Neural Comput & Applic (2016) 27:525–536
DOI 10.1007/s00521-015-1873-4

ORIGINAL ARTICLE

Adaptive trajectory tracking neural network control with robust compensator for robot manipulators

Pham Van Cuong[1,2] · **Wang Yao Nan**[1]

Received: 16 June 2014 / Accepted: 7 March 2015 / Published online: 21 March 2015

Abstract This paper presents an adaptive trajectory tracking neural network control using radial basis function (RBF) for an *n*-link robot manipulator with robust compensator to achieve the high-precision position tracking. One of the difficulties in designing a suitable control scheme which can achieve accurate trajectory tracking and good control performance is to guarantee the stability and robustness of control system, due to friction forces, external disturbances error, and parameter variations. To deal with this problem, the RBF network is investigated to the joint position control of an *n*-link robot manipulator. The RBF network is one approach which has shown a great promise in this sort of problems because of its fast learning algorithm and better approximation capabilities. The adaptive RBF network can effectively improve the control performance against large uncertainty of the system. The adaptive turning laws of network parameters are derived using the back-propagation algorithm and the Lyapunov stability theorem, so that the stability of the entire system and the convergence of the weight adaptation are guaranteed. In this control scheme, a robust compensator plays as an auxiliary controller to guarantee the stability and robustness under various environments such as the mass variation, the external disturbances, and modeling uncertainties. Finally, the simulation and experimental results in comparison with adaptive fuzzy and wavelet network control method are provided to verify the effectiveness of the proposed control methodology.

Keywords Robot manipulator · Neural network · RBF network · Sliding mode control · Adaptive control

1 Introduction

In general, robot manipulators are multivariable nonlinear system, and they usually suffer from various uncertainties in their dynamics, such as external disturbance, nonlinear friction, highly time-varying, and payload variation. Therefore, it is difficult to establish exactly mathematical model to achieve the exact trajectory tracking performance for reference inputs and the robustness for the external disturbances of robot manipulators. To overcome these problems, adaptive control strategies have been developed of many researchers, as shown in [1–10], for example.

Recently, the applications of intelligent control such as fuzzy control and neural control to the position control of robotic manipulator have received considerable attention [11–17]. Research on neural networks techniques have been provided online learning algorithms and deal with unmodeled unknown dynamics in robot model. In [18], the multiplayer perception in the adaptive control of feedback, lineariable minimum phase plants represented by an input–output model. In [19], a new method of indirect control trajectory which is generated by the proposed AICT is developed. Using this direct control trajectory (the desired trajectory of forward movement) can indirectly control the tilt angle such that it tracks the desired trajectory asymptotically. In [20], adaptive model reference control and NN-based trajectory planner have been designed on WIP system for dynamic balance and motion tracking of desired

✉ Pham Van Cuong
pvc0610@yahoo.com

[1] College of Electrical and Information Engineering, Hunan University, Changsha, Hunan, China

[2] College of Electrical Technical Technology, Hanoi University of Industry, Hanoi, Vietnam

trajectories by using the LQR optimization technique to minimize both the motion tracking error and the transient acceleration for the best driving comfort. In [21], an adaptive neural network compensator is proposed for control system to improve the control performance without hardware modification. The proposed adaptive NN compensator is very useful when the conventional controller cannot properly handle large disturbances and parameter changes. In [22], an unified framework is presented for identification and control of nonlinear dynamic systems, in which the parametric method of both adaptive nonlinear control and adaptive linear control theory can be applied to perform the stability analysis. The singular perturbation analysis is employed to investigate the stability and robustness properties of dynamical neural network identifier. Various cases that lead to modeling errors are taken into consideration and prove stability and convergence in [23]. In [24], adaptive neural control is presented for a rehabilitation robot with unknown system dynamics to deal with the system uncertainties and improve the robustness. The adaptive NNs are used to approximate the unknown model of the robot and adapt interactions between the robot and the patient. In [25], the neural adaptive control for single-master/multiple-slave teleoperation is proposed to enforce motion coordination of multiple-slave mobile manipulators so as to guarantee the desired trajectories tracking, whereas the internal force tracking error remains bounded and can be made arbitrarily small. In [26], a neural-based control with robust force motion tracking performance for constrained robot manipulators is proposed to deal with the uncertain environmental constrains, disturbances, and unknown robotic dynamics. In [27], a robust neural network output feedback control scheme that includes a novel neural network observer is presented for the motion control of robot manipulator. In [28], the tracking problem has been investigated by using the SBLF and adaptive NNs. It has been proven that the solutions of the closed-loop system under the proposed control which are SGUB and all the closed-loop signals are bounded. In [29], adaptive robust control strategies have been presented systematically to control coordinated multiple mobile manipulators interacting with a rigid surface in the presence of uncertainties and disturbances. The system's motion/force is controlled to converge to the desired manifold and at the same time guarantee the boundedness of the internal force. In [30], an adaptive critic-based NN controller was proposed for the nonlinear pure feedback systems. The critic NN was used to approximate the strategic utility function, whereas an action NN was deployed to minimize both the strategic utility function and the tracking error.

In addition, the radial basis function network is also applied to control dynamic systems. The RBF network is a special architecture of neural network [31]. It is very useful to control the dynamic systems [32–34]. The RBF network adaptation can effectively improve the control performance against large uncertainty of the system. The adaptation law is derived using the Lyapunov method so that the stability of the entire system and the convergence of the weight adaptation are guaranteed. In [9], the adaptive controller using RBF network is proposed to derive elements of inertia matrix, Coriolis matrix, and gravitational force vector for the dynamics of robot manipulators in task space. By designing the control law in the task space, force control can be easily formulated. In [35], the RBFSMC is proposed to eliminate the chattering and control an SMA actuator. The RBF network also used to adjust adaptively the gain of the sliding mode to eliminate the effects of dynamical uncertainties and guarantee asymptotic error convergence in [36]. In [37], the RBF network is presented to compensate adaptively for output tracking of continuous time nonlinear plants. Sliding mode control (SMC) theory which many researchers have successfully studied is also applied to robot manipulators [38–42] because of its robustness. The SMC is designed to reduce the effects of the approximation errors. It generates smooth switching between the adaptive and robust modes from integration of advantages of robust and intelligent control. And the SMC uses again which is large enough to compensate the bounded uncertainties and guarantees stability and passive of nonlinear system. In [40], an ERL approach is introduced to control mechanism in order to control both the chattering and tracking performances, which is impossible to achieve with the conventional SMC approach. In [41], adaptive algorithms which use adaptation laws for tuning both the SMC gain and the thickness of the boundary layer have been proposed to reject a discontinuous control input and to improve the tracking performance. In [42], multiple parameters model-based sliding mode technique is presented to improve the tracking performance for a certain class of nonlinear mechanical system. The introduced method can be used to remove the chattering and infinitely fast switching control problem from the robust control strategy. In this paper, we propose an adaptive RBF network-based NN controller which combines with SMC robust compensator to control problems for two-link robot manipulator which the conventional controllers cannot properly handle large disturbances and parameter changes. The weights of RBFNN are updated online by using Lyapunov stability theory. We also use Lyapunov stability theory to prove that the proposed scheme is not only ensure the stability of the control system, but also has the robustness to the unmodeled dynamics. When compared with the existing results in the literatures, our proposed controller is more flexible, and the time-consuming training process is not necessary. Furthermore, based on simulation and experiment results, the disadvantages which are the chattering phenomenon, tracking errors from the discontinuous control efforts of SMC, are improved.

