ICIC Express Letters
Part B: Applications
Volume **7**, Number **12**, December **2016**

ICIC International ©2016 ISSN 2185-2766
pp. **2641–2648**

FUZZY NEURAL NETWORK AND SLIDING MODE CONTROL OF ROBOT MANIPULATOR DRIVE SYSTEM

Yubin Fan, Haisheng Yu, Jinpeng Yu and Yingxin Liu

College of Automation and Electrical Engineering
Qingdao University
No. 308, Ningxia Road, Qingdao 266071, P. R. China
yu.hs@163.com

Received June 2016; accepted September 2016

Abstract. *In order to improve the control precision of two-link robot manipulator servo system, a sliding mode fuzzy-neural-network (FNN) controller is designed in this paper. The precise position control of robot servo system is a more complicated problem due to its friction forces, external disturbances and parameter variations. A five-layer FNN is designed to control the nonlinear part of dynamics. The robot is driven by the permanent magnet synchronous motor (PMSM). The current controller of the PMSM is designed as sliding mode controller (SMC). The overall control system possesses global asymptotic stability according to Lyapunov stability theory. The simulation results show that the control method has fast speed responses and good tracking performance.*
Keywords: Robot manipulator, PMSM, Fuzzy neural network, Sliding mode control

1. **Introduction.** In recent years, a large number of controllers for robot manipulators are studied. However, robot manipulators are difficult to establish an appropriate mathematical model due to their uncertainties and nonlinearity [1]. The current trend of control methods focuses on integrating nonlinear control with intelligent control, such as adaptive neural network [2-4], fuzzy neural network [5], and adaptive fuzzy neural network [6].

FNN is a kind of control method which has the advantages of fuzzy systems and neural networks. It can be used in the servo control of the robot manipulators and can solve the problems of robot manipulators servo system due to its good control effect and learning power. An adaptive fuzzy neural network control compensation method is used to solve the accurate position tracking of the control vectors in a large range by using the sample training self-learning and the adaptive variable gain coefficient [7]. The fuzzy neural network control method based on the hybrid learning algorithm can reduce the amount of computation of the neural network, and enhance the ability of the robot to respond to environmental changes [8]. Using the robust fuzzy neural network (RFNN) sliding-mode control, the motion tracking performance is significantly improved, and robustness to parameter variations, external disturbances, cross-coupled interference, and friction force can be obtained as well [9]. As the robot is driven by the motors, the performance of the robot is affected by the motors' performance. FNN can be used in the robot manipulators which are driven by direct-current servo motor [10] or switched reluctance motor [11]. The maintenances of direct-current servo motor and switched reluctance motor are more difficult, and the reliabilities are poor. The PMSM is a kind of motor which has high torque to volume ratio, wide speed range, and reliable operation [12]. Therefore, the PMSM is more suitable for the application in the robot joints. SMC has good robustness and fast speed response, so it can be designed to control the current of PMSM.

The contribution of this paper is to design a sliding mode fuzzy-neural-network controller. The control method not only has the learning ability of the FNN, but also has the fast response ability of the SMC. The fuzzy neural network control method is used

to control the nonlinear part and the uncertain part of the robot dynamics. Learning and approximation ability of fuzzy neural network can make the system achieve stability. Permanent magnet synchronous motor with powerful performance can get a good effect on the robot manipulator. For the current control of PMSM, the sliding mode control is adopted to get good control effect. At the same time, the integrated controller that has robot control and motor control, is closer to the actual and is conducive to the implementation of the industry. Additionally, a suitable Lyapunov function can be constructed, and it demonstrates that the overall control system is global asymptotic stable.

The rest of the paper is organized as follows. Section 2 outlines the two-link robot manipulator model. Then, the control principle of the robot manipulator servo system is described in Section 3. Stability analysis is given in Section 4. Section 5 is devoted to simulation results. At last, some conclusions are drawn.

2. Robot and Its Drive System Model.

Figure 1 shows the architecture of two-link robot manipulator. The drive motors are in A and B. Rotation range of joint 1 is $0° \leq \theta_1 \leq 180°$. Rotation range of joint 2 is $-90° \leq \theta_2 \leq 90°$. The maximum motion range of end-effector is a semicircle. The radius of semicircle is $L_1 + L_2$. The origin position of the robot end-effector is C_0, and it can be expressed as (x_0, y_0).

