DESIGN ADAPTIVE TRAJECTORY TRACKING CONTROLLER FOR ROBOT MANIPULATORS BASED ON NEURAL NETWORK

THIẾT KẾ BỘ ĐIỀU KHIỂN BÁM QUỸ ĐẠO CHO TAY MÁY ROBOT SỬ DỤNG MẠNG NƠ RON

Pham Van Cuong[1,*], Hoang Van Huy[1], Nguyen Duy Minh[2]

ABSTRACT

This paper proposes a trajectory tracking controller based on adaptive neural networks (ANNs) for robot manipulators (RMs) to achieve the high precision position tracking performance. In this controller, adaptive radial basis function (RBF) neural networks control is investigated to control the joints position and approximate the unknown dynamics of an n-link robot manipulators. The adaptive RBF network can effectively improve the control performance against large uncertainty of the system. The online adaptive control training laws are determined by Lyapunov stability and the approximation theory, so that uniformly stable adaptation is guaranteed, and asymptotically tracking is achieved. In adition, a robust control is constructed as an auxiliary controller to suppress the neural network modeling errors and the bounded disturbances to guarantee the stability and robustness under various environments such as the mass variation, the external disturbances and modeling uncertainties. Finally, simulation examples are given to illustrate the effectiveness of the proposed approach control system for two link-robot manipulators. From simulation results, we can find that the proposed adaptive control has fast reduction rate in tracking errors and tracking errors is converged to zero when $t \rightarrow \infty$. Moreover, when the tracking errors reach the big value, there is little chattering in torque.

Keywords: *Robot manipulators, tracking control, neural networks.*

TÓM TẮT

Bài báo này đề cập đến một bộ điều khiển bám quỹ đạo sử dụng mạng nơ ron thích nghi để đạt được hiệu suất bám vị trí chính xác cao cho tay máy robot. Trong bộ điều khiển này, mạng nơ ron RBF được sử dụng để điều khiển vị trí các khớp và xấp xỉ tham số bất định của tay máy robot. Mạng RBF có khả năng cải tiến hiệu suất điều khiển chống lại tính bất định của hệ thống. Các luật dạy thích nghi được đánh giá bằng lý thuyết xấp xỉ và ổn định Lyapunov, vì vậy đảm bảo hệ thống được ổn định và bám tiệm cận. Ngoài ra, bộ điều khiển bền vững cũng được đưa ra đóng vai trò như bộ điều khiển bù để khử các sai lệch của mạng nơ ron và nhiễu để đảm bảo tính ổn định và bền vững dưới sự thay đổi của môi trường như tải thay đổi, nhiễu hay sự bất định của cấu trúc. Cuối cùng, các ví dụ mô phỏng được thực hiện để chứng minh tính hiệu quả của hệ thống điều khiển cho tay máy robot 2 bậc tự do. Kết quả mô phỏng cho thấy tốc độ bám nhanh và sai lệch tiến đến 0 khi thời gian tiến ∞. Hơn nữa, khi sai lệch là lớn nhất, mô men điều khiển cũng ít bị dao động.

Từ khóa: *Tay máy robot, điều khiển bám, mạng nơ ron.*

[1]Faculty of Electrical Engineering, Hanoi University of Industry
[2]Faculty of Electrical Engineering, Electric Power University
[*]Email: cuongpv0610@haui.edu.vn
Received: 15/01/2021
Revised: 18/6/2021
Accepted: 15/11/2021

