An Adaptive CMAC Neural Network Back-Stepping Controller for Robot Manipulators with Unknown Disturbances

Jun Li , Libiao Wang, Feixiang Jin, Zhengchu Wang

Department of Mechanical Engineering, Taizhou University, Taizhou, Zhejiang, 318000, China.
* Corresponding author, E-mail: lijun@tzc.edu.cn

***Abstract* - The performance of robot manipulators might degrade significantly due to the presence of unknown disturbances. An adaptive Cerebellar Model Articulation Controller (CMAC) neural network back-stepping controller is proposed to improve the system performance. The controller is composed of a CMAC neural network identification model and an adaptive back-stepping control. The identification model is used to approximate the unknown nonlinear function of the robot manipulators. Then the back-stepping control is able to track the desired trajectories based on the approximation function. Simulation results show that the proposed controller has strong robustness.**

Keywords - Adaptive control; Backstepping control; CMAC; Robot manipulators; Robust

I. INTRODUCTION

Robot manipulators are widely used in processing industries, such as surgery, manufacturing, aerospace, etc. However, robot manipulators are nonlinear dynamic system because of the imprecision manufacturing of the manipulator links, unknown loads, nonlinear friction and external disturbances. Therefore, design of a robust adaptive controller to improve the performance of robot manipulators is one of the most challenging tasks for engineers, especially when robot manipulators need to operate very quickly in response to external disturbances.

Recently, the sliding mode control [1, 2], Model Predictive Control [3] and intelligent control [4, 5] are used in robot manipulators so that the robot is able to track different types of reference trajectory. Although the robot control field has achieved significant progress, there are still some problems to be solved for practical implementations, particularly in the case of suffering perturbations and lack of accurate system dynamics.

In this paper, we adopt an Adaptive Cerebellar Model Articulation Controller neural network back-stepping Controller (ACMACNNBC) for a 2-DOF robot. The ACMACNNBC is composed of a CMAC neural network identification model and an adaptive back-stepping controller. In the most practice, it is unable to obtain the exact parameters of the system and external disturbances. To overcome this problem, the CMAC neural network identification is applied to approximate the system dynamic parameters. An adaptation law of the controller is synthesized using the Lyapunov stability theorems in order to guarantee global stability of the system.

II. DYNAMICS OF ROBOT MANIPULATOR

As is well known, neglecting friction, the nonlinear dynamic model of an *n* rigid-link robot manipulator is described by the following Lagrange form [6]:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G_g(q) + d = \tau \tag{1}$$

where $q,\dot{q},\ddot{q} \in \Re^n$ represent the joint position, velocity and acceleration vectors, respectively. $M(q) \in \Re^{n\times n}$ is the inertia matrix, $C(q,\dot{q}) \in \Re^{n\times n}$ is Coriolis and centripetal matrix, $G_g(q) \in \Re^{n\times 1}$ is the vector of gravity terms, $d \in \Re^{n\times 1}$ is a bounded unknown disturbances and $\tau \in \Re^{n\times 1}$ is the control input torque.

Property 1. The inertia matrix $M(q)$ is symmetric, b-ounded and positive definite.

Property 2. The matrix $\dot{M}(q) - 2C(q,\dot{q})$ is skew sy-mmetric and $C(q,\dot{q})$ is uniformly bounded.

Property 3. $G_g(q)$ and d are uniformly ultimately boun-ded (*UUB*). Defining $\|G_g(q)\|<G_b$ and $\|d\|<d_b$, G_b and d_b are known positive constants.