This paper is organized as follows. The structure of RBF NN is described in Sect. 2. Section 3 introduces dynamic of robot manipulators. Section 4 presents the adaptive neural network controller with robust compensator. Section 5 provides experimental results of two-link robot manipulator. Finally, in Sect. 6, concluding remarks are given.

2 Structure of RBF neural networks

RBF neural networks have shown much attention due to their good generalization ability and a simple network structure which avoids unnecessary and lengthy calculation. The configuration is described in Fig. 1.

Layer 1 The input layer. In this layer, input signals $x = [x_1, x_2, \ldots, x_n]$ are moved directly to the next layer

Layer 2 The hidden layer. This layer consists of an array of computing units which is called hidden nodes. Each neuron of the hidden layer is activated by a radial basis function. The output of hidden layer is calculated as follows:

$$h_j(x) = \exp\left(-\frac{\|x - c_j\|^2}{2d_j^2}\right), \quad j = 1, \ldots, m \qquad (1)$$

where m is the number of hidden nodes, $c_j = [c_{j1}, \ldots, c_{jn}]$ is the center vector of neural net j, d_j notes the standard deviation of the jth radial basis function, $d = [d_1, \ldots, d_m]^{\mathrm{T}}$, and h_j is Gaussian activation function for neural net j.

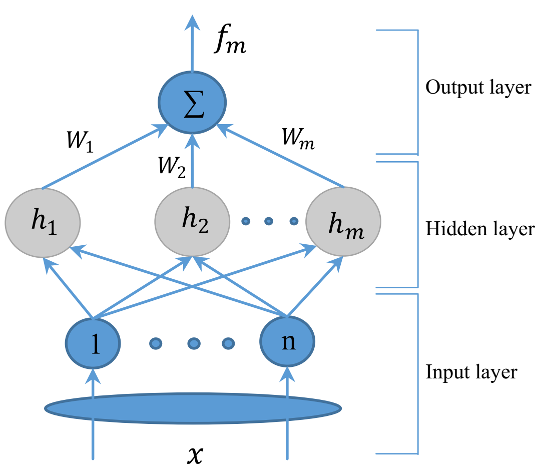

Fig. 1 Structure of RBF network

Layer 3 The output layer. In this layer, the output signal is a linear weighted combination as follows:

$$f_j(x) = \sum_{j=1}^{m} W_{ji} h_j(x, c, d), \quad i = 1, \ldots, n \qquad (2)$$

where W_{ji} is the weight connecting the jth hidden node to the ith output node and n is the number of inputs.

3 Dynamic of robot manipulators

We consider the dynamics of an n-link robot manipulator with external disturbance can be expressed in the Lagrange as follows:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + F(\dot{q}) = \tau - T_d \qquad (3)$$

where $(q, \dot{q}, \ddot{q}) \in R^{n\times 1}$ are the vectors of joint position, velocity, and acceleration, respectively. $M(q) \in R^{n\times n}$ is the symmetric inertial matrix. $C(q, \dot{q}) \in \mathrm{R}^{n\times n}$ is the vector of Coriolis and centripetal forces. $G(q) \in R^{n\times 1}$ expresses the gravity vector. $F(\dot{q})$ represents the vector of the frictions. $T_d \in R^{n\times 1}$ is the unknown disturbances input vector. And $\tau \in R^{n\times 1}$ is the joints torque input vector. For the purpose of designing controller, there are four properties [10] for the dynamics of the robot model (1) as follows.

Property 1 The inertial matrix $M(q)$ is a positive symmetric matrix and bounded:

$$m_1\|x\|^2 \le x^{\mathrm{T}} M(q) x \le m_2\|x\|^2, \quad \forall x \in R^{n\times 1} \qquad (4)$$

with m_1 and m_2 are known positive constants, and they depend on the mass of the robot manipulators.

Property 2 $\dot{M}(q) - 2C(q, \dot{q})$ is skew symmetric matrix, for any vector x:

$$x^{\mathrm{T}}\left[\dot{M}(q) - 2C(q, \dot{q})\right] \quad x = 0 \qquad (5)$$

Property 3 $C(q, \dot{q})\dot{q}$, $G(q)$, and $F(\dot{q})$ are bounded as follows: $C(q, \dot{q})\dot{q} \le C_k \dot{q}^2, G(q) \le G_k,$

$$F(\dot{q}) \le F_k \dot{q} + F_0 \qquad (6)$$

with C_k, G_k, F_k, F_0 are positive constants.

Property 4 $\tau_d > 0; \tau_d \in R^{n\times 1}$ is the unknown disturbance and bounded as:

$$T_d \le \tau_d \qquad (7)$$

4 Design of adaptive neural network controller with robust compensator

In this section, we select the radial basis function network which is combined with sliding mode control robust term function to deal the difficult problems such as the stability and the robustness in NN control systems and the requirement of the model structure in the adaptive control scheme because the RBF network has a simple architecture and mathematically tractable. The detail of this theory is described as follows.

4.1 Adaptive RBFNN with SMC robust compensator

The block diagram of the adaptive radial basis function neural network is shown in Fig. 2. In this scheme, the adaptation laws for updating the weights of the RBF network and the centers and widths of the Gaussian functions are derived to guarantee the stability of control system. And we also use SMC robust term function to guarantee the stability and robustness under the existence of nonlinearities and external disturbances.

The output of the RBF neural network (2) can be rewritten in the following vector as:

$$f(x) = W^{\mathrm{T}} h(x) + \varepsilon \tag{8}$$

where $h = [h_1, h_2, h_3, \ldots, h_m]^{\mathrm{T}}$, W is optimum weight value, and ε is a minimum approximation error vector.

Assumption 1 The approximation error ε is bounded by a positive real constant as:

$$\varepsilon \leq \varepsilon_0 \tag{9}$$

The approximate value of the output RBF is designed as:

$$\hat{f}(x) = \hat{W}^T h(x, c, d) \tag{10}$$

where $\hat{W}^{\mathrm{T}} = [\hat{W}_1^{\mathrm{T}} \hat{W}_2^{\mathrm{T}} \hat{W}_3^{\mathrm{T}} \ldots \hat{W}_m^{\mathrm{T}}]^{\mathrm{T}}$.

From (8) and (10), we have

$$\begin{aligned} \hat{f}(x) &= f(x) - \hat{f}(x) = W^{\mathrm{T}} h(x) + \varepsilon - \hat{W}^{\mathrm{T}} h(x) \\ \hat{f}(x) &= \tilde{W}^{\mathrm{T}} h(x) + \varepsilon \end{aligned} \tag{11}$$

where $\tilde{W} = W - \hat{W}$.