1. INTRODUCTION

Robot manipulators are multivariable nonlinear systems and they suffer from various uncertainties in their dynamics, which deteriorate the system performance and stability, such as external disturbance, nonlinear friction, highly time-varying, and payload variation. Therefore, achieving high performance in trajectory tracking is a very challenging task. To overcome these problems, various control methods have been proposed, including adaptive control, intelligent control, sliding mode control and variable structure control [1-4]. In recent year, the intelligent control based on neural networks (NNs) has been developed fast. With online learning ability of neural network, it has many advantages in the control systems. In [5], the neural adaptive control was proposed for sigle master multiple slave teleoperation to enforce motion coordination of mobile manipulators so as to guarantee the desired trajectories tracking whereas the the tracking error remains bounded. In [6], the proposed MPC approach can handle a formulated QP problem using a neurodynamic optimination approach. The applied neural networks can make the formulated constrained QP converging to the exact optimal values. In [7], authors introduced a general framework for the formation tracking control of all types of wheeled mobile robots (WMRs). In this study, a radial basis function neural network and an adaptive robust compensator were incorporated in the formation controller design to improve the tracking performance in the presence of uncertain non-linearities and unknown parameters. In [8], an adaptive neural network controller was proposed for a 3-DOF robotic manipulator that is subject to backlash-like hysteresis and friction. Two neural networks are used to approximate the dynamics and the hysteresis non-linearity. A neural network, which utilises a radial basis function approximated the robot dynamics. The other neural network, which employs a hyperbolic tangent activation function, was used to approximate the unknown backlash-like hysteresis. The proposed controllers ensured the boundedness of the control signals. In [10], two controllers: a neural controller and the most commonly used PID controller have been implemented to control a nonlinear system. These controllers were evaluated for the quality of work and compared the influence of a given regulators on the control of a nonlinear system in terms of selected criteria. In [11], a robust adaptive tracking controller is developed for robot manipulators with uncertain dynamics using radial basis function neural network. In this controller, the improved radial basis function neural network was chosen to approximate the uncertain dynamics of robot manipulators and learn the upper bound of the uncertainty, and the adaptive law was used to solve the uniform final bounded problem of the radial basis function neural network weights, therefore, the stability was guaranted.

In this paper, an adaptive trajectory tracking neural networks controller is proposed for two-link robot manipulators which the conventional controllers can not properly handle large disturbances and parameter changes. This proposed controller can control not only the position of the end-effector, but also the force exerted by the end-effector on the object. By designing the control law in task space, force control can be easily formulated.

This paper is organized as follows: Section 2, the dynamic of robot manipulators is presented; Section 3 discribes structure of RBF Neural networks; Section 4 presents design of adaptive tracking neural networks; Section 5 provides simulation results of two-link robot manipulators; Finally, in section 6, concluding remarks are given.

2. DYNAMIC OF ROBOT MANIPULATORS

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

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

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 positive inertial Matrix. $C(q,\dot{q})\in R^{n\times n}$ is the vector of Coriolis and Centripetal Matrix. $G(q)\in R^{n\times 1}$ expresses the Gravity force vector. $\tau\in R^{n\times 1}$ is the joints torque input vector.

Relative to the end-effector of the manipulators, the task specification is given. Denote $x\in R^{n}$ are the end-effector position and orientation in the task space. The task space dynamics can be written as follow:

$$M_x(q)\ddot{x}+C_x(q,\dot{q})\dot{x}+G_x(q)=\tau_x \tag{2}$$

where

$$\begin{cases} M_x(q)=\left[\{\Theta\}^T\{\Xi(q)\}\right]+\delta_D(q) \\ C_x(q,\dot{q})=\left[\{A\}^T\{Z(z)\}\right]+\delta_C(z) \\ G_x(q)=\left[\{B\}^T\{H(q)\}\right]+\delta_G(q) \end{cases} \tag{3}$$

With $\{\Theta\},\{\Xi(q)\},\{A\},\{Z(z)\},\{B\},\{H(q)\}$ are matrices with their elements being $\theta_{kj},\xi_{kj}(q),\alpha_{kj},\zeta_{kj},\beta_{kj},\eta_k(q)$, respectively.

