JST: Smart Systems and Devices
Volume 31, Issue 1, May 2021, 050-058

Adaptive Control for Dual-Arm Robotic System Based on Radial Basis Function Neural Network

***Luu Thi Hue*[2], *Nguyen Pham Thuc Anh*[1*]**
[1]*Hanoi University of Science and Technology, Hanoi, Vietnam*
[2]*Electrical Power University, Hanoi, Vietnam*
[*]*Email: anh.nguyenphamthuc@hust.edu.vn*

Abstract

The paper has developed an adaptive control using neural network for controlling a dual-arm robotic system in moving a rectangle object to the desired trajectories. Firstly, the overall dynamics of the manipulators and the object have been derived based on Euler-Lagrangian principle. And then based on the dynamics, a controller has been proposed to achieve the desired trajectories of the grasping object. A radial basis function neural network has been applied to compensate uncertainties of dynamic parameters. The adaptive algorithm has been derived owning to the Lyapunov stability principle to guarantee asymptotical convergence of the closed dynamic system. Finally, simulation work on MatLab has been carried out to reconfirm the accuracy and the effectiveness of the proposed controller.

Keywords: Adaptive control, dynamics, Radial Basis Function Neural Network, dual-arm robotic.

1. Introduction

Researches on multi-manipulator control have received increasing attention due to their advantages over single-manipulator in industrial applications such as assembling, transporting heavy objects, etc. The development of multi-manipulator systems creates an opportunity to replace humans in dangerous environments. However, control of multi-manipulator systems is always a challenge due to the high nonlinearity and complication of their dynamics. A coordination scheme for cooperative manipulation with two-arm systems was introduced by Yun and Kumar and a nonlinear-feedback control algorithm has proposed [1]. A robust algorithm for cooperative control of closed-chain manipulators has been proposed with uncertain dynamics [2]. Adaptive control of multiple-robot manipulation in a dynamical environment has been proposed [3]. An adaptive controller combined with a sliding mode controller has been introduced [4]. A robust adaptive hybrid force/position control scheme for two planar-manipulators coordinating to move an object without knowing its parameters, but knowning the parameters of the robots has been proposed [5]. A robust adaptive algorithm has been proposed for controlling dual planar-manipulators in cooperative manipulation of an object under uncertainty of dynamic parameters [6]. Almost the aforementioned controllers were based on inverse dynamics to create adaptive update algorithms, then they were complicated in mathematic formulae and the practical applicability was limited.

ISSN: 2734-9373
https://doi.org/10.51316/jst.150.ssad.2021.31.1.7
Received: 28 April 2020; accepted: 02 October 2020

Nowadays neural networks (NNs) have created drastic changes in the development of controllers, especially in robotics. The ability of NNs to approximate nonlinear uncertainties leads to the idea of using a neural network directly in a model-based control strategy. The idea traced here is based on the possibility of training networks to compensate the dynamic parameters of multi-manipulators. A dual neural network has been used to control the coordination of two redundant robots in real-time [7]. An adaptive hybrid force/position has been developed for cooperative multiple-manipulators carrying and manipulating a common rigid object by Panwar et al [8]. A framework for NNs based consensus control has been proposed for multiple robotic manipulators under leader-follower communication topology. Two situations: fixed and switching communication topologies, were studied by using adaptive and robust control principles [9]. A synchronized NN approach has been proposed for controlling multiple robotic manipulators based on the leader-follower network communication topology [10]. An adaptive robust control (SOSMC) algorithm has been considered for dual-arm manipulators using the combination of second-order sliding mode control and neural networks. The SOSMC deals with the system robustness when faced with external disturbances and parametric uncertainties [11]. The problem of self-tuning control with a two-manipulator system holding a rigid object in the presence of inaccurate translational base frame parameters is addressed. An adaptive robust neural controller is proposed to cope with inaccurate translational base frame parameters, internal force, modeling uncertainties, joint friction, and external disturbances. A radial basis function neural network is adopted for all kinds of dynamical estimation, including undesired internal force. Specialized robust compensation is established for global stability [12]. NN-based adaptive controllers have proved the effectiveness in compensating dynamic uncertainties. However, they are based on inverse dynamics and complicated in mathematical formulae.

The aforementioned adaptive NN-based controllers are effective in compensation of uncertainties of dynamic parameters of systems. However, they are based on inverse dynamics of the systems, then the proposed controllers are complicated and do not ensure for keeping contact between end-effectors and the object.