In this section, we consider that by using an approximate adaptive robust law, the system stability is guaranteed and tracking errors is converged to zero when $t \to \infty$. Define a tracking error vector and the sliding mode function as the following equation:

$$e(t) = q_d - q \tag{12}$$

$$s(t) = \dot{e} + \lambda e \tag{13}$$

where $\lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots \lambda_n)$ is a diagonal positive definite matrix. It is typical to define an error metric $s(t)$ to be a performance measure. When the sliding surface $s(t) = 0$, the sliding mode is governed by the following differential equation according to the theory of sliding mode control: $\dot{e} = -\lambda e$. The behavior of the system on the sliding surface is determined by the structure of the matrix λ. When the

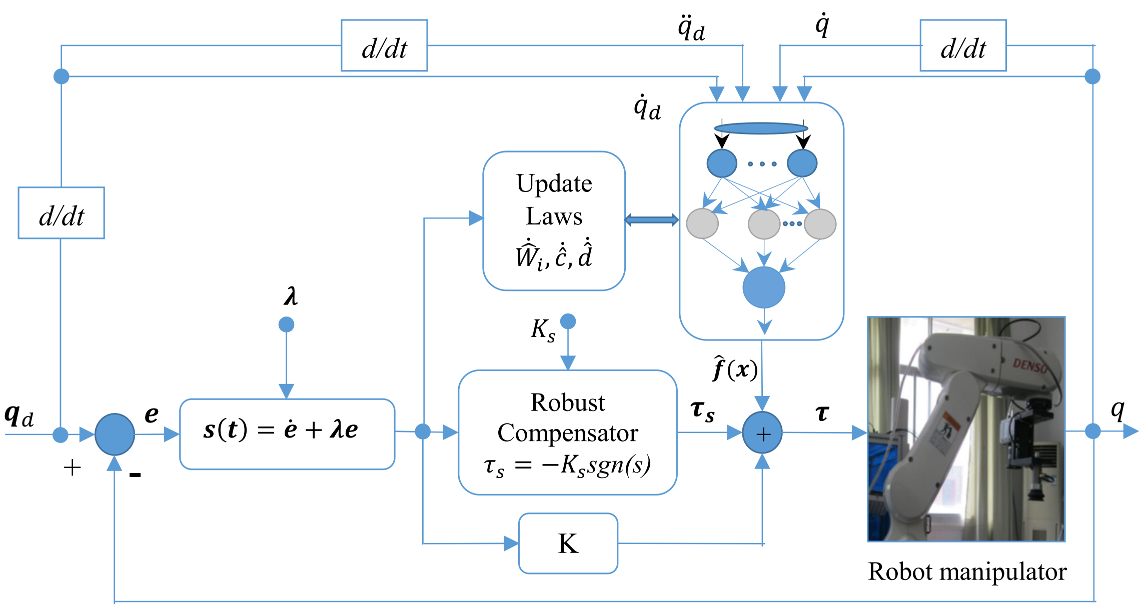

Fig. 2 Architecture of the adaptive RBFNN

error metric $s(t)$ is smaller, the performance is better. Therefore, (3) can be rewritten as:

$$\begin{aligned} M\dot{s} &= M(\ddot{e} + \lambda\dot{e}) = M(\ddot{q}_d + \lambda\dot{e}) - M\ddot{q} \\ M\dot{s} &= f(x) - Cs - \tau + T_d \end{aligned} \tag{14}$$

in which $f(x)$ is defined as follows:

$$f(x) = M(q)(\ddot{q}_d + \lambda\dot{e}) + C(q,\dot{q})(\dot{q}_d + \lambda e) + G(q) + F(\dot{q}) \tag{15}$$

For the dynamics of an n-link robot manipulator (3), the adaptive law is represented as follows:

$$\tau = -\tau_s + Ks + \hat{f}(x) \tag{16}$$

where K is the positive definite matrix and $K = \mathrm{diag}(k_1, k_2, \ldots, k_n)$; $\hat{f}(x)$ is the approximation of the adaptive function $f(x)$. And τ_s is a sliding mode controller robust term that is used to suppress the effects of uncertainties and approximation errors.The robust compensator τ_s is designed by:

$$\tau_s = -K_s \mathrm{sgn}(s) \tag{17}$$

where K_s is selected as: $K_s = (\varepsilon_0 + \tau_d)$

Substituting (16) into (14) and using (11) yields

$$\begin{aligned} M\dot{s} &= f(x) - Cs - (-\tau_s + Ks + \hat{f}(x)) + T_d \\ &= \tilde{f}(x) - (K + C)s + \tau_s + T_d \\ &= -(K + C)s + \tilde{W}^{\mathrm{T}}h(x) + \varepsilon + T_d + \tau_s \end{aligned} \tag{18}$$

By applying the adaptive control law (16) to the dynamic (1), we have, respectively, the estimated values of $M(q)$, $C(q,\dot{q}), G(q)$ and $F(\dot{q})$ in (15) as follows: $\hat{M}(q) = \hat{W}_1^{\mathrm{T}}h_1$; $\hat{C}(q,\dot{q}) = \hat{W}_2^{\mathrm{T}}h_2$; $\hat{G}(q) = \hat{W}_3^{\mathrm{T}}h_3$; $\hat{F}(q,\dot{q}) = \hat{W}_4^{\mathrm{T}}h_4$, and using the sliding mode control robust compensator (17), the on-line RBF adaptive update laws are selected as:

$$\begin{cases} \dot{\hat{W}}_i = -k_i\Phi_{wi}\hat{W}_i s + \Phi_{wi}h_i s^{\mathrm{T}}, \quad i = 1,2,3,4. \\ \dot{\hat{c}} = -\Phi_c \hat{c} s \\ \dot{\hat{d}} = -\Phi_d \hat{d} s \end{cases} \tag{19}$$

whereΦ_w, Φ_c, Φ_d are positive and diagonal constant definite matrices, $\hat{W}^{\mathrm{T}} = \left[\hat{W}_1^{\mathrm{T}}\hat{W}_2^{\mathrm{T}}\hat{W}_3^{\mathrm{T}}\hat{W}_4^{\mathrm{T}}\right]^{\mathrm{T}}; h = [h_1, h_2, h_3, h_4]^{\mathrm{T}}$, and k_i are positive adaptation rates.

4.2 Stability and convergence analysis

The stability of the closed-loop system in Fig. 2 which is described by (16) is established in the following theorem.

Theorem *Consider an n-link robot manipulator represented by (3). If the RBF adaptive update law is designed as (19), and the SMC robust compensator τ_s is given by (17), then the tracking error and the convergence of all the system parameters can be assured and approached to zero*

Figure 2 depicts (16). According to the Lyapunov stability analysis, if the Lyapunov function is positive definite and its derivative is negative semidefinite, then the control system is stable. Therefore, to guarantee the stability of the total control system, we choose the Lyapunov function as follows:

$$V(t) = \frac{1}{2}\left[s^{\mathrm{T}}Ms + \sum_{i=1}^{4} tr\left(\tilde{W}_i^T \Phi_{wi}^{-1}\tilde{W}_i\right) + tr\left(\tilde{c}^T\Phi_c^{-1}\tilde{c}\right) + tr\left(\tilde{d}^T\Phi_d^{-1}\tilde{d}\right)\right] \tag{20}$$

where $\mathrm{tr}(\cdot)$ is a trace operator and $\tilde{W} = W - \hat{W}, \tilde{c} = c - \hat{c}, \tilde{d} = d - \hat{d}$.