It is observed that both $M_x(q)$ and $G_x(q)$ are only functions of q. Whereas, for $C_x(q,\dot{q})$, a dynamic neural network of q and $\dot{q}$ is needed to model it. Assume that, $m_{xkj}(q), g_{xk}(q)$ and $c_{xkj}(q,\dot{q})$ can be modeled as follows:

$$\begin{cases} m_{xkj}(q)=\theta_{kj}^T\xi_{kj}(q)+\varepsilon_{mkj} \\ c_{xkj}(q,\dot{q})=\alpha_{kj}^T\zeta_{kj}(z)+\varepsilon_{ckj} \\ g_{xk}(q)=\beta_k^T\eta_k(q)+\varepsilon_{gk} \end{cases} \tag{4}$$

where $\theta_{kjl},\beta_{kl},\alpha_{kjl}$ are the weight of the neural networks, $\xi_{kjl},\eta_{kl},\zeta_{kjl}$ are the corresponding Gaussian basic functions, $\varepsilon_{mkj},\varepsilon_{ckj},\varepsilon_{gk}$ are the modeling errors of $m_{xkj}(q), c_{xkj}(q,\dot{q}), g_{xkj}\,q)$, respectively, and $z=[q^T\dot{q}^T]^T\in R^{2n}$.

For the purpose of designing controller, there are two properties for the dynamics of the robot model (2) as follows.

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

Property 2: $\dot{M}_x(q)-2C_x(q,\dot{q})$ is skew symmetric matrix.

3. DESIGN OF ADAPTIVE TRACKING NEURAL NETWORKS

3.1. 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 Figure 1 [9].

Layer 1: The input layer. In this layer, input signals $x=[x_1,x_2,...,x_n]$ 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 are activated by a radial basis function. The output of hidden layer is calculated as follows:

$$h_j(x)=\exp\left(-\frac{\|x-c_j\|^2}{2d_j^2}\right),\quad j=1,\ldots,m \tag{5}$$

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

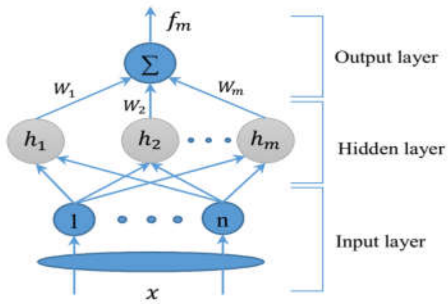

Figure 1. Structure of RBF network

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

$$f_j(x)=\sum\nolimits_{j=1}^{m} W_{ji}h_j(x,c,d) \qquad i=1,\dots,n \tag{6}$$

where W_{ji} is the weight connecting the *j*-th hidden node to the *i*-th output node, and n is the number of inputs.

The output of the RBF neural network (6) can be rewritten in the following vector as:

$$f(x)=W^T h(x) \tag{7}$$

where $h=[h_1,h_1,\dots,h_m]^T$, W is optimum weight value.

The approximate value of the output RBF is designed as:

$$\hat f(x)=\hat W^T h(x,c,d) \tag{8}$$

where $\hat W^T=[\hat W_1^T\ \hat W_2^T\ \dots\ \hat W_m^T]$

From (7) and (8), we have

$$\tilde f(x)=f(x)-\hat f(x)=W^T h(x)-\hat W^T h(x)$$

$$\tilde f(x)=\tilde W^T h(x) \tag{9}$$

where $\tilde W=W-\hat W$

3.2. Design controller

In this section, we consider that by using proposing an approximate adaptive robust law, the system stability is guaranteed and tracking errors is converged to zero when $t\to\infty$. The block diagram of the adaptive Radial Basis Function neural network is shown in Figure 2.

Let $x_d(t);\dot x_d(t);\ddot x_d(t)$ are the desired trajectory, velocity and acceleration, respectively.

Define a tracking error vector and the sliding mode function as the following equations:

$$\begin{cases} e(t)=x_d(t)-x(t)\\ \dot x_r(t)=\dot x_d(t)+\lambda e(t)\\ r(t)=\dot e(t)+\lambda e(t)\end{cases} \tag{10}$$

where, $\lambda=\mathrm{diag}(\lambda_1,\lambda_2,\dots,\lambda_n)$ is a diagonal positive definite matrix.