In the paper, we propose Radial Basis Function Neural Network (RBFNN) in controlling a dual-arm robotic system manipulating a rectangle object. The rest of the paper is organized as follows: section 2 addresses to formulating dynamics of dual-arm robot and object system, section 3 aims to build up adaptive RBFNN-based controller, section 4 introduces simulation results and section 5 is for conclusions.

2. Formulation of Dynamics of Overall Dual Robot-Object System

2.1. System Description

The model under study consists of a dual-arm robotic system in grasping a rigid object and is depicted in Fig.1. The dual-arm robotic system has two 3-DOF planar robots in antagonistic arrangement. The left robot is numbered the first and the right one is the second. The object is rigid and rectangle, then its surfaces are flat. Coordinated frames, main parameters and variables of the objective system have been defined in detail in the previous work [13]. The dual-robot system is responsible for stable grasping the object and then manipulating it dexterously. It is assumed that the end-effectors and the object are rigid and point-contacted. When the end-effector i contacts with the corresponding surface, there exists a force f_i that occurs perpendicular and a force λ_i that occurs tangential to the contact surface. The whole system works in the vertical plane, then it is affected by gravity.

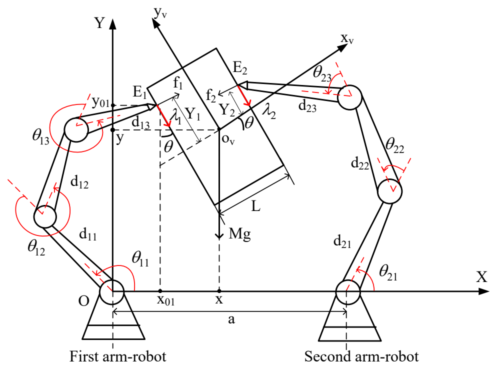


Fig. 1. Description of the dual-arm robotic and object system

2.2 Kinematic Relations

Refer to the Fig.1, the position $o_v(x,y)$ of the mass center of the object in the reference frame {OXY} can be calculated as:

$$x = x_{0i} - (-1)^i \frac{L}{2}\cos\theta + Y_i \sin\theta;$$
$$y = y_{0i} - (-1)^i \frac{L}{2}\sin\theta - Y_i \cos\theta, \qquad (1)$$

where $E_i(x_{0i}, y_{0i})$ is the position of the end-effector i in the frame {OXY}.

Differentiating (1)

$$\dot{x} = \dot{x}_{0i} + (-1)^i \frac{L}{2}\sin\theta.\dot{\theta} + Y_i \cos\theta.\dot{\theta}$$
$$\dot{y} = \dot{y}_{0i} - (-1)^i \frac{L}{2}\cos\theta.\dot{\theta} + Y_i \sin\theta.\dot{\theta}.$$

then

$$\begin{bmatrix}\cos\theta & \sin\theta & -Y_i\end{bmatrix}\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{bmatrix} = \begin{bmatrix}\cos\theta & \sin\theta\end{bmatrix} J_{0i}.\dot{q}_i\,, \qquad (1a)$$

where J_{0i} is the Jacobian matrix.

$$J_{0i} = \left(\frac{\partial x_{0i}}{\partial q_i}, \frac{\partial y_{0i}}{\partial q_i}\right)^T$$

It is possible to express (1a) into

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{bmatrix} = D^+.\begin{bmatrix}\cos\theta & \sin\theta\end{bmatrix}.J_{0i}.\dot{q}_i$$

with $D = \begin{bmatrix}\cos\theta & \sin\theta & -Y_i\end{bmatrix}$ and $D^+ = D^T(D.D^T)^{-1}$ is pseudo-inverse matrix of D matrix.

Define vector $z = \begin{bmatrix}x & y & \theta\end{bmatrix}^T$ then

$$\dot{z} = A_i.\dot{q}_i;\ \ddot{z} = A_i.\ddot{q} + \dot{A}_i\dot{q}\,,$$

That means

$$\dot{q} = A.\dot{z};\ \ddot{q} = A.\ddot{z} + B\dot{z}\ , \qquad (2)$$

where

$$A = [(A_1^{-1})^T, (A_2^{-1})^T]^T,$$
$$B = [(\text{-}A_1^{-1}\dot{A}_1A_1^{-1})^T, (-A_2^{-1}\dot{A}_2A_2^{-1})^T]^T.$$

2.3. Dynamics

The dynamics of the system has been formulated based on the Euler-Lagrange principle. The Lagrangian function is defined as:

$$L = K - P$$

where K is the kinematic energy and P is the potential energy.