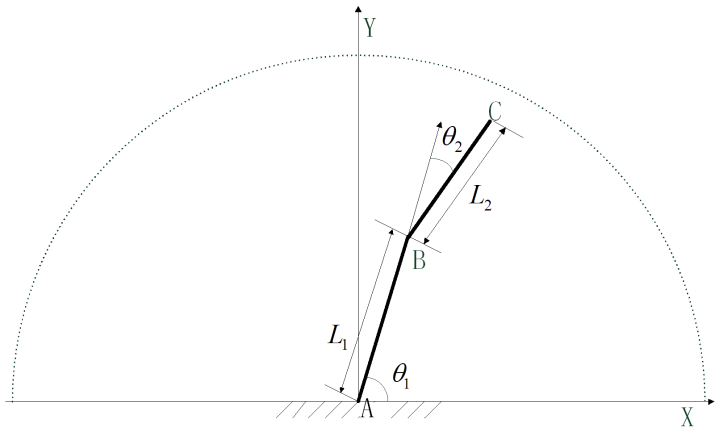


FIGURE 1. Architecture of two-link robot

2.1. Inverse kinematic model.

Let $C_d(x_d, y_d)$ be the desired position of the robot end-effector. According to the principle of inverse kinematics, we can get

$$\boldsymbol{\theta_d} = \begin{bmatrix} \theta_{1d} \\ \theta_{2d} \end{bmatrix} = \begin{bmatrix} \arctan(y_d/x_d) + \arccos\left(\left(x_d^2 + y_d^2 + L_1^2 - L_2^2\right)/2L_1\sqrt{x_d^2 + y_d^2}\right) \\ -\arccos\left(\left(x_d^2 + y_d^2 - L_1^2 - L_2^2\right)/2L_1L_2\right) \end{bmatrix} \tag{1}$$

2.2. Dynamic model.

The dynamics model of a two-link robot manipulator can be expressed in the following Lagrange equation from [13].

$$\boldsymbol{D}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{\tau} - \boldsymbol{\tau_f} \tag{2}$$

where $\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}} \in R^{2\times1}$ are the joint positions, velocity and acceleration vectors, respectively. $\boldsymbol{D}(\boldsymbol{q}) \in R^{2\times2}$ denotes the inertia matrix, $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in R^{2\times2}$ expresses the matrix of centripetal and Coriolis forces, $\boldsymbol{G}(\boldsymbol{q}) \in R^{2\times1}$ is the gravity vector, $\boldsymbol{\tau} \in R^{2\times1}$ is the vector of control torque developed, and $\boldsymbol{\tau_f} \in R^{2\times1}$ represents the vector of friction, external disturbance.

2.3. **The model of PMSM.** The drive motors of robot are PMSM. The model of PMSM can be described in synchronously rotating d-q reference frame as [14].

$$\begin{cases} L_d \dfrac{di_d}{dt} = -R_s i_d + n_p \omega L_q i_q + u_d \\ L_q \dfrac{di_q}{dt} = -R_s i_q - n_p \omega \phi - n_p \omega L_d i_d + u_q \\ J_m \dfrac{d\omega}{dt} = \tau - \tau_L - R_m \omega = n_p \left[\left(L_d - L_q \right) i_d i_q + \phi i_q \right] - \tau_L - R_m \omega \end{cases} \tag{3}$$

$$\tau = n_p \left[\left(L_d - L_q \right) i_d i_q + \phi i_q \right]$$

where L_d and L_q are d-axis and q-axis stator inductances, respectively. i_d, i_q and u_d, u_q are d-axis and q-axis stator currents and voltages, respectively. R_s is the stator resistance per phase, n_p is the number of pole pairs, J_m is the moment of inertia, and τ and τ_L are the electromagnetic and load torques, respectively. ϕ is the rotor flux linking of the stator, and ω is the angular velocity of motor, respectively.

3. **Controller Design.** According to the inverse kinematics model and dynamic model of two-link robot manipulator, the servo control system is designed as shown in Figure 2.