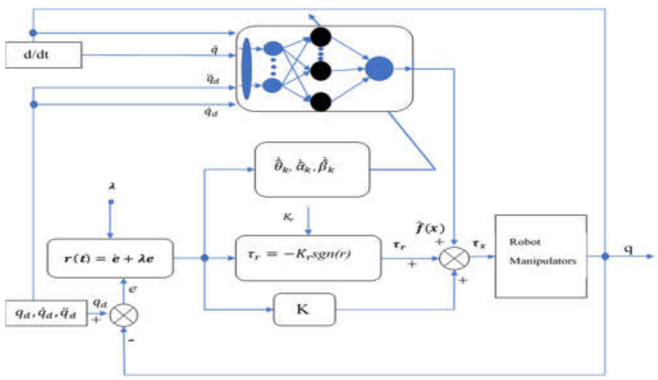

Figure 2. Architecture of the adaptive RBFNN

From control system in Figure 2, the adaptive control law is proposed as:

$$\tau_x=\hat f(x)+\tau_r+Kr=\hat f(x)+k_s\mathrm{sign}(r)+Kr \tag{11}$$

where $K=\mathrm{diag}(k_1,k_2,\dots,k_n)$ is the positive definite matrix; $\hat f(x)$ is the approximation of the adaptive function f(x) and is defined as $\hat f(x)=\hat M_x(q)\ddot x_r+\hat C_x(q,\dot q)\dot x_r+\hat G_x(q)$; $\tau_r=k_s\mathrm{sign}(r)$ is a sliding mode controller robust term, and $k_s>\|\delta\|$, with $\delta=\delta_M(q)\ddot x_r+\delta_C(q)\dot x_r+\delta_G(q)$.

Substituting (11) into (2) and using (10), yields:

$$\tilde M_x(q)\ddot x_r+\tilde C_x(q,\dot q)\dot x_r+\tilde G_x(q)+\delta=M_x(q)\dot r+C_x(q,\dot q)r+k_s\mathrm{sign}(r)+Kr \tag{12}$$

By applying the adaptive control law (11) to the dynamic (2), the online RBF neural networks parameters adaptive update laws are designed as:

$$\begin{cases}\dot{\hat\theta}_k=\Gamma_k\xi_k(q)\ddot x_r r_k\\ \dot{\hat\alpha}_k=Q_k\zeta_k(z)\dot x_r r_k\\ \dot{\hat\beta}_k=N_k\eta_k(q)r_k\end{cases} \tag{13}$$

where $\hat\theta_k$ and $\hat\alpha_k$ are the column vectors with their elements being $\hat\theta_{kj}$ and $\hat\alpha_{kj}$, respectively, then $\hat\theta_k,\hat\alpha_k,\hat\beta_k\in L_\infty$, e and $\dot e\to 0$ when $t\to\infty$; Γ_k,Q_k,N_k are dimentional compatible symmetric positive-definite matrics, and Γ_k,Q_k are defined as follows:

$$\Gamma_k=(\Gamma_k^T>0)=\begin{bmatrix}\Gamma_{k1} & 0 & . & . & . & 0\\ 0 & \Gamma_{k2} & . & . & . & 0\\ . & . & . & . & . & .\\ . & . & . & . & . & .\\ . & . & . & . & . & .\\ 0 & 0 & . & . & . & \Gamma_{kn}\end{bmatrix};$$

$$Q_k=(Q_k^T>0)=\begin{bmatrix}Q_{k1} & 0 & . & . & . & 0\\ 0 & Q_{k2} & . & . & . & 0\\ . & . & . & . & . & .\\ . & . & . & . & . & .\\ . & . & . & . & . & .\\ 0 & 0 & . & . & . & Q_{kn}\end{bmatrix} \tag{14}$$

with Γ_{kj},Q_{kj} are multudimentional compatible matrix block, and $1\le j\le n$.

To guarantee the stability of the control system, the Lyapunov function is chosen as follows:

$$L=\frac{1}{2}r^T M_x(q)r+\frac{1}{2}\sum_{k=1}^{n}\tilde\theta_k^T\Gamma_k^{-1}\tilde\theta_k+\frac{1}{2}\sum_{k=1}^{n}\tilde\alpha_k^T Q_k^{-1}\tilde\alpha_k+\frac{1}{2}\sum_{k=1}^{n}\tilde\beta_k^T N_k^{-1}\tilde\beta_k \tag{15}$$