To simplify the problem, we choose a two-degree of freedom (2-DOF) robot manipulator as the control plant, which may serve as representative of many industrial processes. As illustrated in Fig. 1, Let $x_1 = q = [\theta_1, \theta_2]^T$, $x_2 = \dot{q}$, the dynamic system is expressed in Eq.(1) can be written as [7]:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= M^{-1}(x_1)\tau - M^{-1}(x_1)C(x_1,x_2)x_2 - M^{-1}(x_1)(G_g + d) \end{aligned} \tag{2}$$

And the matrices are given:

$$M(q)=\begin{bmatrix} P_1+P_2+2M_2r_2l_1\cos\theta_2 & P_2+M_2r_2l_1\cos\theta_2 \\ P_2+M_2r_2l_1\cos\theta_2 & P_2 \end{bmatrix} \quad (3)$$

$$C(q,\dot{q})=\begin{bmatrix} -M_2r_2l_1\dot{\theta}_2\sin\theta_2 & -M_2r_2l_1(\dot{\theta}_1+\dot{\theta}_2)\sin\theta_2 \\ M_2r_2l_1\dot{\theta}_1\sin\theta_2 & 0 \end{bmatrix} \quad (4)$$

$$G_g(q)=\begin{bmatrix} (M_1r_1+M_2l_1)g\cos\theta_1+M_2r_2g\cos(\theta_1+\theta_2) \\ M_2r_2g\cos(\theta_1+\theta_2) \end{bmatrix} \quad (5)$$

where $P_1=1.33M_1r_1^2+M_2l_1^2$, $P_2=1.33M_2r_2^2$, M_n, θ_n, l_n represent mass, joint position and length of link n, for $n=1,2$. In addition, r_1 is the distance between the center of gravity of the link 1 and the axis of the joint 1 and r_2 is the distance between the center of gravity of the link 2 and the axis of the joint 2. g is gravitation constant.

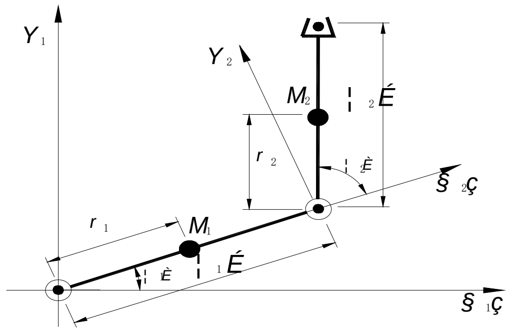

Fig. (1). a Two-DOF of Robot Manipulator

III. DESIGN OF THE CONTROLLER

A. Description of CMAC

The structure of CMAC includes input space, association memory space, receptive-field space, weight memory space and output space. According to Ref. [8, 9], the signal propagation and the basic function are introduced in each space of the CMAC as follows:

(1) Input space: $\mu=[\mu_1,\mu_2,\dots,\mu_L]^T\in\Re^L$ is the input. Each input state variable can be quantized into discrete regions (called an element) according to a given control space.

(2) Association memory space: Several elements are accumulated as a block, each block performs a receptive-field basis function. The Gaussian function is usually chosen as the receptive-field basis function

$$\varphi_{ji}(\mu_j)=\exp\left(\frac{-\left(\mu_j-C_{ji}\right)^2}{b_{ji}^2}\right),\text{for } i=1,2,\dots,H \quad (6)$$

where $\varphi_{ji}(\mu_j)$ denote the ith block receptive-field basis function for the jth input with the center of C_{ji} and the width b_{ji}, H is the number of the blocks.

(3) Receptive-field space: In this space, the mth multidimensional receptive-field function is defined as

$$\Phi_m(\mu)=\prod_{j=1}^{L}\varphi_{ji}(\mu_j),\text{for } m=1,2,\dots,N \quad (7)$$

where N is the number of receptive-field. The function can be written in a vector notation as

$$\Phi(\mu,C,b)=[\Phi_1,\Phi_m\dots,\Phi_N]^T \quad (8)$$

where $C=[C_{11},\dots,C_{L1},C_{12},\dots,C_{L2},\dots,C_{1H},\dots C_{LH}]^T$, and $b=[b_{11},\dots,b_{L1},b_{12},\dots,b_{L2},\dots,b_{1H},\dots b_{LH}]^T$.