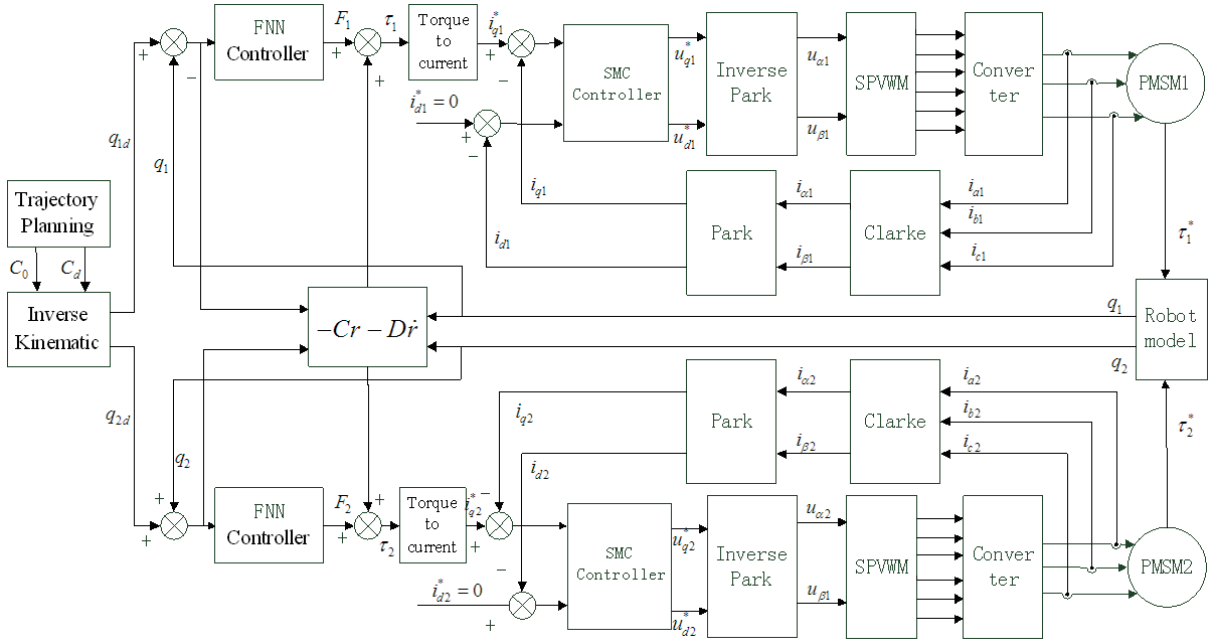


FIGURE 2. Servo control system of two-link robot manipulator

3.1. **FNN controller design.** As shown in Figure 2, when the origin and the desired position of the robot end-effector are known, from (1), we can get desired angular displacement is $\boldsymbol{q_d} = \boldsymbol{\theta_d} - \boldsymbol{\theta_0}$. The position error is $\boldsymbol{e(t)} = \boldsymbol{q_d} - \boldsymbol{q}$.

The error function is defined as

$$\boldsymbol{r} = \dot{\boldsymbol{e}} + \boldsymbol{\lambda e} \tag{4}$$

where $\boldsymbol{\lambda} = \boldsymbol{\lambda}^T > 0$, we get

$$\dot{\boldsymbol{q}} = -\boldsymbol{r} + \dot{\boldsymbol{q}}_{\boldsymbol{d}} + \boldsymbol{\lambda e} \tag{5}$$

$$\begin{aligned}\boldsymbol{D}(\boldsymbol{q})\dot{\boldsymbol{r}} &= \boldsymbol{D}(\boldsymbol{q})(\ddot{\boldsymbol{q}}_{\boldsymbol{d}} - \ddot{\boldsymbol{q}} + \boldsymbol{\lambda}\dot{\boldsymbol{e}}) \\ &= \boldsymbol{D}(\boldsymbol{q})(\ddot{\boldsymbol{q}}_{\boldsymbol{d}} + \boldsymbol{\lambda}\dot{\boldsymbol{e}}) - \boldsymbol{D}(\boldsymbol{q})\ddot{\boldsymbol{q}} \\ &= \boldsymbol{D}(\boldsymbol{q})(\ddot{\boldsymbol{q}}_{\boldsymbol{d}} + \boldsymbol{\lambda}\dot{\boldsymbol{e}}) + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) - \boldsymbol{\tau} + \boldsymbol{\tau}_{\boldsymbol{f}} \\ &= \boldsymbol{D}(\boldsymbol{q})(\ddot{\boldsymbol{q}}_{\boldsymbol{d}} + \boldsymbol{\lambda}\dot{\boldsymbol{e}}) - \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\boldsymbol{r} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})(\dot{\boldsymbol{q}}_{\boldsymbol{d}} + \boldsymbol{\lambda}\boldsymbol{e}) + \boldsymbol{G}(\boldsymbol{q}) - \boldsymbol{\tau} + \boldsymbol{\tau}_{\boldsymbol{f}} \\ &= \boldsymbol{F}(\boldsymbol{x}) - \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\boldsymbol{r} - \boldsymbol{\tau}\end{aligned} \tag{6}$$