Defferentiating L along to time, the following equation can be obtained as:

$$\dot L=r^T(M_x(q)\dot r+C_x(q,\dot q)r+\sum_{k=1}^{n}\tilde\theta_k^T\Gamma_k^{-1}\dot{\tilde\theta}_k +\sum_{k=1}^{n}\tilde\alpha_k^T Q_k^{-1}\dot{\tilde\alpha}_k+\sum_{k=1}^{n}\tilde\beta_k^T N_k^{-1}\dot{\tilde\beta}_k \tag{16}$$

Substituting equation (12) and using equation (17) as follows:

$$\begin{cases} r^T\tilde{M}_x\ddot{x}_r = \sum_{k=1}^{n}\{\tilde{\theta}_k\}^T\{\xi_k(q)\}\ddot{x}_r r_k \\ r^T\tilde{C}_x\ddot{x}_r = \sum_{k=1}^{n}\{\tilde{\alpha}_k\}^T\{\zeta_k(z)\}\dot{x}_r r_k \\ r^T\tilde{G}_x = \sum_{k=1}^{n}\tilde{\beta}_k^T\eta_k(q)r_k \end{cases} \tag{17}$$

Equation (16) becomes:

$$\begin{aligned} \dot{L} = &-r^Tkr - k_s r^T \operatorname{sgn}(r) \\ &+\sum_{k=1}^{n}\left[\{\tilde{\theta}_k\}^T\{\xi_k(q)\}\ddot{x}_r r_k\right] + \sum_{k=1}^{n}\left[\{\tilde{\alpha}_k\}^T\{\zeta_k(q)\}\dot{x}_r r_k\right] \\ &+\sum_{k=1}^{n}\left[\tilde{\beta}_k^T\eta_k(q)r_k\right] + \sum_{k=1}^{n}\left[\tilde{\theta}_k^T\Gamma_k^{-1}\dot{\tilde{\theta}}_k\right] + \sum_{k=1}^{n}\left[\tilde{\alpha}_k^T Q_k^{-1}\dot{\tilde{\alpha}}_k\right] \\ &+\sum_{k=1}^{n}\left[\tilde{\beta}_k^T N_k^{-1}\dot{\tilde{\beta}}_k\right] + r^T\delta \end{aligned} \tag{18}$$

Applying the parameters adaptive update laws (13) into (18), and considering $k_s > \|\delta\|$, the derivative $\dot{L}$ can be bounded

$$\dot{L} \le -r^T K r \le 0 \tag{19}$$

If all parameters of the adaptive control system are bounded with $t > 0$, and all initial conditions are bounded at $t = 0$, $0 \le V(0) \le \infty$ is ensured. Furthermore, integrating $\dot{V}(t)$ with respect to time as follows:

$$\int_0^\infty \dot{L}(t)dt \le -\int_0^\infty r^T K r dt \tag{20}$$

Equation (20) can be rewritten as:

$$\int_0^\infty r^T K r dt \le -\int_0^\infty \dot{L}(t)dt = L(0) - L(\infty) \tag{21}$$

Because L(0) is a bounded function, and L(t) is non-increasing and bounded, we have

$$\lim_{t\to\infty}\int_0^t r^T K r dt < \infty \tag{22}$$

According to Barbalat's Lemma, it can be shown that $\lim_{t\to\infty}\int_0^t r^T K r dt = 0$. Since V(0) and K are positive constants, it follows that $r \in L_2^\infty$. From equation (19), it follows that $\dot{L}(t) \in L_\infty$. Therefore, we can conclude that, both the global stability of the system and the tracking errors are guaranteed and converged to zero when $t \to \infty$ by the adapting control law (13). The proof is completed.

4. SIMULATION RESULTS

In this section, for illustrative purposes, a two-link robot manipulator is employed to verify the effectiveness of the proposed control scheme [9].