(4) Weight memory space: Each location of the receptive-field space contains adjustable weights values in the weight memory space with N components can be expressed in a vector as

$$W=[W_1,W_m,\dots,W_N]^T \quad (9)$$

where W_m denotes the connecting weights value of the mth receptive-field.

(5) Output space: the output of CMAC NN is the algebraic sum of the activated weights in weight memory space, which can be written in a vector form as

$$y=W^T\Phi(\mu) \quad (10)$$

B. Design of ACMACNNBC

Assuming that y_d and α_1 are desired value of state variables x_1 and x_2, respectively, the error states between the state variables and the desired values can be defined as follows:

$$z_1=x_1-y_d \quad (11)$$

and

$$z_2=x_2-\alpha_1 \quad (12)$$

By substituting (12) into the time derivative of (11), we can acquire results in the following tracking error dynamics as

$$\dot{z}_1=\dot{x}_1-\dot{y}_d=x_2-\dot{y}_d=z_2+\alpha_1-\dot{y}_d \quad (13)$$

Select the virtual controller as $\alpha_1=-\lambda_1z_1+\dot{y}_d$, where $\lambda_1>0$.

Consider the following Lyapunov function candidate $V_1=\frac{1}{2}z_1^Tz_1$.

By differentiating Lyapunov function and using (13), the first derivative of the Lyapunov function can be written as

$$\begin{aligned}\dot{V}_1&=z_1^T\dot{z}_1=z_1^T(\dot{x}_1-\dot{y}_d)=z_1^T(\dot{x}_1-\dot{y}_d)=z_1^T(x_2-\dot{y}_d)\\&=z_1^T(z_2+\alpha_1-\dot{y}_d)=-\lambda_1z_1^Tz_1+z_1^Tz_2\end{aligned} \quad (14)$$

From (2) and (12), it can be obtained:

$$\dot{z}_2=\dot{x}_2-\dot{\alpha}_1=-M^{-1}Cx_2-M^{-1}(G_g+d)+M^{-1}\tau-\dot{\alpha}_1 \quad (15)$$

To make z_2 as small as possible, the following control input torque τ is selected as

$$\tau = -\lambda_2 z_2 - z_1 - F \tag{16}$$

where λ_2 is a positive constant, F is the nonlinear robot function.

In this case, consider the second Lyapunov function as follows

$$V_2 = V_1 + \frac{1}{2} z_2^T M z_2 \tag{17}$$

In view of (15) and using Property 2, differentiating (17) yields

$$\begin{aligned}\dot{V}_2 &= \dot{V}_1 + \frac{1}{2} z_2^T M\dot{z}_2 + \frac{1}{2}\dot{z}_2^T M z_2 + \frac{1}{2} z_2^T \dot{M} z_2 \\ &= -\lambda_1 z_1^T z_1 + z_1^T z_2 + z_2^T M(\dot{x}_2 - \dot{\alpha}_1) + z_2^T C z_2 \\ &= -\lambda_1 z_1^T z_1 + z_1^T z_2 + z_2^T(-Cx_2 + Cz_2 + \tau \\ &\quad - M\dot{\alpha}_1 - (G_g + d)) \\ &= -\lambda_1 z_1^T z_1 + z_1^T z_2 + z_2^T(f + \tau) - z_2^T(G_g + d)\end{aligned} \tag{18}$$

where $f = -C\alpha_1 - M\dot{\alpha}_1$. Substituting (16) into (18) yields

$$\dot{V}_2 = -\lambda_1 z_1^T z_1 - \lambda_2 z_2^T z_2 + z_2^T(f - F) - z_2^T(G_g + d) \tag{19}$$

The CMAC is adopted to approximate the unknown function f so that we can conclude that asymptotic stability of the system is guaranteed. The next step is to select an appropriate CMAC adaptive weight law, and then stability synthesis can be performed for closed-loop system by Lyapunov method.