The dynamic equation of the whole system can be expressed as:

a) For the i^{th} robot

$$\begin{aligned}\tau_i &= H_i(q_i)\ddot{q}_i + C_i(q_i,\dot{q}_i) + G_i(q_i)\\ &- (-1)^i J_{0i}^T \begin{bmatrix}\cos\theta\\ \sin\theta\end{bmatrix} f_i + J_{0i}^T \begin{bmatrix}\sin\theta\\ -\cos\theta\end{bmatrix}\lambda_i,\end{aligned} \tag{3}$$

where q_i is joint angle vector, $H_i(q_i)$ is inertia matrix, $C_i(q_i,\dot{q}_i)$ is Coriolis and centrifugal matrix, and $G_i(q_i)$ is gravity vector, for i=1,2.

b) For the object:

$$H_z\,\ddot{z} + \sum_{i=1}^{2}(-1)^i f_i \begin{bmatrix}\cos\theta\\ \sin\theta\\ -Y_i\end{bmatrix} - \sum_{i=1}^{2}\lambda_i \begin{bmatrix}\sin\theta\\ -\cos\theta\\ -(-1)^i.\dfrac{L}{2}\end{bmatrix} + \begin{bmatrix}0\\ M.g\\ 0\end{bmatrix} = 0, \tag{4}$$

where H_z is the inertia matrix of the object.

3. Design Control Law

3.1. Controller Design

The dynamic equation of the dual-arm system can be rewritten in a general form as follows:

$$H(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + J_B.F = \tau, \tag{5}$$

where $q = [q_1^T, q_2^T]^T$;

$$\tau = [\tau_1, \tau_2]^T;$$

$$F = \begin{bmatrix} f_1 & \lambda_1 & f_2 & \lambda_2\end{bmatrix}^T;$$

$$H(q) = blockdiag[\mathrm{H}_1(q_1), H_2(q_2)];$$
$$C(q,\dot{q}) = blockdiag[\mathrm{C}_1(q_1,\dot{q}_1), C_2(q_2,\dot{q}_2)];$$
$$G(q) = [G_1(q_1), G_2(q_2)]^T.$$

And the dynamic equation of the object also can be rewritten in the following form:

$$H_z\ddot{z} + C_z(z,\dot{z})\dot{z} + g_z = F_z. \tag{6}$$

Forces and moments that apply to the object are:

$$F_z = E.F.$$

So $F = E^+.F_z$, $\quad (7)$

where $E^+ = E^T(E.E^T)^{-1}$ is pseudo-inverse matrix of E. Combining (2), (5), (6), (7) leads to the dynamic equation of the whole system can be expressed as follows:

$$H_p\ddot{z} + C_p\dot{z} + G_p = \tau, \tag{8}$$

in which $H_p = H(q)A + J_B E^+ H_z$;

$$C_p = H(q)B + C(q,\dot{q}).A + J_B E^+ C_z(z,\dot{z});$$

$$G_p = G(q) + J_B E^+ g_z.$$

It is clear that H_p is an inertia matrix, C_p is a centrifugal/Coriolis matrix, and G_p is a gravity force in the dynamic equation of the whole system (8). It is known that dynamic parameters of system dynamics such as mass and inertia term of links and the object, friction coefficients are uncertain. It is reasonable to consider H_p, C_p, G_p including two terms: known H_0, C_0, G_0, and unknown ΔH_p, ΔC_p, ΔG_p. That means:

$$H_p = H_0 + \Delta H_p;$$

$$C_p = C_0 + \Delta C_p;$$

$$G_p = G_0 + \Delta G_p.$$

Therefore the dynamic equation (8) can be expressed in the following form:

$$H_0\ddot{z} + C_0\dot{z} + G_0 + \Delta f(z,\dot{z}) = \tau, \tag{9}$$

where $\Delta f(z,\dot{z})$ includes unknown terms

$$\Delta f(z,\dot{z}) = \Delta H_p\ddot{z} + \Delta C_p\dot{z} + \Delta G_p.$$

The error between desired trajectory and the actual trajectory of the object can be defined as:

$$e_p = z_d - z. \tag{10}$$

Define:

$$s = \dot{e}_p + \Lambda.e_p;\ \xi(t) = A.s. \tag{11}$$

where Λ is diagonal positive matrix. If $s \to 0$ then $z \to z_d$ when $t \to \infty$

