^T A s + \sum_{i=1}^{2} \gamma_i s_i^2 - s^T v \\
&= \sum_{i=1}^{2} -a_i s_i^2 + \gamma_i s_i^2 - \|s\| (\eta_0 + b_d) \\
&\le 0
\end{aligned} \tag{29}$$

Where $\gamma_i = diag[\gamma_1, \gamma_2]$, $a_i > \gamma_i$, $\eta_0 > 0$, $b_d > 0$.

According to Eq. (29), only when $s = 0$, $\dot{L} = 0$, the adaptive law asymptotic convergence. Finally, we may get the following conclusion:

$$\lim_{t \to \infty} s = \lim_{t \to \infty} (\dot{e} + \lambda e) = 0 \tag{30}$$

Namely:

$$\lim_{t\to\infty} q = q_d \, , \; \lim_{t\to\infty} \dot{q} = \dot{q}_d \tag{31}$$

From the above analysis it can be seen that the RBFNNSMC method can guarantee that the tracking errors converge arbitrarily close to zero. The following case study based on simulation will demonstrate this conclusion.

IV. Simulation Results

A. The Simulation of Two DOF manipulator

In this section, a simulation study is conducted to demonstrate the performance of our algorithm. A simple two degrees of freedom (DOF) manipulator is shown in Fig. 3. The dynamic equation is given as follows:

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + \tau_d = \tau \tag{32}$$

Where:

$$D(q) = \begin{pmatrix} p_1 + p_2 + 2p_3\cos q_2 & p_2 + p_3\cos q_2 \\ p_2 + p_3\cos q_2 & p_2 \end{pmatrix} \tag{33}$$

$$C(q,\dot{q}) = \begin{pmatrix} -p_3\dot{q}_2\sin q_2 & -p_3(\dot{q}_1 + \dot{q}_2)\sin q_2 \\ p_3\dot{q}_1\sin q_2 & 0 \end{pmatrix} \tag{34}$$

$$G(q) = \begin{pmatrix} p_4 g\cos q_1 + p_5 g\cos(q_1 + q_2) \\ p_5 g\cos(q_1 + q_2) \end{pmatrix} \tag{35}$$

$$\tau_d = [0.2\sin(t) \quad 0.2\sin(t)]^T \tag{36}$$

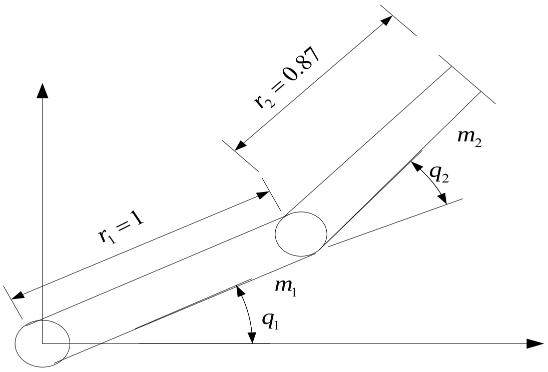

Fig. 3. Structure of the manipulator

The mass of link 1 is $m_1 = 2.04kg$, the mass of link 2 is $m_2 = 1kg$, the length of link 1 is $r_1 = 1m$, and the length of link 2 is $r_2 = 0.87m$. The $p_1 = (m_1 + m_2)r_1^2$, $p_2 = m_2 r_2^2$, $p_3 = m_2 r_1 r_2$, $p_4 = (m_1 + m_2)r_1$, $p_5 = m_2 r_2$, and $g = 9.8m/s^2$.

The desired trajectory of the manipulator is $q_{1d} = 2\sin(t)$ and $q_{2d} = \sin(t)$, and the initial states of the manipulator are $q_1(0) = q_2(0) = 0$, $\dot{q}_1(0) = \dot{q}_2(0) = 0$. The RBF neural network input is sliding mode function s and its differential term $\dot{s}$. The parameters of the Gaussian basis function are $c = \begin{pmatrix} -100 & -50 & 0 & 50 & 100 \\ -10 & -5 & 0 & 5 & 10 \end{pmatrix}$, $b = 50$.