where $\boldsymbol{F}(\boldsymbol{x}) = \boldsymbol{D}(\boldsymbol{q})(\ddot{\boldsymbol{q}} + \boldsymbol{\lambda}\dot{\boldsymbol{e}}) + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})(\dot{\boldsymbol{q}} + \boldsymbol{\lambda}\boldsymbol{e}) + \boldsymbol{G}(\boldsymbol{q}) + \boldsymbol{\tau}_{\boldsymbol{f}}$. Then, the robot dynamics equation can be described as

$$\boldsymbol{\tau} = \boldsymbol{F} - \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\boldsymbol{r} - \boldsymbol{D}(\boldsymbol{q})\dot{\boldsymbol{r}} \tag{7}$$

where $\boldsymbol{F}$ is an uncertain part of dynamics, and the parameters include the robot uncertainty and external disturbance. According to the principle of the FNN, the structure diagram is designed as shown in Figure 3. As shown in Figure 3, $\boldsymbol{x_1} = \boldsymbol{e}(\boldsymbol{t})$, $\boldsymbol{x_2} = \dot{\boldsymbol{e}}(\boldsymbol{t})$, $\boldsymbol{y} = \boldsymbol{F}(\boldsymbol{x})$. The signal propagation and the basic function in each layer of the FNN are introduced in the following:

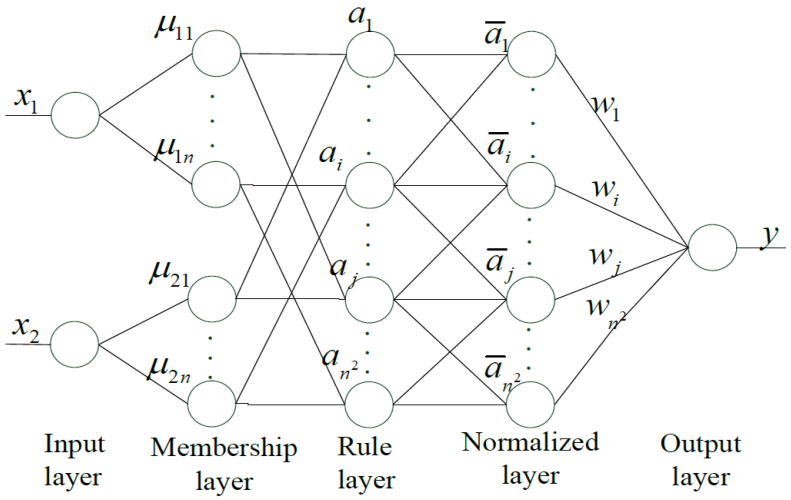


FIGURE 3. Structure of FNN

Input layer transmits the input linguistic variables x_1, x_2 to the next layer, where x_1, x_2 are error and error derivative.

Membership layer represents the input values with the following Gaussian membership functions

$$\mu_{ij}(x_i) = \exp\left[-(x_i - c_{ij})^2/\sigma_j^2\right], \quad i = 1, 2, \ldots, n, \quad j = 1, 2, \ldots, n \tag{8}$$

where c_{ij} and σ_j are the mean and standard deviation of the Gaussian function which are determined by self-learning ability of FNN.

Rule layer implements the fuzzy inference mechanism. The output is given as

$$a_k = \prod_{k=1}^{n^2} \mu_{ij}, \quad i = 1, 2, \ldots, n, \quad j = 1, 2, \ldots, n, \quad k = 1, 2, \ldots, n^2 \tag{9}$$

Normalized layer corresponds to the normalization of the results of each rule.

$$\bar{a}_k = a_k \Big/ \sum_{k=1}^{n^2} a_k, \quad k = 1, 2, \ldots, n^2 \tag{10}$$

Layer five is the output layer, and nodes in this layer represent output linguistic variables. Each node y can be represented as

$$y = \sum_{k=1}^{n^2} w_{ij}\bar{a}_k, \quad i = 1, 2, \ldots, n, \quad j = 1, 2, \ldots, n, \quad k = 1, 2, \ldots, n^2 \tag{11}$$

where w_{ij} are the weights between the normalized layer and the output layer.