Assume that the nonlinear function F is appr-oximated by CMAC NN for some constant known ideal weight W from (10) and expressed as

$$F = W^T \Phi(\mu) \tag{20}$$

Define the estimate of the value of (20) as

$$\hat{F} = \hat{W}^T \Phi(\mu) \tag{21}$$

with $\hat{W}$ the current values of the NN estimated weights.

It's provided by the adaptive weight law. So estimation error of the weight is

$$\tilde{W} = W - \hat{W} \tag{22}$$

The ideal weight is bounded by known positive values $W_{\max}$ as follows:

$$\|W\|_F \le W_{\max} \tag{23}$$

Since W is a constant, take a derivative with (22), it can be obtain $\dot{\tilde{W}} = -\dot{\hat{W}}$.

To speed up the convergence of the controller, the adaptive weights law is defined as

$$\dot{\hat{W}} = -kG\|z_2\|\hat{W} - z_2^T G\Phi(\mu) \tag{24}$$

where G is a positive definite matrix and k is a positive constant.

Select the global Lyapunov function candidate as $V = V_2 + \frac{1}{2} tr\left\{\tilde{W}^T G^{-1} \tilde{W}\right\}$.

By differentiating the yields of V, we obtain

$$\begin{aligned}\dot{V} &= \dot{V}_2 + tr\left\{\tilde{W}^T G^{-1}\dot{\tilde{W}}\right\} \\ &= -\lambda_1 z_1^T z_1 - \lambda_2 z_2^T z_2 + z_2^T(f - F) \\ &\quad - z_2^T(G_g + d) + tr\left\{\tilde{W}^T G^{-1}\dot{\tilde{W}}\right\} \\ &= -\lambda_1 z_1^T z_1 - \lambda_2 z_2^T z_2 + z_2^T(f - \hat{F} + \hat{F} - F) \\ &\quad - z_2^T(G_g + d) + tr\left\{\tilde{W}^T G^{-1}\dot{\tilde{W}}\right\}\end{aligned} \tag{25}$$

There exists optimal error between the estimation function $\hat{F}$ and the unknown function f in [10] such that

$$\left\|f - \hat{F}\right\| \le \varepsilon_0 \tag{26}$$

where ε_0 is a small positive constant. Obtained from (20)-(22), the formula (25) can be written as

$$\begin{aligned}\dot{V} &= -\lambda_1 z_1^T z_1 - \lambda_2 z_2^T z_2 + z_2^T(f - \hat{F} + \hat{W}^T\Phi(\mu) - W^T\Phi(\mu)) \\ &\quad - z_2^T(G_g + d) + tr\left\{\tilde{W}^T G^{-1}\dot{\tilde{W}}\right\} \\ &= -\lambda_1 z_1^T z_1 - \lambda_2 z_2^T z_2 + z_2^T(f - \hat{F}) - z_2^T\tilde{W}^T\Phi(\mu) \\ &\quad - z_2^T(G_g + d) + tr\left\{\tilde{W}^T G^{-1}\dot{\tilde{W}}\right\} \\ &= -\lambda_1 z_1^T z_1 - \lambda_2 z_2^T z_2 + z_2^T(f - \hat{F}) - z_2^T(G_g + d) \\ &\quad + tr\left\{\tilde{W}^T(G^{-1}\dot{\tilde{W}} - z_2^T\Phi(\mu))\right\}\end{aligned}$$

Then, using (24), (26), and Property 3 yields

$$\begin{aligned}\dot{V} &\le -\lambda_1 z_1^T z_1 - \lambda_2 z_2^T z_2 + \|z_2\|\varepsilon_0 + \|z_2\|(G_b + d_b) \\ &\quad + k\|z_2\| tr\left\{\tilde{W}^T(W - \tilde{W})\right\}\end{aligned} \tag{27}$$