The control parameters $\lambda = diag\{100,100\}$, $A = diag\{27,27\}$, $P = diag\{10,10,10,10,10\}$. In the robust term, $\eta_0 = 0.3$, $b_d = 0.2$. SIMULINK and S function is used to design the control system. In order to compare the advantages of the proposed method, under the same given parameters, we simulate the sliding mode control method and the proposed method. The simulation results are shown as Fig. 4-Fig. 12.

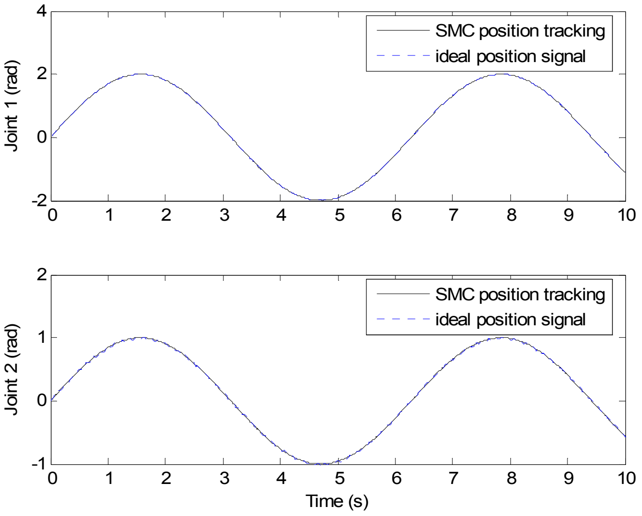

Fig. 4. Position tracking of SMC

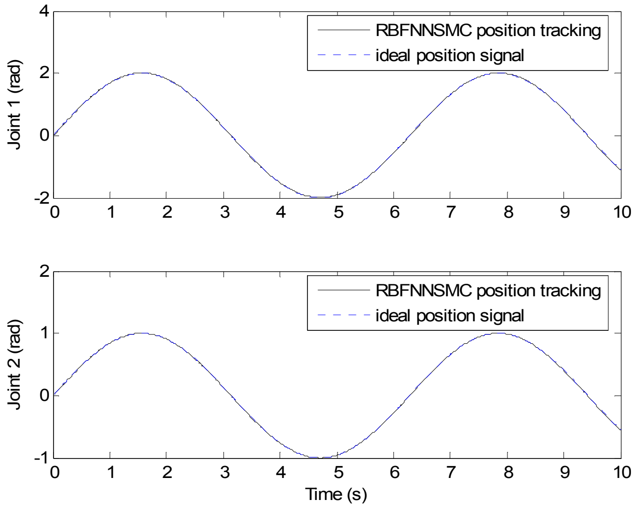

Fig. 5. Position tracking of RBFNNSMC

It can be seen from Fig. 4 and Fig. 5, the position tracking curves of joint 1 and 2 of manipulator are both ideal under the control of the two kind of algorithms.