Differentiating $V(t)$ along to time, we have:

$$\begin{aligned} \dot{V}(t) &= s^{\mathrm{T}}M\dot{s} + \frac{1}{2}\dot{M}s + \sum_{i=1}^{4}\mathrm{tr}\left(\tilde{W}_i^{\mathrm{T}}\Phi_{wi}^{-1}\tilde{W}_i\right) \\ &\quad - \mathrm{tr}\left(\tilde{c}^{\mathrm{T}}\Phi_c^{-1}\dot{\hat{c}}\right) - \mathrm{tr}\left(\tilde{d}^{\mathrm{T}}\Phi_d^{-1}\dot{\hat{d}}\right) \end{aligned} \tag{21}$$

Substituting (18) into (21), yields

$$\begin{aligned} \dot{V}(t) &= -s^{\mathrm{T}}Ks + \frac{1}{2}s^{\mathrm{T}}(\dot{M} - 2C)s + s^{\mathrm{T}}E + s^{\mathrm{T}}\tau_s \\ &\quad + \sum_{i=1}^{4}\mathrm{tr}\left[\tilde{W}_i^{\mathrm{T}}\left(\Phi_{wi}^{-1}\tilde{W}_i + h_i s^{\mathrm{T}}\right)\right] - \mathrm{tr}\left(\tilde{c}^{\mathrm{T}}\Phi_c^{-1}\dot{\hat{c}}\right) \\ &\quad - \mathrm{tr}\left(\tilde{d}^{\mathrm{T}}\Phi_d^{-1}\dot{\hat{d}}\right) \end{aligned} \tag{22}$$

where $E = (\varepsilon + T_d)$

By using *property 2*, and since $\dot{\tilde{W}} = -\dot{\hat{W}}, s^{\mathrm{T}}\tilde{W}^{\mathrm{T}}h(x)(x) = \mathrm{tr}\left(\tilde{W}^{\mathrm{T}}h(x)s^{\mathrm{T}}\right)$, and from adaptive law (19), (22) becomes:

$$\begin{aligned} \dot{V}(t) &= -s^{\mathrm{T}}Ks + s^{\mathrm{T}}E + s^{\mathrm{T}}\tau_s + \sum_{i=1}^{4}\mathrm{str}\left[\tilde{W}_i^{\mathrm{T}}\left(k_i\hat{W}_i\right)\right] \\ &\quad + \mathrm{str}\left(\tilde{c}^{\mathrm{T}}\hat{c}\right) + \mathrm{str}\left(\tilde{d}^{\mathrm{T}}\hat{d}\right) \\ \dot{V}(t) &= -s^{\mathrm{T}}Ks + s^{\mathrm{T}}E - s^{\mathrm{T}}K_s\mathrm{sign}(s) \\ &\quad + \sum_{i=1}^{4}\mathrm{str}\left[k_i\tilde{W}_i^{\mathrm{T}}(W_i - \tilde{W}_i)\right] + \mathrm{str}\left[\tilde{c}^{\mathrm{T}}(c - \tilde{c})\right] \\ &\quad + \mathrm{str}\left[\tilde{d}^{\mathrm{T}}(d - \tilde{d})\right] \\ \dot{V}(t) &= -s^{\mathrm{T}}Ks + s^{\mathrm{T}}(\varepsilon + T_d) - s(\varepsilon_0 + \tau_d) \\ &\quad + s\left\{\sum_{i=1}^{4} tr[k_i\tilde{W}_i^{\mathrm{T}}\left(W_i - \tilde{W}_i\right)] + \mathrm{tr}\left[\tilde{c}^{\mathrm{T}}(c - \tilde{c})\right]\right. \\ &\quad \left. + \mathrm{tr}\left[\tilde{d}^{\mathrm{T}}(d - \tilde{d})\right]\right\} \end{aligned} \tag{23}$$

From *Property 4* and *assumption 1*, (23) becomes:

$$\begin{aligned} \dot{V}(t) &\leq -s^{\mathrm{T}}Ks + s\left\{\sum_{i=1}^{4} tr[k_i\tilde{W}_i^{\mathrm{T}}\left(W_i - \tilde{W}_i\right)]\right. \\ &\quad \left. + \mathrm{tr}\left[\tilde{c}^{\mathrm{T}}(c - \tilde{c})\right] + \mathrm{tr}\left[\tilde{d}^{\mathrm{T}}(d - \tilde{d})\right]\right\} \end{aligned} \tag{24}$$

Since $tr[\tilde{x}^{\mathrm{T}}(x-\tilde{x})] \le \tilde{x}_F x_F - \tilde{x}_F^2$, and $x_F \le x_{\max}$, we have:

$$\begin{cases} \mathrm{tr}\left[\tilde{W}^{\mathrm{T}}\left(W-\tilde{W}\right)\right] \le \tilde{W}_F W_F - \tilde{W}_F^2 \le \tilde{W}_F(W_{\max}-\tilde{W}_F) \\ \mathrm{tr}[\tilde{c}^{\mathrm{T}}(c-\tilde{c})] \le \tilde{c}_F c_F - \tilde{c}_F^2 \le \tilde{c}_F(c_{\max}-\tilde{c}_F) \\ \mathrm{tr}\left[\tilde{d}^{\mathrm{T}}\left(d-\tilde{d}\right)\right] \le \tilde{d}_F d_F - \tilde{d}_F^2 \le \tilde{d}_F\left(d_{\max}-\tilde{d}_F\right) \end{cases}$$

Now, (24) becomes:

$$\begin{aligned} \dot{V}(t) \le & -s^{\mathrm{T}}Ks + s\left\{\sum_{i=1}^{4} k_i \tilde{W}_{iF}\left(W_{i\max}-\tilde{W}_{iF}\right)\right. \\ & \left. + \tilde{c}_F(c_{\max}-\tilde{c}_F) + \tilde{d}_F(d_{\max}-\tilde{d}_F)\right\} \\ & \le -s\left(Ks + \sum_{i=1}^{4} k_i\left[\left(\tilde{W}_{iF} - \frac{W_{i\max}}{2}\right)^2 - \frac{W_{i\max}^2}{4}\right]\right) \\ & + \left[\left(\tilde{c}_F - \frac{c_{\max}}{2}\right)^2 - \frac{c_{\max}^2}{4}\right] + \left[\left(\tilde{d}_F - \frac{d_{\max}}{2}\right)^2 - \frac{d_{\max}^2}{4}\right] \end{aligned} \tag{25}$$

In (25), if gain K and s are selected to satisfy the following inequality:

$$K \ge \frac{1}{s}\left[\sum_{i=1}^{4} \frac{k_i W_{i\max}^2}{4} + \frac{c_{\max}^2}{4} + \frac{d_{\max}^2}{4}\right] \quad i = 1,\ldots,4. \tag{26}$$

Then

$$\dot{V}(t) \le 0 \tag{27}$$

Hence, if all parameters of the control system are bounded with $t > 0$, and all initial conditions are bounded at initial $t = 0$, $0 \le V(0) \le \infty$ is ensured. According to Lyapunov's direct method, this demonstrates the uniformly ultimately bounded of s. By selecting large control gains K, the tracking error can be achieved as small as desired. Moreover, the neural network weight errors are bounded by $W_{i\max}$, the know bound on the ideal weights W_i, and the neural network reconstruction error ε_0 and the bounded disturbances τ_d increase the bounds on s and $\tilde{W}_{iF}$. Therefore, the neural network weights can be initialized at zero, so tracking and stability are guaranteed. This means that there is no off-line learning phase, but neural network occurs in real time.

5 Simulation and experimental results

5.1 Simulation results

In this section, a two-link robot manipulator is utilized to verify the effectiveness of the proposed control scheme and a series of simulation researches is carried out.