Since Cauchy–Schwarz inequality

$$\begin{aligned}tr\left\{\tilde{W}^T(W-\tilde{W})\right\} &= (\tilde{W}, W)_F - \left\|\tilde{W}\right\|_F^2 \\ &\le \left\|\tilde{W}\right\|_F \|W\|_F - \left\|\tilde{W}\right\|_F^2 \le \left\|\tilde{W}\right\|_F (W_{\max} - \left\|\tilde{W}\right\|_F)\end{aligned}.$$

The formula (27) can be expressed as

$$\begin{aligned}\dot{V} &\le -\lambda_1 z_1^T z_1 - \|z_2\|(\lambda_2\|z_2\| \\ &\quad + k\left\|\tilde{W}\right\|_F(\left\|\tilde{W}\right\|_F - W_{\max}) - (\varepsilon_0 + (G_b + d_b)))\end{aligned} \tag{28}$$

This is a negative as long as the term in braces positive, and completing the square in braces term yields

$$\begin{aligned}&\lambda_2\|z_2\| + k\left\|\tilde{W}\right\|_F(\left\|\tilde{W}\right\|_F - W_{\max}) - (\varepsilon_0 + (G_b + d_b)) \\ &= k(\left\|\tilde{W}\right\|_F - W_{\max}/2)^2 - kW^2_{\max}/4 + \lambda_2\|z_2\| - (\varepsilon_0 + (G_b + d_b))\end{aligned} \tag{29}$$

which is guaranteed to be positive as long as

$$\|z_2\| > \frac{kW^2_{\max}/4 + (\varepsilon_0 + (G_b + d_b))}{\lambda_2} \tag{30}$$

Or

DOI 10.5013/IJSSST.a.17.36.04 ISSN: 1473-804x online, 1473-8031 print

$$\left\|\tilde{W}\right\|_F > W_{\max}/2+\sqrt{W^2{}_{\max}/4+(\varepsilon_0+(G_b+d_b))/k} \quad (31)$$

Owing to all parameters in above (30) and (31) such as k, ε_0, $W_{\max}$, λ_2, G_b and d_b are all positive constants. Therefore, $\|z_2\|$ and $\left\|\tilde{W}\right\|_F$ are demonstrated to be *UUB*. As a result, the fact that error z_2 is bounded, which guarantees z_1 and $\dot{z}_1$ bounded. With x_1 and x_2 bounded, which results in the desired trajectory y_d bounded. Moreover, $\hat{W}$ and $\hat{F}$ are also bounded because of $\tilde{W}$ bounded. Thus, $\dot{V}$ is negative outside a compact set so that the closed-loop system is asymptotically stable.

IV. SIMULATION RESULTS

In order to verify the better effect of the proposed controller than the conventional PID controller, the simulation tests have been carried out on the 2-DOF robot manipulator. The parameters of the robot manipulator are chosen as: M_1=1.5kg, M_2=0.765kg, l_1= l_2=0.25m, r_1= r_2=0.15m and g=9.8m/s^2. The values of external disturbances and the desired trajectories are set as d= [0.25sinπt 0.25sinπt]^T,D= [sin2.1πt sin2.1πt]^T, respectively. The parameters of PID controller proportional, integral, differential are selected as k_p=2.5, k_I=0.52 and k_D=0.03, respectively. The parameters of the ACMACNNB controller are selected as below: λ_1=30, λ_2=50, k=1.5 and G is a second-order unit positive matrix. The CAMCNN contains 3-nodes for each robot link to approximate unknown dynamics with centers of receptive-field function C_{ji} evenly space [-1, 1]. The widths are all selected as $b_{ji=}$1.5. The NN weights are initialized as 0.15 in simulation test.