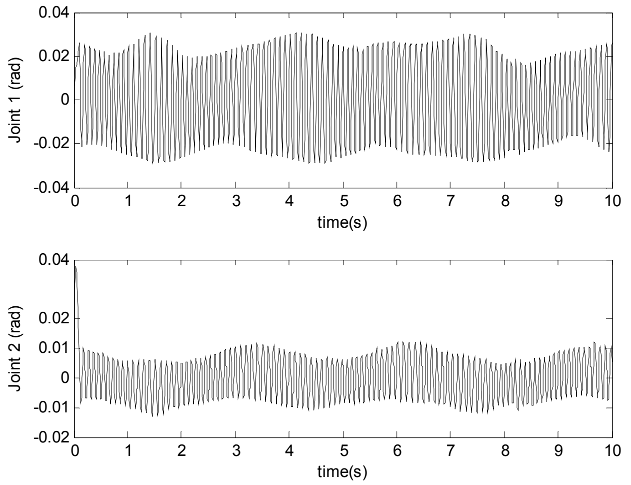

Fig. 6. Position tracking error of SMC

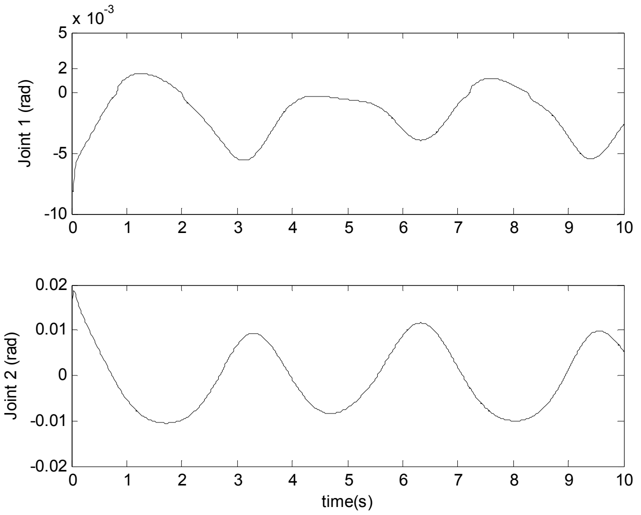

Fig.7. Position tracking error of RBFNNSMC

Fig. 6 and Fig. 7 show the position tracking error of the two joints. Fig. 6 shows that the position tracking error under the control of SMC, the sdeady error of joint 1 fluctuates between 0.025 to -0.025, and the sdeady error of joint 2 fluctuates between 0.01 to -0.01. Fig. 7 shows that the position tracking error under the control of RBFNNSMC, the steady state error of joint 1 ranges from 0.002 to -0.006, the steady state error of joint 2 ranges from 0.01 to -0.01. From the error comparison of joint 1, we can know that the control precision of RBFNNSMC is better than SMC. As to the steady state error of joint 2, it is hard to compare the RBFNNSMC quantitatively with SMC, but the initial error under the control of RBFNNSMC is smaller than SMC. From Fig. 6 and 7 we can prove, the global performance of the system is improved using the RBF neural network, and because of using of the robust control term, the adverse effects which caused by the neural network approximation error is compensated effectively.

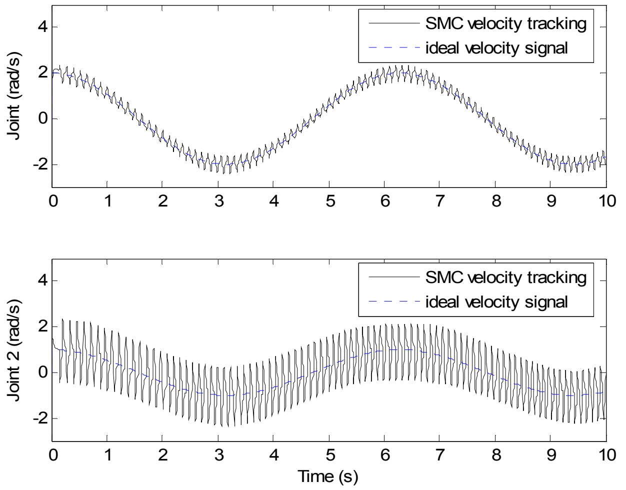

Fig. 8. Velocity tracking of SMC

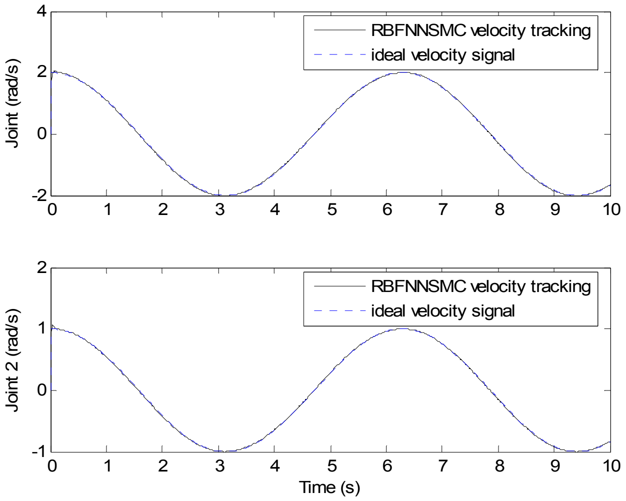

Fig. 9. Velocity tracking of RBFNNSMC

From Fig. 8 and Fig. 9, we can conclude that the velocity tracking errors have big fluctuation under the control of SMC. However, the velocity curve under the control of RBFNNSMC can trace the given velocity signal smoothly.