The parameters of the fuzzy neural network can be obtained by self-learning ability. The error function of fuzzy neural network is defined as $E = (y_d - y)^2/2$, and we can get the learning algorithm for parameter as [11]

$$w_{ij}(k+1) = w_{ij}(k) - \beta\frac{\partial E}{\partial w_{ij}},\ c_{ij}(k+1) = c_{ij}(k) - \beta\frac{\partial E}{\partial c_{ij}},\ \sigma_{ij}(k+1) = \sigma_{ij}(k) - \beta\frac{\partial E}{\partial \sigma_{ij}} \quad (12)$$

where y, y_d are the expected output and actual output, and $\beta > 0$ is learning rate.

3.2. **SMC design.** As shown in Figure 2, $\boldsymbol{i_d^*}$, $\boldsymbol{i_q^*}$, $\boldsymbol{i_d}$ and $\boldsymbol{i_q}$ are d axis and q axis reference currents and feedback currents, respectively. The control $\boldsymbol{i_d^* = 0}$ is used in the system. The current error of d axis and q axis are $\tilde{\boldsymbol{i}}_{\boldsymbol{q}} = \boldsymbol{i_q^* - i_q}$, $\tilde{\boldsymbol{i}}_{\boldsymbol{d}} = -\boldsymbol{i_d}$, respectively.

According to the current errors $\tilde{\boldsymbol{i}}_{\boldsymbol{q}}$ and $\tilde{\boldsymbol{i}}_{\boldsymbol{d}}$, the designs of the sliding surface $\boldsymbol{s_1}$, $\boldsymbol{s_2}$ are

$$\boldsymbol{s_1 = c_1\tilde{i}_q},\ \boldsymbol{s_2 = c_2\tilde{i}_d} \quad (13)$$

where $\boldsymbol{c_1 = c_1^T > 0}$, $\boldsymbol{c_2 = c_2^T > 0}$. The exponential reaching law of sliding mode controller of q axis is designed as [14]

$$\boldsymbol{\dot{s}_1 = -\varepsilon_1 sgns_1 - k_1 s_1} \quad (14)$$

where $\boldsymbol{-\varepsilon_1 sgns_1}$ is constant reaching; $\boldsymbol{k_1 s_1}$ is exponential reaching; $\boldsymbol{\varepsilon_1}$ is constant reaching coefficient; $\boldsymbol{k_1}$ is exponential approximation coefficient. From (13)-(14), we get

$$\boldsymbol{\dot{s}_1 = c_1\dot{\tilde{i}}_q = c_1\left(\dot{i}_q^* - \dot{i}_q\right) = c_1\left(\dot{i}_q^* - \left(-R_s i_q - n_p\omega\phi - n_p\omega L_d i_d + u_q\right)\Big/ L_q\right)} \quad (15)$$

$$\boldsymbol{u_q = R_s i_q - n_p\omega\phi + n_p\omega L_d i_d + L_q\left(\varepsilon_1 sgns_1 + k_1 s_1/c_1 + \dot{i}_q^*\right)} \quad (16)$$

In the same way, we can get the control rule for the d axis as follows

$$\boldsymbol{u_d = R_s i_d - n_p\omega L_q i_q + L_d(\varepsilon_2 sgns_2 + k_2 s_2)/c_2} \quad (17)$$

3.3. **Torque to current.** As shown in Figure 2, the torque of the robot system is converted into the current of PMSM. From (3) and $\boldsymbol{\dot{i}_d^* = 0}$, we can get the current of q axis as

$$\boldsymbol{\dot{i}_q^* = \tau/n_p\phi} \quad (18)$$

4. **System Stability Analysis.** Define the Lyapunov function

$$V = \boldsymbol{r^T D(q) r/2 + s_1^T c_1 s_1/2 + s_2^T c_2 s_2/2} \quad (19)$$

where $\boldsymbol{D(q)} > 0$, $\boldsymbol{c_1 = c_1^T > 0}$, $\boldsymbol{c_2 = c_2^T > 0}$, $V > 0$. From (19), we get

$$\dot{V} = \boldsymbol{r^T D(q)\dot{r} + r^T\dot{D}(q) r/2 + s_1^T c_1\dot{s}_1 + s_2^T c_2\dot{s}_2} \quad (20)$$