Simulation results are shown in Fig. 2-Fig.8 As you can see in Fig. 2-3, the actual trajectories are tracking the desired trajectories by PID. The responses of the ACMACNNBC are performed in Fig. 4-5. The CMAC NN adaptive weights are depicted in Fig. 6-7 by using the adaptive law (24). To study the advantages of the proposed controller over the conventional PID controller, we compare the position tracking error of the two systems in the Fig. 8-9. The results show that the proposed controller can respond convergence rates faster and make smaller error than the PID controller with external disturbances.

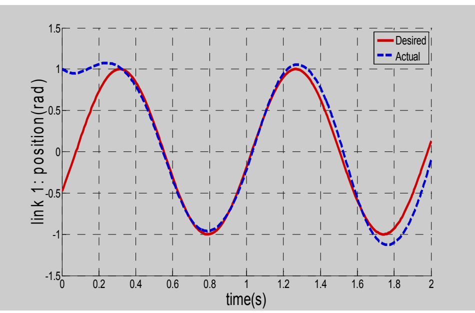

Fig. (2). Link 1 with PID Controller

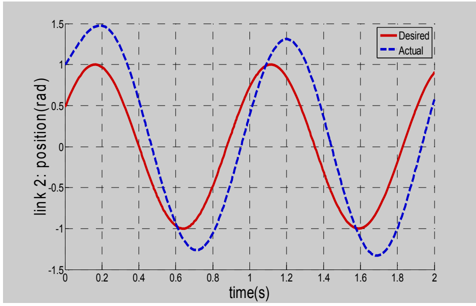

Fig. (3). Link 2 with PID Controller

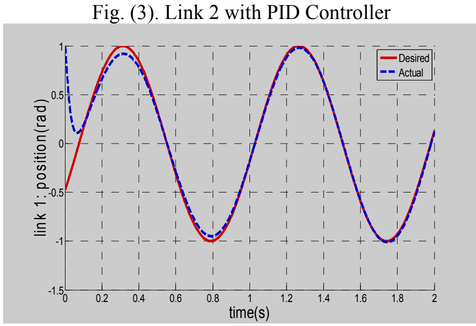

Fig. (4). Link 1 with ACMACNNB Controller

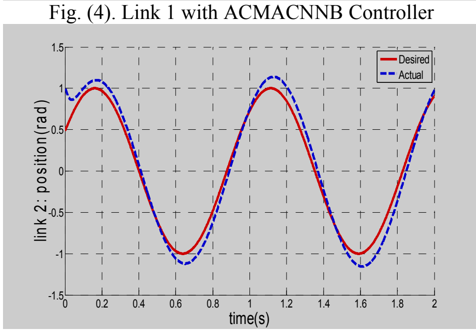

Fig. (5). Link 2 with ACMACNNB Controller

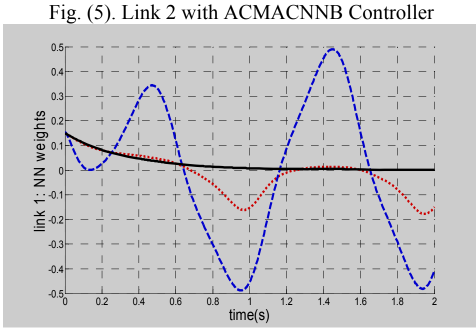

Fig. (6). the NN Adaptive Weights for Link 1

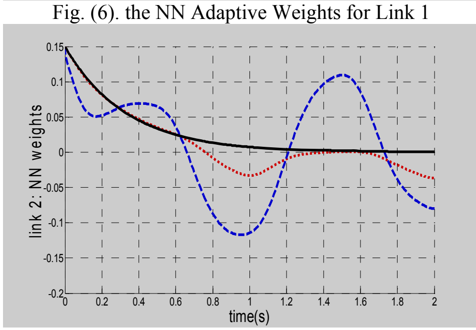

Fig. (7). the NN Adaptive Weights for Link 2

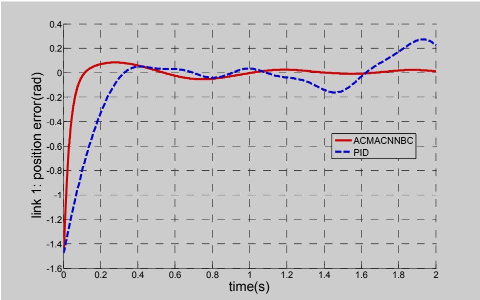

Fig. (8). Comparison of Both Controllers for Link 1

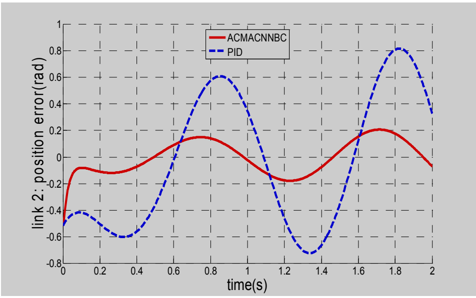

Fig. (9). Comparison of Both Controllers for Link 2

V. CONCLUSION

To achieve stability for the robot manipulator with unknown external disturbances in the course of the complex environment work, an adaptive controller has been developed based on back-stepping control theory. The architecture of the proposed controller consists of a back-stepping control and a CMAC NN identification model which copes with the unknown parameters and external disturbances. The overall control system is very easy to implement because of no a priori parameters of the robot manipulator dynamics. Another advantage of this approach is the fact that it tracks well with the desired trajectories and the convergence rates can be improved even if the existence of unknown external disturbances. Moreover, the stability of the closed-loop system is guaranteed by Lyapunov stability theorems so that the system has strong robustness.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflicts of interest.