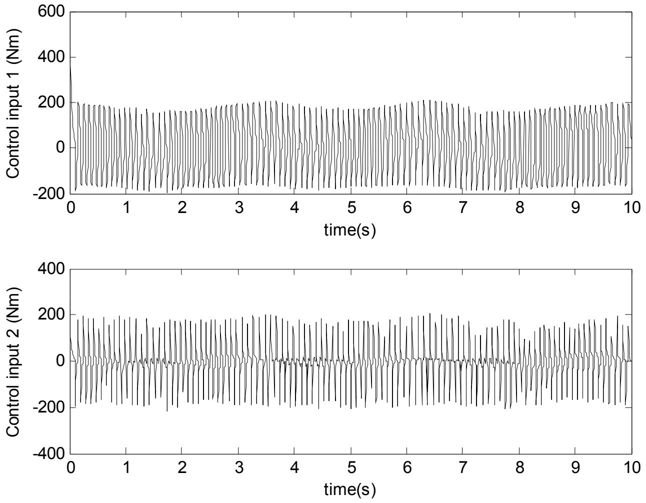

Fig. 10. Control torque of SMC

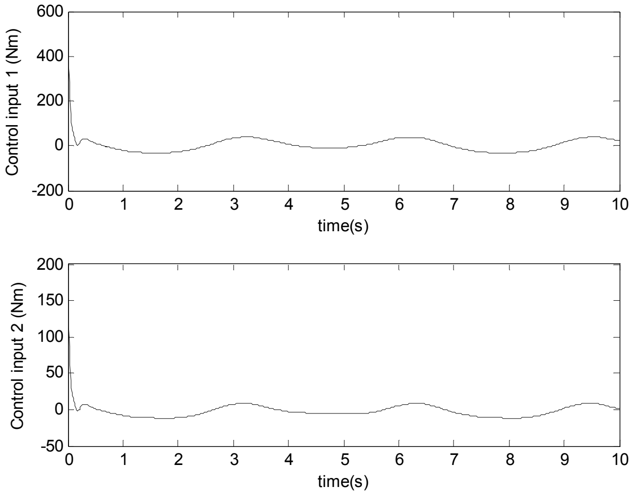

Fig. 11. Control torque of RBFNNSMC

Fig. 10 and Fig. 11 show the output control torque of manipulator, Fig. 10 shows the chattering of SMC is very big, so this control method is very difficult to apply in practice. After joining the RBF neural network controller, the system can adaptively adjust the switching gain, which reduces the system chattering caused by sliding mode control.

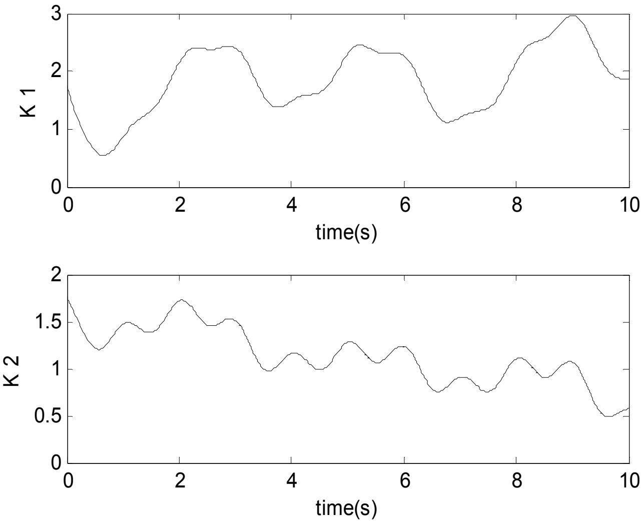

Fig. 12. The adaptive change of gain under the RBFNNSMC

Fig. 12 shows the changing process of switching gain under the adjustment of the RBF neural network.

B. The Simulation of Three DOF manipulator

In order to verify the effectiveness of the algorithm, we use the assembly manipulator in the literature [14] to do the simulation. It is a space manipulator with three degrees of freedom. The diagram is shown in Fig. 13, the first joint do the translational motion, the second joint and the third joint do the revolving motion.