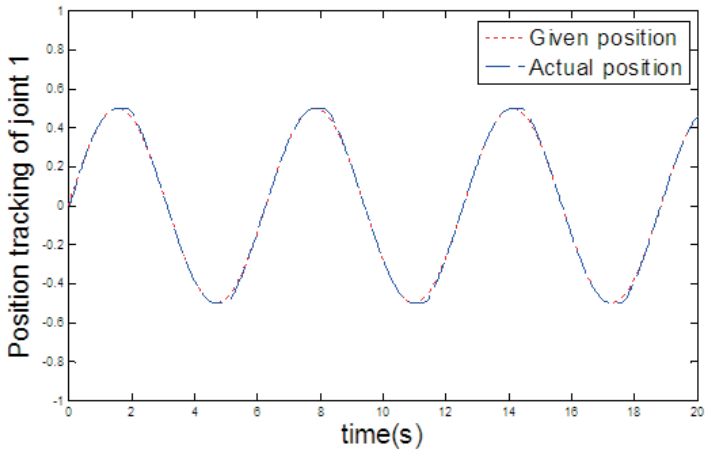
According to (14)-(17) and [2,14], we get

$$\begin{aligned}\dot{V} &= \boldsymbol{r^T D(q)\dot{r} + r^T\dot{D}(q) r/2 + s_1^T c_1(-\varepsilon_1 sgns_1 - k_1 s_1) + s_2^T c_2(-\varepsilon_2 sgns_2 - k_2 s_2)}\\ &\le 0\end{aligned} \quad (21)$$

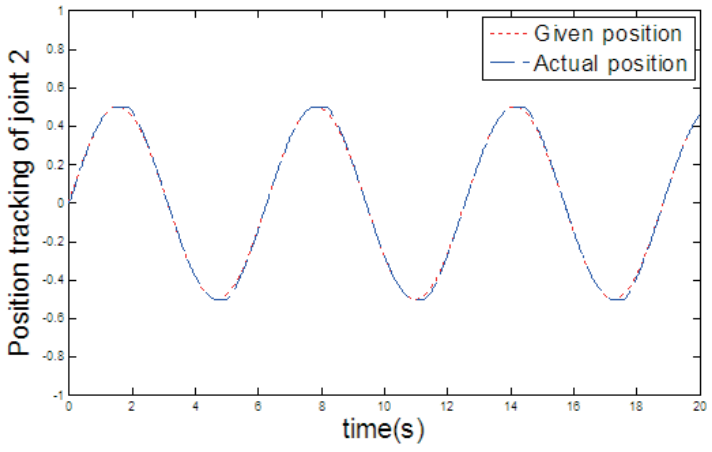
Only if $\boldsymbol{r = s_1 = s_2} = 0$, $\dot{V} = 0$. Therefore, $\dot{V}$ is negative definite and the system is global asymptotic stable according to Lyapunov's stability theory.

5. **Simulation Results.** The simulations are performed to evaluate the performance of the robot system by using Matlab/Simulink. The parameters of the PMSM 1 are: $R_{s_1} = 2.875\Omega$, $L_{d_1} = L_{q_1} = 0.0085$H, $\phi_1 = 0.175$Wb, $J_{m_1} = 0.025$Kg·m^2, $n_{p_1} = 4$, nominal power is 1.1KW, and nominal voltage is 220V. The parameters of the PMSM 2 are: $R_{s_2} = 2\Omega$, $L_{d_2} = L_{q_2} = 0.0054$H, $\phi_2 = 0.175$Wb, $J_{m_2} = 0.025$Kg·m^2, $n_{p_2} = 4$, nominal power is 0.5KW, and nominal voltage is 220V. The parameters of the robot are $m_1 = m_2 = 1$kg, $L_1 = L_2 = 0.5$m. The right system parameters are obtained through debugging. The parameters of controller are $\boldsymbol{\lambda} = diag[10\ 10]$, $\boldsymbol{c_1} = diag[100\ 100]$, $\boldsymbol{c_2} = diag[50\ 50]$, $\beta = 10$, $\boldsymbol{\varepsilon_1} = \boldsymbol{\varepsilon_2} = diag[0.05\ 0.05]$, $\boldsymbol{k_1} = \boldsymbol{k_2} = diag[1000\ 1000]$.

Figure 4 shows the positions tracking curve of joint 1 and joint 2. The parameters of positions are: $q_{1d} = 0.5\sin t$, $q_{2d} = 0.5\sin t$. As shown in Figure 4, the system has fast response, the response time is short and the position error is small. The system has good tracking performance.