We consider the two-link robot manipulator model that is shown in Fig. 3, and the dynamic equation can be described by using Lagrange method.

$$M(q)\ddot{q} + C(q,\dot{q}) + G(q) + F(\dot{q}) = \tau - T_d$$

where

$$\begin{aligned} M(q) &= \begin{bmatrix} M_{11}(q_2) & M_{12}(q_2) \\ M_{21}(q_2) & M_{22}(q_2) \end{bmatrix} \\ &= \begin{bmatrix} (m_1+m_2)l_1^2 + m_2 l_2^2 + 2m_2 l_1 l_2 \cos(q_2) & m_2 l_2^2 + m_2 l_1 l_2 \cos(q_2) \\ m_2 l_2^2 + m_2 l_1 l_2 \cos(q_2) & m_2 l_2^2 \end{bmatrix} \\ C(q,\dot{q}) &= \begin{bmatrix} C_{11}(q_2) & C_{12}(q_2) \\ C_{21}(q_2) & C_{22}(q_2) \end{bmatrix} \\ &= \begin{bmatrix} -m_2 l_1 l_2 \sin(q_2)\dot{q}_2 & -m_2 l_1 l_2 \sin(q_2)(\dot{q}_1+\dot{q}_2) \\ m_2 l_1 l_2 \sin(q_2)\dot{q}_1 & 0 \end{bmatrix} \\ G(q) &= \begin{bmatrix} G_1(q_1,q_2) \\ G_2(q_1,q_2) \end{bmatrix} \\ &= \begin{bmatrix} (m_1+m_2)l_1 g\cos(q_2) + m_2 l_2 g\cos(q_1+q_2) \\ m_2 l_2 g\cos(q_1+q_2) \end{bmatrix} \\ F(\dot{q}) &= \begin{bmatrix} F_1(\dot{q}_1) \\ F_2(\dot{q}_2) \end{bmatrix} = \begin{bmatrix} 0.5\,\mathrm{sign}(\dot{q}_1) \\ 0.5\,\mathrm{sign}(\dot{q}_2) \end{bmatrix}; \tau = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}; \\ T_{1d} &= T_{2d} = 0.5\sin(t); \end{aligned}$$

in which m_1 and m_2 are the mass of joint 1 and joint 2, respectively, l_1 and l_2 are the length of joint 1 and joint 2, respectively, and g is acceleration of gravity.The parameters of two-link robot manipulator are given as:

$$m_1 = 3(\mathrm{kg});\ m_2 = 2(\mathrm{kg});$$
$$l_1 = 0.8(\mathrm{m});\ l_2 = 1(\mathrm{m});$$
$$g = 10(\mathrm{m/s^2}).$$

The parameter values used in the adaptive control system are:

$$\lambda = \mathrm{diag}[5,5];\ K = \mathrm{diag}[20,20];\ \Phi_w = \mathrm{diag}[15,15];$$

The control objective is to control the joint angles of a two-link robot manipulator to follow the desired trajectories. The desired position trajectories of two-link robot manipulators are chosen by: $q_{1d} = q_{2d} = 0.5\sin(t)$ (in radians), and initial

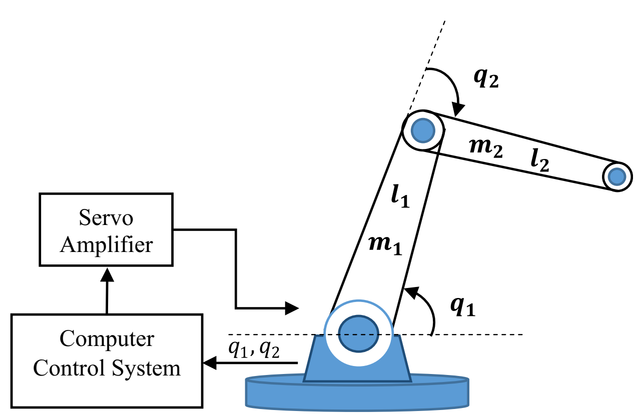

Fig. 3 Two-link robot manipulator control system

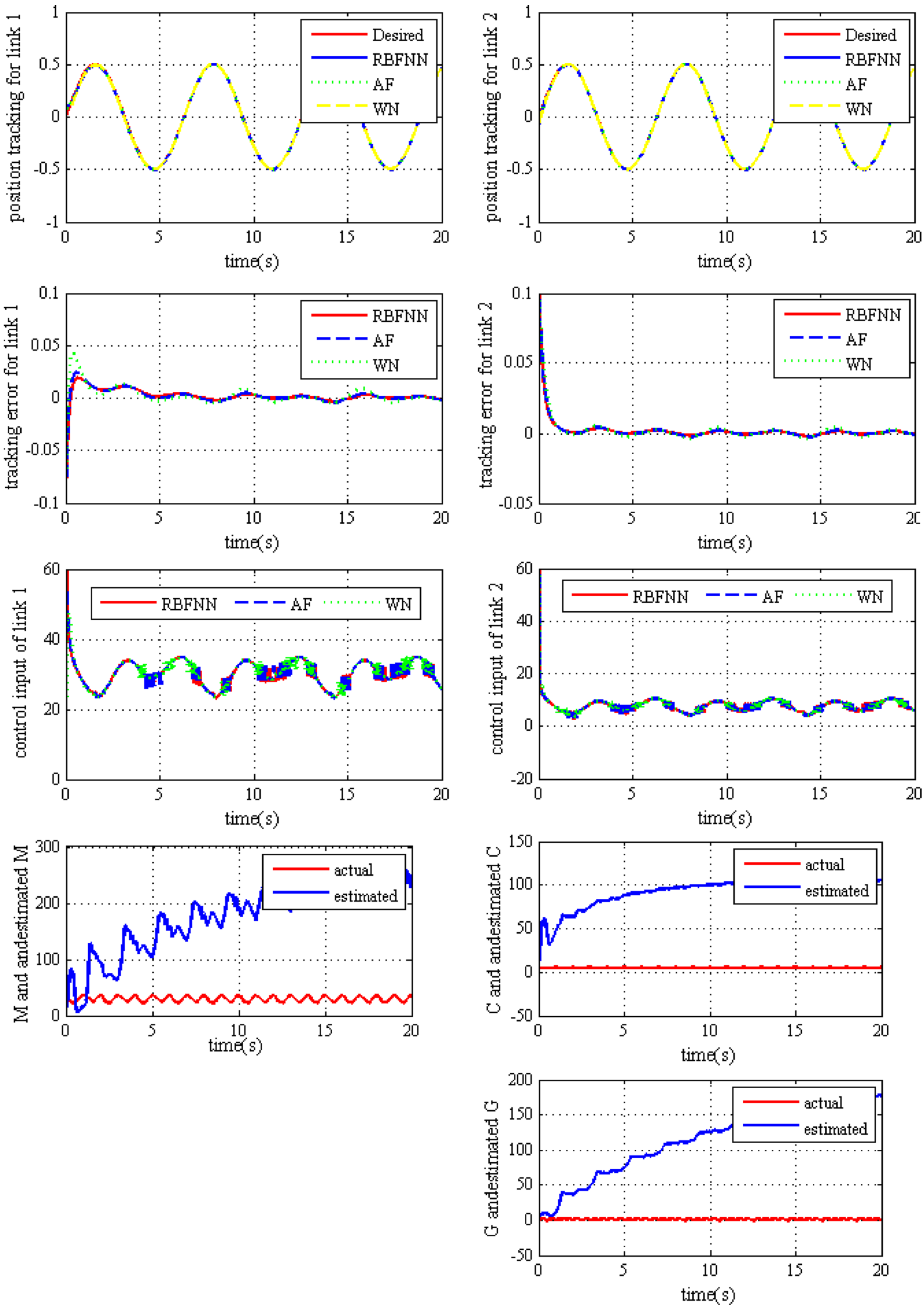

Fig. 4 Simulated position responses, tracking errors, and control efforts of the proposed system, AF, WN control, and estimated norms of M, C, G

positions of joints are $q_0 = [0.1 \quad -0.1]^{\mathrm{T}}$, and initial velocities of joints are $\dot{q}_0 = [0.0 \quad 0.0]^{\mathrm{T}}$. Let $e = q_d - q = [e_1 \quad e_2]^{\mathrm{T}}$ be the tracking errors.