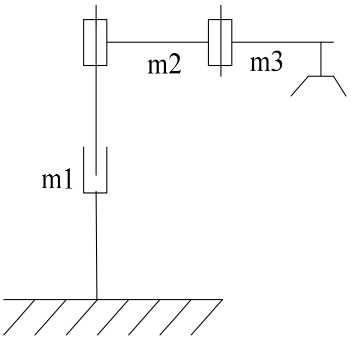

Fig.13. The diagram of the three DOF manipulator

We can know the parameters from the literature [14], the mass of link 1 is $m_1 = 1kg$, the mass of link 2 is $m_2 = 1kg$, the mass of link 3 is $m_3 = 1kg$, the length of link 2 is $l_1 = 0.2m$, and the length of link 3 is $l_2 = 0.2m$.

Dynamic matrices are shown below:

$$D(q) = \begin{bmatrix} a_{11} & 0 & 0 \\ 0 & a_{22} & a_{23} \\ 0 & a_{32} & a_{33} \end{bmatrix} \tag{37}$$

$$C(q,\dot{q}) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & b_{22} & b_{23} \\ 0 & b_{32} & 0 \end{bmatrix} \tag{38}$$

$$G(q) = \begin{bmatrix} (m_1 + m_2 + m_3)g \\ 0 \\ 0 \end{bmatrix} \tag{39}$$

Where

$$a_{11} = m_1 + m_2 + m_3 \tag{40}$$

$$a_{22} = m_2 l_2^2 / 3 + m_3 l_3^2 / 3 + m_3 l_2^2 + l_2 l_3 m_3 \cos(q_3) \tag{41}$$

$$a_{23} = a_{32} = m_3 l_3^2 / 3 + l_2 l_3 \cos(q_3) m_3 / 2 \tag{42}$$

$$b_{22} = -l_2 l_3 \sin(q_3) m_3 \dot{q}_3 / 2 \tag{43}$$

$$b_{23} = -m_3 l_2 l_3 \sin(q_3)\dot{q}_3 / 2 - m_3 l_2 l_3 \sin(q_3)\dot{q}_2 / 2 \tag{44}$$

$$b_{32} = -m_3 l_2 l_3 \sin(q_3)\dot{q}_2 / 2 \tag{45}$$

The desired trajectory of the manipulator is $q_{1d} = 2\sin(t)$, $q_{2d} = \sin(t)$, $q_{3d} = \sin(2t)$, and the initial states of the manipulator are $q_1(0) = q_2(0) = q_3(0) = 0$, $\dot{q}_1(0) = \dot{q}_2(0) = \dot{q}_3(0) = 0$. External disturbance is $\tau_d = [0.2\sin(t) \quad 0.2\sin(t) \quad 0.1\sin(t)]^T$. The RBF neural network parameters are $c = \begin{bmatrix} 10 & 10 & 10 & 10 & 10 \\ 1 & 1 & 1 & 1 & 1 \\ 5 & 5 & 5 & 5 & 5 \end{bmatrix}$ and $b = 5$.

The inputs are the sliding mode function of the three joints respectively, namely, $s(1)$, $s(2)$ and $s(3)$. The structure of the RBF neural network is 3-5-1. The control parameters $\lambda = diag\{80,80\}$, $A = diag\{30,30\}$ and

$P = diag\{10,10,10,10,10\}$. In the robust term, $\eta_0 = 0.3$, $b_d = 0.2$. The simulation results are shown as Fig. 14-Fig. 18 .