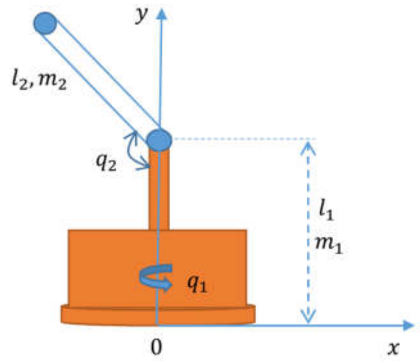

Figure 3. Two link robot manipulators

Consider the two-link robot manipulators model that is shown in Figure 3, and the dynamic equation can be described by using Lagrang method as in equation (1).

where

$$M(q) = \begin{bmatrix} M_{11}(q_2) & M_{12}(q_2) \\ M_{21}(q_2) & M_{22}(q_2) \end{bmatrix};$$

$$C(q,\dot{q}) = \begin{bmatrix} C_{11}(q_2) & C_{12}(q_2) \\ C_{21}(q_2) & C_{22}(q_2) \end{bmatrix}; G(q) = \begin{bmatrix} G_1(q_1,q_2) \\ G_2(q_1,q_2) \end{bmatrix};$$

$M_{11}(q_2) = (m_1 + m_2)l_1^2 + m_2 l_2^2 + 2m_2 l_1 l_2 \cos(q_2)$;

$M_{12}(q_2) = M_{21}(q_2) = m_2 l_2^2 + m_2 l_1 l_2 \cos(q_2)$;

$M_{22}(q_2) = m_2 l_2^2$; $C_{11}(q_2) = -m_2 l_1 l_2 \sin(q_2)\dot{q}_2$;

$C_{12}(q_2) = -m_2 l_1 l_2 \sin(q_2)(\dot{q}_1 + \dot{q}_2)$;

$C_{21}(q_2) = m_2 l_1 l_2 \sin(q_2)\dot{q}_1$; $C_{22}(q_2) = 0$;

$G_1(q_1,q_2) = (m_1 + m_2)l_1 g\cos(q_2) + m_2 l_2 \cos(q_1 + q_2)$;

$G_2(q_1,q_2) = m_2 l_2 \cos(q_1 + q_2)$

in which, m_1 and m_2 are the mass of joint 1 and joint 2, respectivly; l_1 and l_2 are the length of joint 1 and joint 2, respectively; g is acceleration of gravity.

The parameters of two link robot manipulator are given as: $m_1 = 2(kg)$ and $m_2 = 1(kg)$; $l_1 = l_2 = 1(m)$; $g = 10(m/s^2)$

The desired position trajectory of two link robot manipulators is chosen by:

$$x_{1d}(t) = 1.0 + 0.1\cos(2\pi t); x_{2d}(t) = 1.0 + 0.1\sin(2\pi t)$$

In the following passage, this proposed control scheme is applied to a two link robot manipulators. The simulation results of joint position responses, velocity tracking, control input, and tracking trajectory in following the desired tracking trajectories for joint 1 and joint 2, respectively, are shown in Figure 4. From these results, we can find that the proposed adaptive control has fast reduction rate in tracking errors and position tracking times are fast (0.1s for joint 1 and 0.05s for joint 2). Moreover, we can observe that when the tracking errors reach the big value, there is little chattering in torque, the maximum of control torques are 250Nm and -200Nm for link 1 and link 2, respectively. This means that the desired trajectory is not exciting persistently, which happens often in real application. Therefore, the use of proposed scheme 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 excellent and effective.