In the following passage, our proposed control scheme is applied to a two-link robot manipulator in comparison with the adaptive fuzzy [1] and the wavelet network [5] control. The simulation results of joint position responses, tracking errors, control torques in following the desired trajectories for joint 1 and joint 2, and the estimations $\hat{M}(q)$, $\hat{G}(q)$ and $\hat{C}(q,\dot{q})$ of $M(q), G(q)$ and $C(q,\dot{q})$, respectively, are shown in Fig. 4. From these results, we can find that the proposed adaptive scheme has faster reduction rate in tracking errors than both the AF and WN control system. Moreover, we can observe that when the tracking errors reach the big value, there is little chattering in torque. This is resulted from robust compensator. Furthermore, Fig. 4 also shows that both the estimations $\hat{M}(q), \hat{G}(q)$ and $\hat{C}(q,\dot{q})$ do not converge to $M(q), G(q)$ and $C(q,\dot{q})$, respectively. This means that the desired trajectory is not exciting persistently, which happens often in real application. Therefore, the use of proposed scheme with adaptation weights can effectively improve the performance of the closed-loop system compared with the existing results. It seems that the robust tracking performance of the proposed control scheme is more excellent and effective than the AF and WN control.

5.2 Experimental results

In this section, a two-link robot manipulator in our laboratory for intelligent automation technology is applied to verify the effectiveness of the proposed control scheme. The experimental control system model is shown in Fig. 5. It consists of an IBM PC with Pentium microprocessor, an encoder board to acquire the angles of the joint 1 and joint 2, and a A/D module to send commend signals to the servo amplifier. The proposed control algorithm is implemented using MATLAB Simulink.

In this study, two different experimental cases are adopted to investigate the applicability and the performance of the proposed technique under various environments as the parameter variation and the change of the external disturbance.

(a) The first experimental case

In this case, we add more 1(kg) on the link 2 of robot with the same desired trajectories and the others parameters as in the simulation case. Figure 6 shows the experimental results of joint position responses, tracking errors, control torques, and the estimations $\hat{M}(q), \hat{G}(q)$ and $\hat{C}(q,\dot{q})$, respectively, of $M(q), G(q)$ and $C(q,\dot{q})$. From these results, it is easy to see that the responses and the tracking error norm

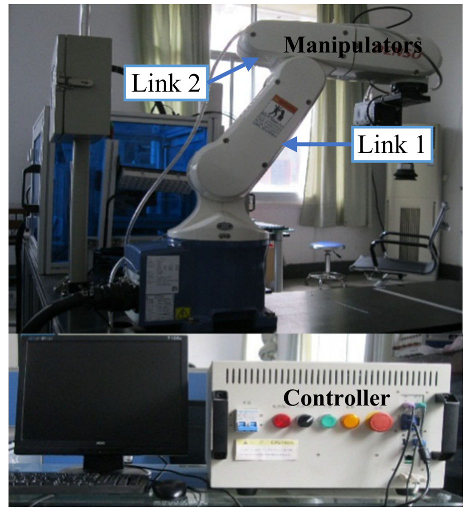

Fig. 5 Experimental control system

of the proposed control scheme are quite better than both AF and WN control method. Moreover, Fig. 6 implies that our control torques are less and smooth than the methods in [1, 5]. Furthermore, Fig. 6 also shows that both the estimations $\hat{M}(q), \hat{G}(q)$ and $\hat{C}(q,\dot{q})$ do not converge to $M(q), G(q)$ and $C(q,\dot{q})$, respectively. This means that the desired trajectory is not exciting persistently, which happens often in real application. Therefore, the robust tracking performance of our proposed control scheme is better than the AF and WN control in [1, 5], respectively, under parameters variation.

(b) The second experimental case

In this case, we assume that the robot is tracking a trajectory and suddenly the external disturbance is injected into control system. This happened after the first 5 s of the experimental time, and all other parameters are chosen as in the simulation case. The shapes of the external disturbance are expressed as follows: $d(t) = [20\sin(10t) \quad 20\sin(10t)]^{\mathrm{T}}$.

The experimental responses of joint position, tracking error, control torque, and the estimations $\hat{M}(q), \hat{G}(q)$ and $\hat{C}(q,\dot{q})$, respectively, of $M(q), G(q)$ and $C(q,\dot{q})$ are shown in Fig. 7. From this experimental, we can find that the control performance and robustness of the proposed controller under external disturbance are better than AF and WN controller. The performance of our proposed approach is slightly affected more than AF approach, while the performance of WN approach is seriously affected. Furthermore, Fig. 7 also shows that both estimations $\hat{M}(q), \hat{G}(q)$ and $\hat{C}(q,\dot{q})$ do not converge to $M(q), G(q)$ and $C(q,\dot{q})$, respectively. This means that the desired trajectory is not exciting persistently, which happens often in real application.