ACKNOWLEDGMENT

This work was supported by the Science-Technology Foundation of Taizhou, Zhejiang province, China (No. 131KY10).

REFERENCES

[1] R. J. Lian, "Design of an enhanced adaptive self-organizing fuzzy sliding-mode controller for robotic systems", *Expert Systems with Applications,* vol. 39, pp.1545-1554, 2012.

[2] T. Sun, H. Pei, and Y. Pan, "Neural network-based sliding mode adaptive control for robot manipulators", *Neurcomputing,* vol. 74, pp. 2377-2384, 2011.

[3] L. Bossi, C. Rottenbacher, and G. Mimmi, "Multivariable predictive control for vibrating structures: An application", *Control Engineering Practice,* vol. 19, pp. 1087-1098, 2011.

[4] A. Forouzantabar, H. A. Talebi, and A. K. Sedigh, "Adaptive neural network control of bilateral teleportation with constant time delay", *Nonlinear Dynamics,* vol. 67, pp. 1123-1134, 2012.

[5] L. Cheng, Z. G. Hou, M. Tan, "Adaptive neural network tracking control for manipulators with uncertain kinematics, dynamics and actuator model", *Automatica,* vol. 45, pp. 2312-2318, 2009.

[6] Y. H. Kim, F. L. Lewis, "Neural network output feedback control of robotic manipulators", *IEEE Transactions on robotic and automation,* vol. 15, pp. 301-309, 1999.

[7] J. K. Liu, X. H. Wang, *Advanced sliding mode control for mechanical systems: Design, Analysis and MATLAB Simulation.* Springer: Berlin, 2011.

[8] Y. F. Peng, "Robust intelligent sliding model control using recurrent cerebellar model articulation controller for uncertain nonlinear chaotic systems", *Chaos, Solitons & Fractals,* vol. 39, pp. 150-167, 2009.

[9] C. F. Hsu, C. M. Chung, and C. M. Lin, C. Y. Hsu, "Adaptive CMAC neural control of chaotic systems with a PI-type learning algorithm", *Expert Systems with Applications,* vol. 36, pp. 11836-11843, 2009.

[10] C. M. Lin, Y. F. Peng, "Missile guidance law design using adaptive cerebellar model articulation controller", *IEEE Transactions on Neural Networks,* vol. 16, pp. 636-644, 2005.