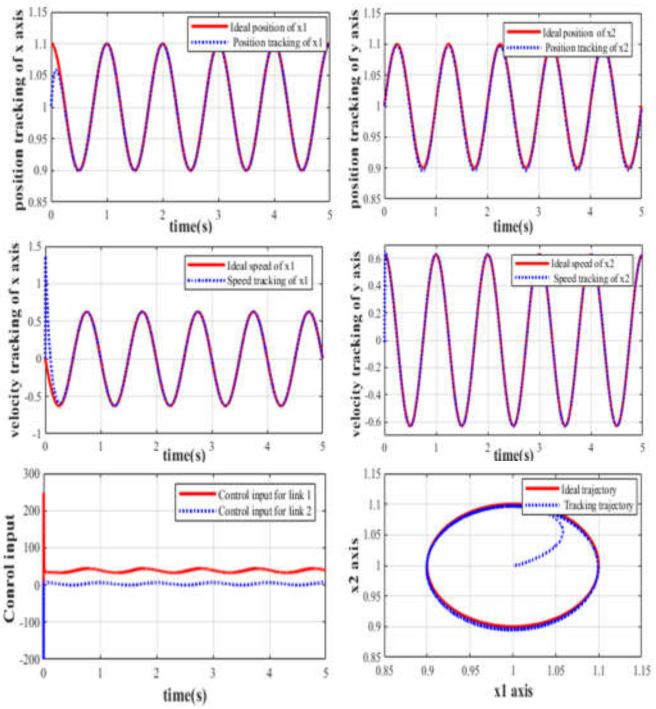

Figure 4. Simulated result position responses, velocity tracking, control input, and tracking trajectory of the proposed system

5. CONCLUSION

In this paper, adaptive control of robot manipulators has been studied. It has been shown that, if Gaussian radial basis function networks that generates control input signals are used, and the adaptive control law is derivered to guarantee the stability of the control system base on the Lyapunov method, uniformly stable adaptation is assused, and asymptotically tracking is achieved. In adition, the proposed controller can be easily modified to achieve robustness to nework modeling errors and bounded disturbance. Finally, base on the simulation results, the proposed adaptive control scheme has demonstrated the performance in the trajectory tracking control of two-link Robot Manipulator.

REFERENCES

[1]. Li Z. J., Li J. X., Kang Yu, 2010. *Adaptive robust coordinated control of multiple mobile manipulators interacting with rigid environments.* Automatica, 46, 2010, pp 2028-2034.

[2]. Sun W., Wang YN., 2005. *A robust robotic tracking controller based on neural network.* Int. Journal of Robotic and Automation. 20 (3), pp 199–204.

[3]. Elshafei A., Karray F., 2005. *Variable structure based fuzzy logic Identification of a class of nonlinear systems.* IEEE Trans. Contr. Sys. Tech, 13(4), pp 646-653.

[4]. Li Z. J., Yang C. G., Tang Y., 2013. *Decentralized adaptive Fuzzy control of Coordinated multiple mobile manipulators interacting with nonrigid environments.* IET Control Theory and Application. 7(3), pp 397-410.

[5]. Li Zhijun, Chun Yisu, 2013. *Neural adaptive control of single master multiple slaves teleoperation for coordinated multiple mobile manipulators with time varying communication delays and input uncertainties.* IEEE Trans. Neural network and learning system. 24(9, pp 1400-1413.

[6]. Li ZJ., Xia YQ., Su CY., Deng J., Fu J., He W., 2014. *Missile Guidance Law Based on Robust Model Predictive Control Using Neural Network Optimization*. IEEE Transactions on Neural Networks and Learning Systems. 1(99),1-8.

[7]. K. Shojaei, 2017. *Neural adaptive output feedback formation control of type (m, s) wheeled mobile robots*. IET Control theory & Applications, 11(4), pp.504-515.

[8]. He W., Amoateng D. O., Yang C., Gong D. I., 2017. *Adaptive neural network control of a robotic manipulator with unknown backlash - like hysteresis.* IET Control theory & Applications, 11(4), pp.567-575.

[9]. Pham Van Cuong, Wang Yao Nan, 2015. *Adaptive trajectory tracking neural network control with robust compensator for robot manipulators.* Neural Computing and Applications 27 (2), 525–536.

[10]. Agata Nawrocka, Marcin Nawrocki, Andrzej Kot, 2019. *Neural network control for Robot manipulator.* International Carpathian Control Conference, DOI: 10.1109/CarpathianCC.2019.8765995.

[11]. Donghui Wang, Shijie Zhang, 2020. *Improved neural network-based adaptive tracking control for manipulators with uncertain dynamics.* International Journal of Advanced Robotic Systems, DOI: 10.1177/1729881420947562.

THÔNG TIN TÁC GIẢ

Phạm Văn Cường[1], Hoàng Văn Huy[1], Nguyễn Duy Minh[2]

[1]Khoa Điện, Trường Đại học Công nghiệp Hà Nội

[2]Khoa Kỹ thuật điện, Trường Đại học Điện lực