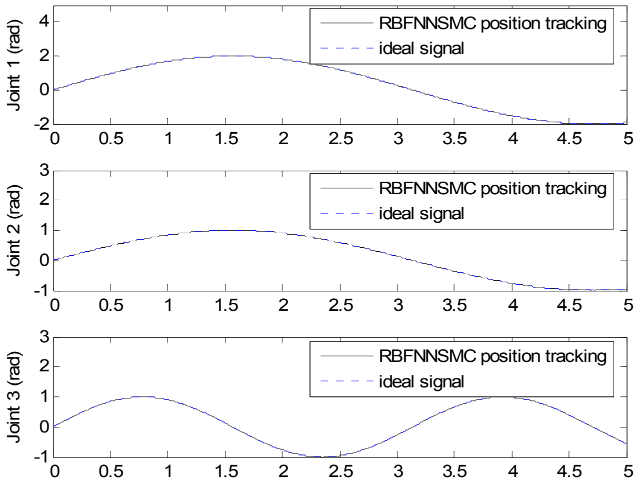

Fig. 14. Position tracking of RBFNNSMC

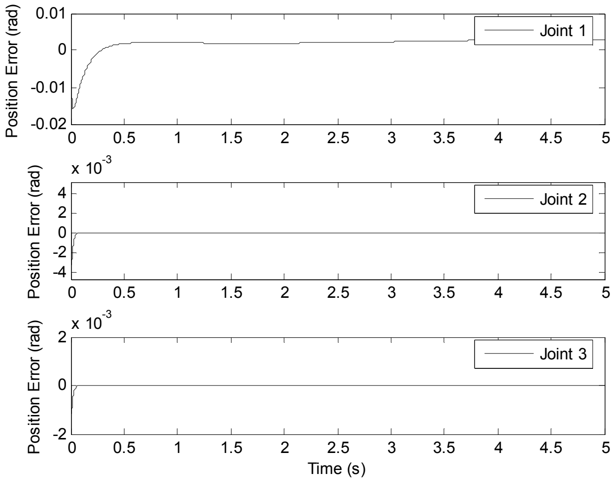

Fig.15. Position tracking error of RBFNNSMC

We can conclude from the Fig. 14 that tracking curves of the three joints are very ideal. Fig. 15 are the position error curves of three joints. The tracking error of joint 1 is 0.002; the tracking error of the joint 2 and joint 3 is also very small, it is 10^{-3} order of magnitude.

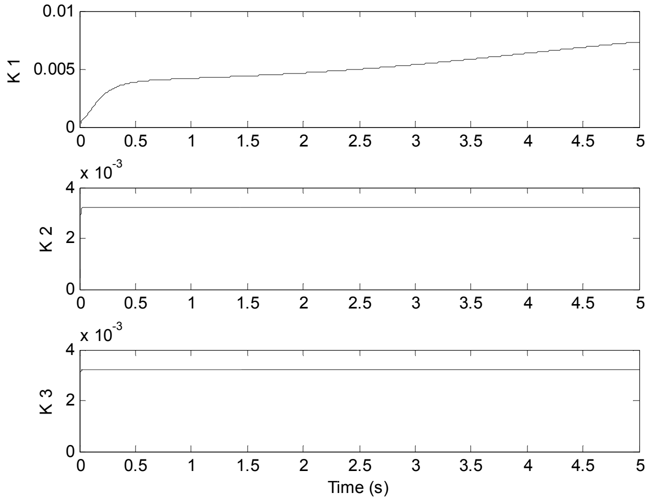

Fig. 16. The adaptive change of gain under the RBFNNSMC

Fig. 16 is the changing process of switching gain under the adjustment of the RBF neural network.