(a)



(b)

FIGURE 4. Position tracking: (a) joint 1, (b) joint 2

The parameters of end-effector's motion trajectory are: semicircle, the origin point is $(2, 0)$, and the end point is $(1, 0)$. Figure 5(a) gives the position error curve of joint 1. The steady state time when PD control and sliding mode control are used is 3s. When fuzzy neural network control and sliding mode control are used, the time is reduced to 0.6s. Figure 5(b) gives the position error curve of joint 2, and the steady state time when PD control and sliding mode control are used is 2.5s. When fuzzy neural network control and sliding mode control are used, the time is reduced to 0.6s. The time to reach steady

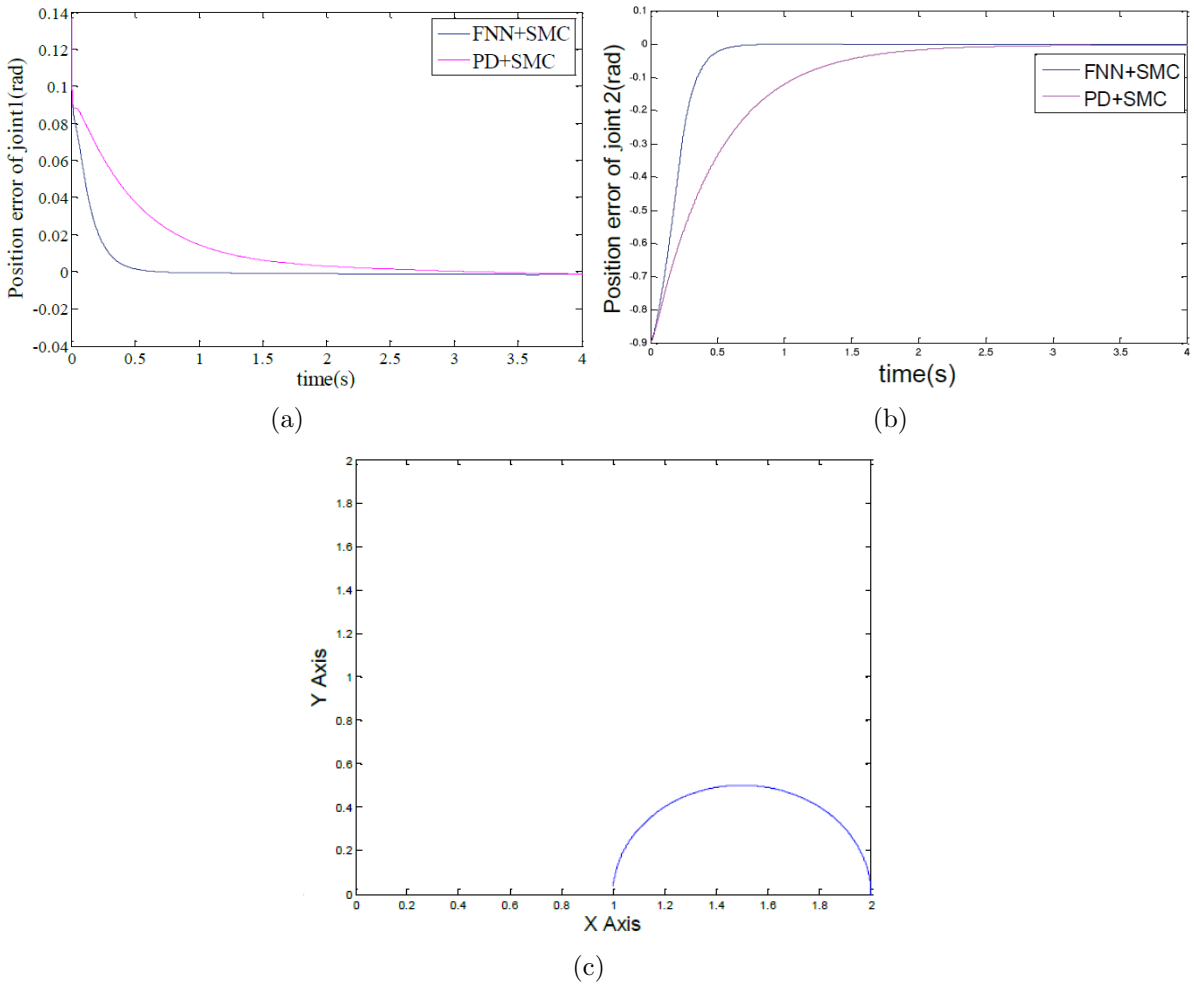


FIGURE 5. (a) Position error of joint 1, (b) position error of joint 2, (c) trajectory of robot

state is reduced. As shown in Figure 5(c), the trajectory is the same as planning. As a result, the fuzzy neural network sliding mode control can reduce steady state time and has accurate motion trajectory.