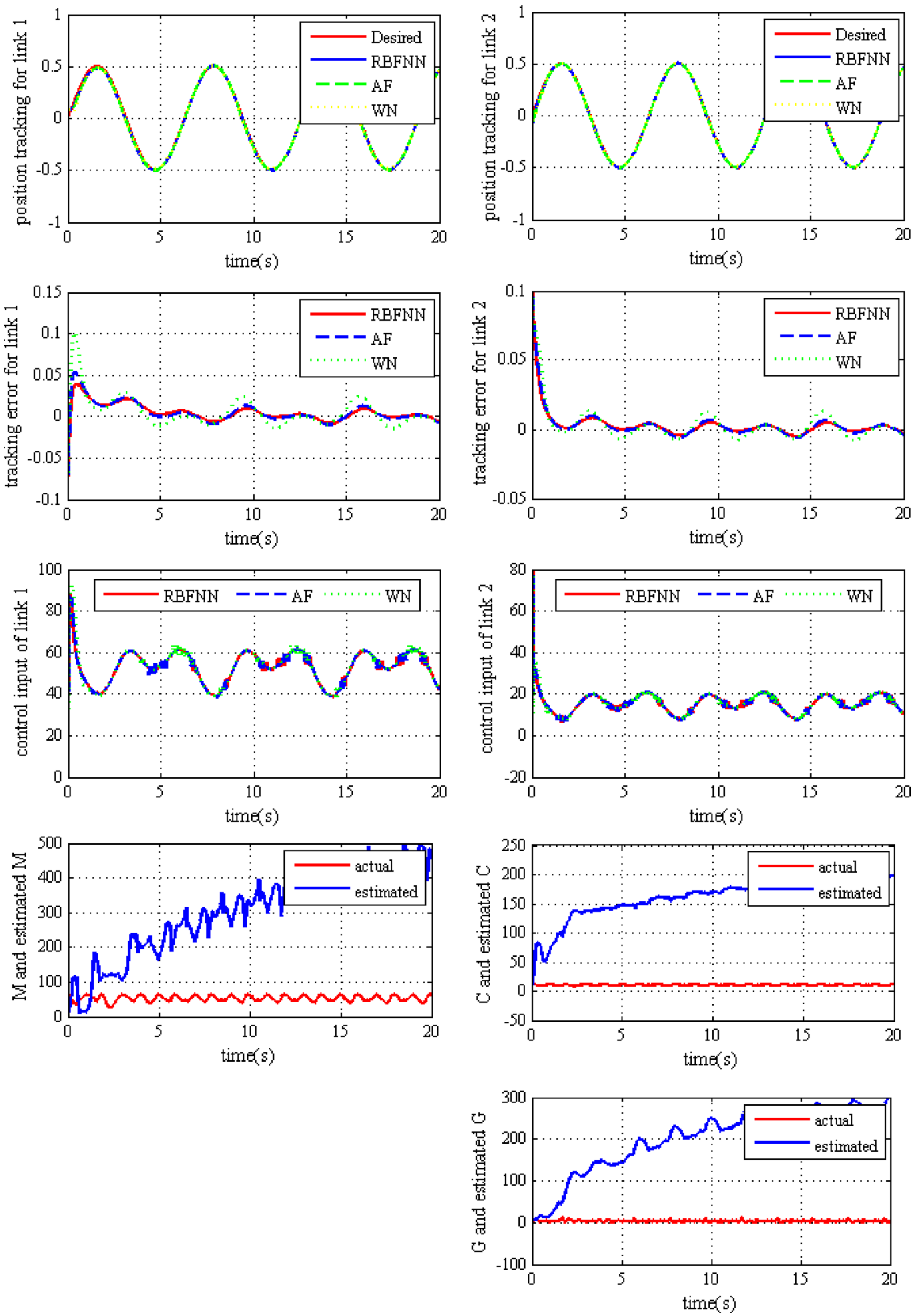

Fig. 6 The experimental results of position responses, tracking errors, control efforts, and estimated norms of M, C, G for the first case

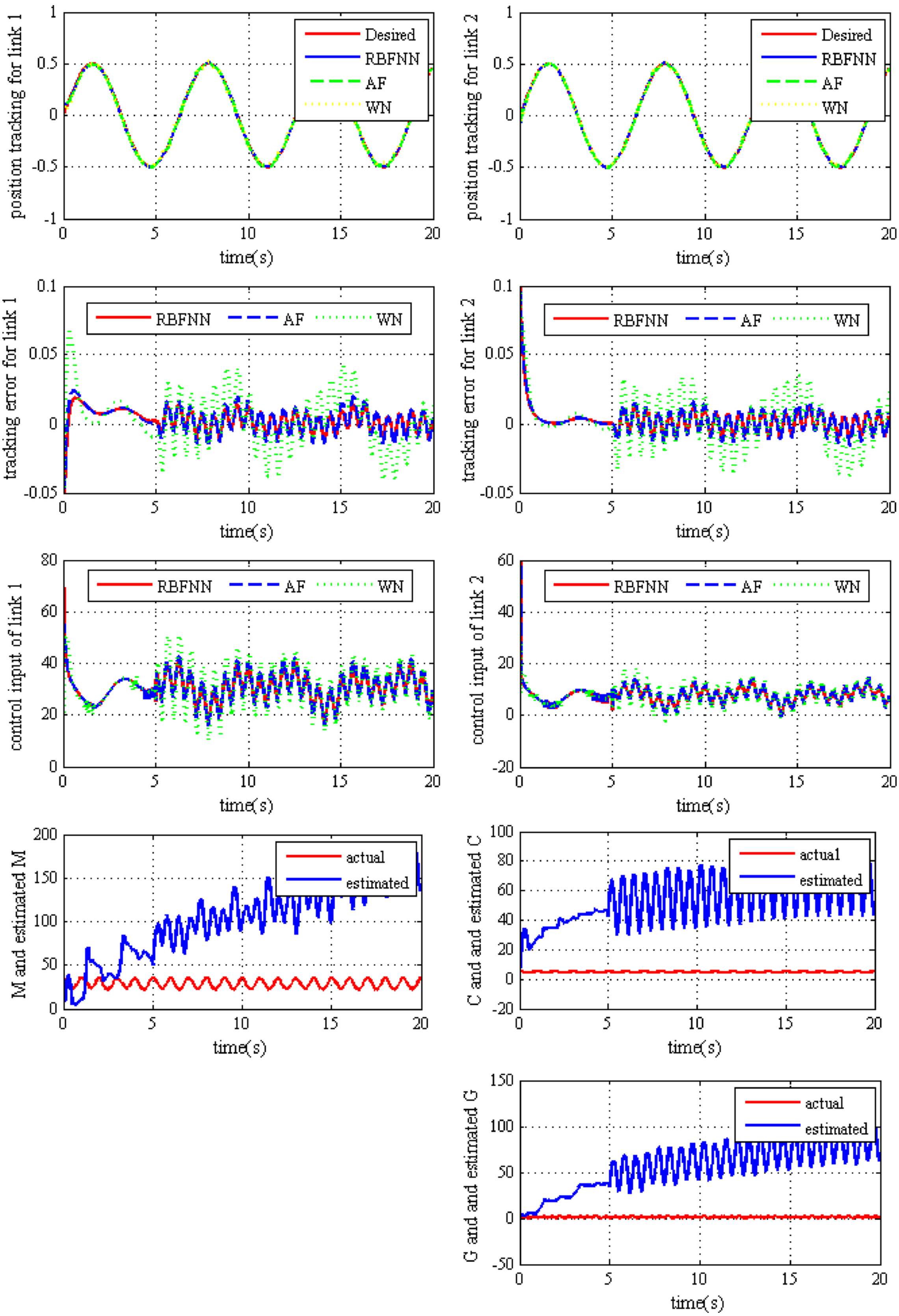

Fig. 7 The experimental results of position responses, tracking errors, control efforts, and estimated norms of M, C, G for the second case

6 Conclusion

In this paper, we have successfully demonstrated the development and application of an adaptive radial basis function neural network control system for the trajectory tracking to control the joints position of two-link robot manipulator. The structure of the proposed system incorporates the advantages of radial basis function neural network and sliding mode robust term function. The radial basis function network generates control input signals, and the adaptive control law is derived to guarantee the stability of the control system based on the Lyapunov method which is often used in the conventional adaptive control method. In this scheme, the robust compensator acts as an auxiliary controller to guarantee the stability and robustness of the control system under the existence of disturbance, mass variation, and modeling errors. Finally, based on the simulation and experimental results, the proposed adaptive control scheme is shown to work well in the trajectory tracking control of two-link robot manipulator and can also be applied to other system, such as AC servo system, mobile robotic, and so on.

Acknowledgments This work was supported by National Natural Science Foundation of China (Grant No. 61175075) National High-tech Research and Development Projects (Grant Nos. 2012AA 112312, 2012AA11004). The authors would like to thank the associate editor and the reviewers for their constructive comments, which greatly improved the quality for this work.