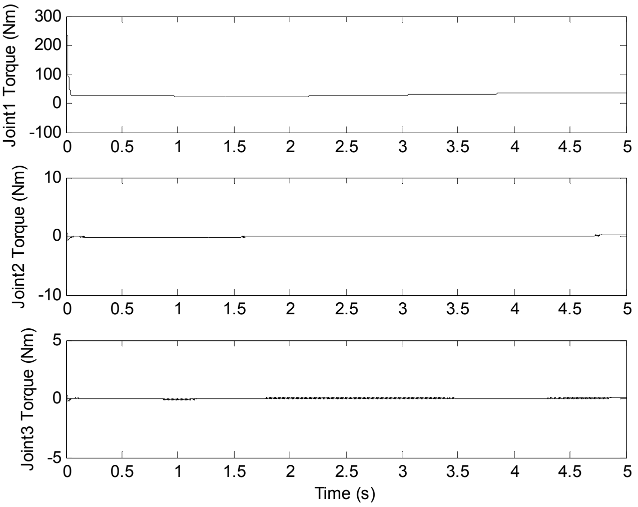

Fig. 17. Control torque of RBFNNSMC

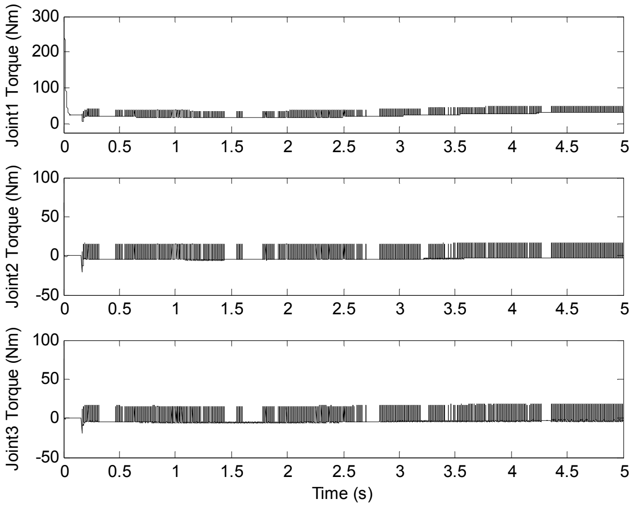

Fig. 18. Control torque of SMC

Fig. 17 is the output torque under the control of RBFNNSMC. Fig. 18 is the output torque under the control of traditional SMC. Because of the gravity term of joint 2 and joint 3 are zero, so the output torques are very small in the Fig. 17 and Fig. 18. We can see that the chattering is very big under the control of traditional SMC. However, there is almost no chattering under the control of RBFNNSMC.

V. CONCLUSION

This paper mainly studies the multi-joint manipulator system with uncertainties for the trajectory tracking control. This paper combines the sliding mode control with RBF neural network. The sliding mode control is used to resist interference and ensure the stability of the system, RBF neural network is used to adjust the sliding mode gain online, so as to reduce the chattering of output torque. The RBF neural network approximation error is overcome by adding the robust term. The design guarantees the closed-loop stability by using Lyapunov method. The simulation results show that the proposed control method is appropriate for design of manipulator with uncertainty and the external interference.

REFERENCES

[1] M. Rasheedat, Mahamood, "Improving the Performance of Adaptive PDPID Control of Two-Link Flexible Robotic Manipulator with ILC," Engineering Letters, vol. 20, no. 3, pp. 259-270, 2012.

[2] J. S. Kong, E. H. Lee, B. H. Lee and J. G. Kim, " Study on the real- time walking control of a humanoid robot using fuzzy algorithm," International Journal of Control, Automation and Systems, vol. 6, no. 4, pp. 551-558, 2008.

[3] P. K. Vempaty, K. C. Cheok, R. N. K. Loh, and S. Hasan, "Model Reference Adaptive Control of Biped Robot Actuators for Mimicking Human Gait," Engineering Letters, vol. 18, no. 2, pp. 165-174, 2010.

[4] W. N. Zhai，Y. W. Ge and S. Z. Song, "Sliding Mode Control for Robotic Manipulators Based on the Improved Reaching Law," Information and Control, vol. 43, no. 3, pp. 300-305, 2014.

[5] C. F, Wang Y N, He J, Long Y H, "GA-NN-integrated sliding-mode control system and its application in the printing press," Control Theory and Applications, vol. 20, no. 2, pp. 217-222, 2003.

[6] N. FEIY, J. S. SMITH and Q. H. Wu, "Sliding mode control of robot manipulators based on sliding mode perturbation observation," Systems and Control Engineering, vol. 220, no. 1, pp. 201-210, 2006.

[7] T. Kobayashi and S. Tsuda, "Sliding Mode Control of Space Robot for Unknown Target Capturing," vol. 19, no. 2, pp. 105-111, 2011.

[8] 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.

[9] A. Gokhan Ak, G. Cansever and A, Delibai, "Trajectory tracking control of an industrial robot manipulator using fuzzy SMC with RBFNN," Gazi University Journal of Science, vol. 28, no. 1, pp. 141-148, 2015.

[10] M. Tan, D. Xu, Z. G. Hou, S. Wang and Z. Q. Cao. Advanced robot control. Beijing: Higher Education Press, 2007, pp. 452-454.

[11] Y. S. Yang and X. F. Wang, "Adaptive $H\infty$ tracking control for a class of uncertain nonlinear systems using radial-basis-function neural networks," Neurocomputing, vol. 70, no. 4, pp. 935-941, .2007.

[12] H. R. Wang, C. N. Liu and Y. X. Zhang, "A neural network robust control based on the dissipative theory for robot,". Control Engineering China, vol. 17, no. 6, pp. 853-855, 2010.

[13] J. K. Liu. Design and Matlab simulation of robotic control systems. Beijing, Qinghua university press, 2008

[14] Q. Liu. Research and application of error analysis and compensation methods of assembling manipulator. Zhenjiang, Jiangsu University, 2014.