6. **Conclusions.** In this paper a novel control method is proposed for two-link robot manipulator that PMSM as drive motors. An effective FNN control method is developed in application to the robot position control system. FNN can control the nonlinear part of two-link robot manipulator servo system due to its approximation and learning ability. The control $\boldsymbol{i_d^*} = 0$ is used in PMSM control system and a sliding mode controller is designed to control the current. SMC can improve the active performances of the PMSM control system. Simulation results show that the proposed control method has good tracking performance and fast speed responses. In the future work, we will focus on the practical application of the proposed control algorithm.

Acknowledgment. This work is partially supported by the Natural Science Foundation of China (61573204, 61573203, 61501276) and Shandong Province Outstanding Youth Fund (ZR2015JL022).

REFERENCES

[1] R. J. Wai and R. Muthusamy, Fuzzy-neural-network inherited sliding-mode control for robot manipulator including actuator dynamics, *IEEE Trans. Neural Networks & Learning System*, vol.24, no.2, pp.274-287, 2013.

[2] X. Li and C. Cheah, Adaptive neural network control of robot based on a unified objective bound, *IEEE Trans. Control Systems Technology*, vol.22, no.3, pp.1032-1043, 2014.
[3] W. He and Y. Dong, Adaptive neural impedance control of a robotic manipulator with input saturation, *IEEE Trans. Systems, Man, and Cybernetics: Systems*, vol.46, no.3, pp.334-344, 2016.
[4] W. He, Y. Chen and Z. Yin, Adaptive neural network control of an uncertain robot with full-state constraints, *IEEE Trans. Cybernetics*, vol.46, no.3, pp.620-629, 2016.
[5] X. Liang, J. Zhang and W. Li, T-S fuzzy neural network control for autonomous underwater vehicles, *Electric Machines & Control*, vol.14, no.7, pp.99-104, 2010.
[6] R. J. Wai, Y. C. Huang, Z. W. Yang and C. Y. Shih, Adaptive fuzzy-neural-network velocity sensorless control for robot manipulator position tracking, *Control Theory & Applications*, vol.4, no.6, pp.1079-1093, 2010.
[7] X. Zhu and J. Zhou, Control compensation methods for hydraulic servo joint with adaptive-network-based fuzzy inference system, *Control Theory and Application*, vol.22, no.5, pp.694-698, 2005.
[8] X. Liang, J. Zhang and W. Li, T-S fuzzy neural network control for autonomous underwater vehicles, *Electric Machines and Control*, vol.14, no.7, pp.99-104, 2010.
[9] F. Lin and P. Shen, Robust fuzzy neural network sliding-mode control for two-axis motion control system, *IEEE Trans. Industrial Electronics*, vol.53, no.4, pp.1209-1225, 2006.
[10] R. J. Wai and Z. W. Yang, Adaptive fuzzy neural network control design via a T-S fuzzy model for a robot manipulator including actuator dynamics, *IEEE Trans. Systems Man & Cybernetics Part B Cybernetics*, vol.38, no.5, pp.1326-1346, 2008.
[11] B. M. Ge, J. H. Li and A. T. de Almeida, Fuzzy neural network control for robot manipulator directly driven by switched reluctance motor, *International Journal of Cognitive Informatics and Natural Intelligence*, vol.5, no.3, pp.573-577, 2011.
[12] H. Yu, J. Yu, X. Liu, Y. Zhao and Q. Song, Port-hamiltonian system modeling and position tracking control of PMSM based on maximum output power principle, *ICIC Express Letters*, vol.6, no.2, pp.437-442, 2012.
[13] Z. Cai, *Robotics*, Tsinghua University Press, Beijing, 2009.
[14] T. Lu, H. Yu and B. Shan, Adaptive sliding mode maximum torque per ampere control of permanent magnet synchronous motor servo system, *Control Theory & Applications*, vol.32, no.2, pp.251-255, 2015.