References

1. Sun FC, Sun ZQ, Feng G (1999) An adaptive fuzzy controller based on sliding mode for robot manipulators. IEEE Trans Syst Man Cybern B 29(5):661–667
2. Zou Yi, Wang Yaonan, Liu XinZhi (2010) Neural network robust H_∞ tracking control strategy for robot manipulators. Appl Math Model 34(7):1823–1838
3. Topalov V, Cascella GL, Giordano V, Cupertino F, Kaynak O (2007) Sliding mode neural-adaptive control for electrical drives. IEEE Trans Ind Electron 54(1):671–679
4. Omatu S, Khalid M, Yusof R (1996) Neuro-control and its applications: advances in industrial control. Springer, Berlin
5. Wang FF, Zhu SQ, Liu SG (2009) Robust adaptive wavelet network control for robot manipulators. IEEE Glob Congr Intell Syst 2:313–317
6. Lee CH, Theng CC (2000) Identification and control of dynamic systems using recurrent fuzzy neural network. IEEE Trans Fuzzy Syst 8(4):349–366
7. Alavandar S, Nigam MJ (2008) Neuro-fuzzy based approach for inverse kinematics solution of industrial robot manipulators. Int J Comput Commun 3(3):224–234
8. Li ZJ, Chen WD (2008) Adaptive neural-fuzzy control of uncertain constrained multiple coordinated nonholomic mobile manipulators. Eng Appl Artif Intell 21(7):985–1000
9. Ge SS, Hang CC, Woon LC (1997) Adaptive neural network control of robot manipulator in task space. IEEE Trans Ind Electron 44(6):746–752
10. Lewis FL, Dowson DM, Abdallah CT (2004) Robot manipulator control theory and practice. Marcel Dekker, New York
11. Ngo T, Wang Y, Mai TL, Nguyen MH, Chen J (2012) Robust adaptive neural-fuzzy network tracking control for robot manipulator. Int J Control 7(2):341–352
12. Wai RJ, Chen PC (2006) Robust neural-fuzzy-network control for robot manipulator including actuator dynamics. IEEE Trans Ind Electron 53(4):1328–1349
13. Luo J, Li ZJ, Ming A, Ge SS (2006) Robust motion/force control of nonholonomic mobile manipulators using hybrid joints. IEEE Intell Control Autom. doi:10.1109/WCICA.2006.1712348
14. Song Z, Yi J, Zhao D, Li X (2005) A computed torque controller for uncertain robotic manipulator systems: fuzzy approach. Fuzzy Sets Syst 154:208–226
15. Juang CF (2002) A TSK-type recurrent fuzzy network for dynamic system processing by neural network and genetic algorithm. IEEE Trans Fuzzy Syst 10(2):155–170
16. Li ZJ, Yang CG, Tang Y (2012) Decentralised adaptive fuzzy control of coordinated multiple mobile manipulators interacting with non-rigid environments. IET Control Theory Appl 7(3):397–410
17. He W, Ge SS, How BE, Choo YS (2014) Dynamics and control of mechanical systems in offshore engineering. Springer, London
18. Chen FC, Khail HK (1992) Adaptive control of nonlinear systems using neural networks. Int J Control 55(6):1299–1317
19. Yang CG, Li ZJ, Cui RX, Xu BG (2014) Neural network based motion control of under-actuated wheeled inverted pendulum models. IEEE Trans Neural Netw Learn Syst 25(11):2004–2016
20. Yang CG, Li ZJ, Li J (2013) Trajectory planning and optimized adaptive control for a class of wheeled inverted pendulum vehicle models. IEEE Trans Cybern 43(1):24–36
21. Choi YK, Lee MJ, Kim S, Kay YC (2001) Design and implementation of an adaptive neural-network compensator for control system. IEEE Trans Ind Electron 48(2):416–423
22. Polycarpou MM (1996) Stable adaptive neural control scheme for nonlinear systems. IEEE Trans Autom Control 41(3):447–451
23. Rovithakis GA, Christodoulou MA (1994) Adaptive control of unknown plants using dynamical neural networks. IEEE Trans Syst Man Cybern 24(3):400–412
24. He W, Ge SS, Li Y, Chew E, Ng YS (2014) Neural network control of a rehabilitation robot by state and output feedback. J Intell Robot Syst. doi:10.1007/s10846-014-0150-6
25. Li ZJ, Su CY (2013) Neural-adaptive control of single-master multiple slaves teleoperation for coordinated multiple mobile manipulators with time-varying communication delays and input uncertainty. IEEE Trans Neural Netw Learn Syst 24(9):1400–1413
26. Li Z, Yang C, Gu J (2007) Neuro-adaptive compliant force/motion control for uncertain constrained wheeled mobile manipulator. Int J Robot Autom 22(3):206–214
27. Kim YH, Lewis FL (1999) Neural network output feedback control of robot manipulators. IEEE Trans Robot Autom 15(2):301–309
28. Zhao Z, He W, Ge SS (2014) Adaptive neural network control for a fully actuated marine surface vessel with multiple output constraints. IEEE Trans Control Syst Technol 22(4):1536–1543
29. Li ZJ, Li JX, Kang Y (2010) Adaptive Robust coordinated control of multiple mobile manipulators interacting with rigid environments. Automatica 46:2028–2034
30. Xu B, Yang CG, Shi ZK (2014) Reinforcement learning output feedback NN control using deterministic learning techniques. IEEE Trans Neural Netw Learn Syst 25(3):635–641
31. Haykin S (1994) Neural networks: a comprehensive foundation. Macmillan College Publishing Company, New York
32. Zheng Rong Yang (2006) A novel radial basis function neural network for discriminant analysis. IEEE Trans Neural Networks 17(3):604–612

33. Li Y, Qiang S, Zhuang X, Kaynak O (2004) Robust and adaptive back-stepping control for nonlinear systems using RBF neural networks. IEEE Trans Neural Networks 15(3):693–701
34. Patino HD, Liu D (2000) Neural network-based model reference adaptive control system. IEEE Trans Syst Man Cybern B 30(1):198–204
35. Tai NT, Ahn KK (2010) A RBF neural network sliding mode controller for SMA actuator. Int J Control Autom Syst 6(6):1296–1305
36. Zhihong M, Wu HR, Palaniswami M (1998) An adaptive tracking controller using neural networks for a class of nonlinear systems. IEEE Trans Neural Networks 9(5):947–955
37. Seshagiri S, Khail HK (2000) Output feedback control of non-linear systems using RBF neural networks. IEEE Trans Neural Networks 11(1):69–79
38. Veluvolu KC, Soh YC (2009) High-gain observers with sliding mode for state and unknown input estimation. IEEE Trans Ind Electron 56(9):3386–3393
39. Cao JB, Cao BG (2009) Neural network sliding mode control based on online identification for electric vehicle with ultraca-pacitor-battery hybrid power. Int J Control Autom Syst 7(3):409–418
40. Fallaha CJ, Saad M, Kanaan HY, Al-Haddad K (2011) Sliding-mode robot control with exponential reaching law. IEEE Trans Industr Electron 58(2):600–610
41. Park JS, Han GS, Ahn HS, Kim DH (2001) Adaptive approaches on the sliding mode control of robot manipulators. Trans Control Autom Syst Eng 3(1):15–20
42. Islam S, Liu PX (2011) Robust sliding mode control for robot manipulator. IEEE Trans Ind Electron 58(6):2444–2453