Substituting (10), (11) into (9) leads to

$$H_p\dot{s} = f_0(e_p,\dot{e}_p) + \Delta f(e_p,\dot{e}_p) - C_p.s - \tau, \tag{12}$$

where
$$f_0(e_p,\dot{e}_p) = H_0(\ddot{z}_d + \Lambda.\dot{e}_p) + C_0.(\dot{z}_d + \Lambda.e_p) + G_0$$

is a nonlinear and known function and

$$\begin{aligned}\Delta f(e_p,\dot{e}_p) &= \Delta H_p(\ddot{z}_d + \Lambda.\dot{e}_p) + \Delta C_p(\dot{z}_d + \Lambda.e_p)\\ &+ \Delta G_p\end{aligned} \tag{13}$$

is the unknown function.

In the ideal case, $\Delta f(e_p,\dot{e}_p) = 0$, the control input is proposed as:

$$\tau = f_0(x) + K_s\xi, \tag{14}$$

where K_s is positive matrix.

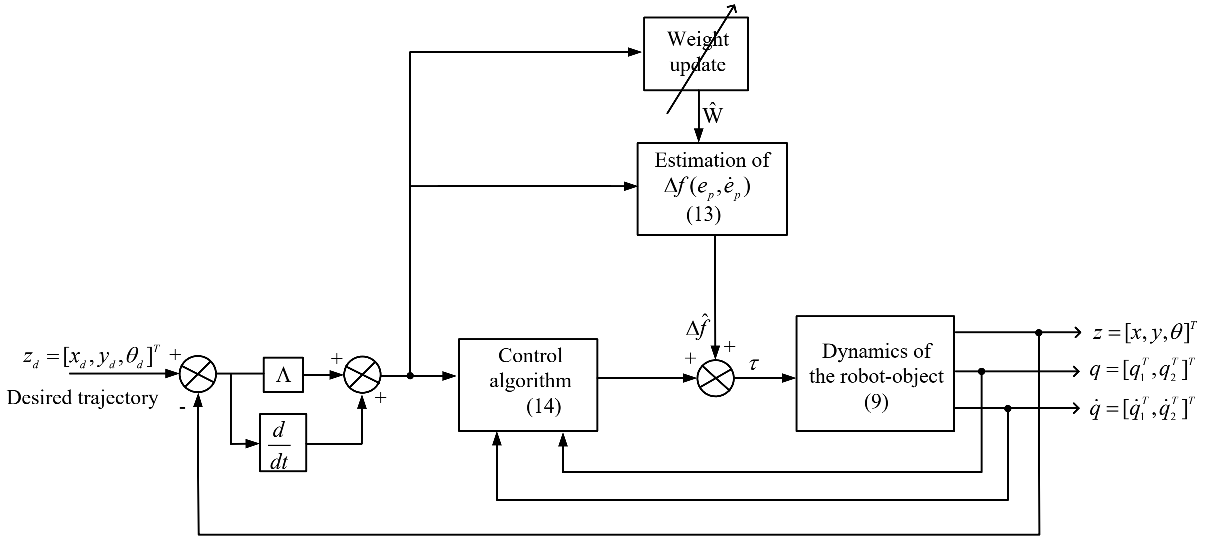


Fig. 2. Scheme of overall dual-arm robot and object system under RBRNN-adaptive controller

The closed dynamics work stably according to the Lyapunov principle. It is proved that

$$\begin{cases} z \to z_d \\ \dot{z} \to \dot{z}_d \end{cases} \text{when } t \to \infty$$

However, due to the uncertainties of dynamic parameters, the term $\Delta f(e_p, \dot{e}_p)$ always exists that breaks the stability of the whole system. In order to compensate for the uncertainties of the system dynamics, adaptive control based on Radial Basis Function Network is proposed. Fig.2 illustrates the control model using RBFNN to compensate for the uncertainties of the system

The control algorithms are:

$$\tau = f_0(e_p, \dot{e}_p) + \Delta\hat{f}(e_p, \dot{e}_p) + \mathrm{K}_s\,\xi\,, \tag{15}$$

where $\Delta\hat{f}(e_p, \dot{e}_p)$ is an approximated function of $\Delta f(e_p, \dot{e}_p)$.

3.2 Design of Radial Basis Function Neural Network

The general RBF Neural Network has a structure as depicted in Fig.3 [15]. Its simplest form is a three-layer feedforward network. The first layer corresponds to the inputs of the network, the second one is a hidden layer consisting of nonlinear activation units and the last one is the output layer corresponding to the final output of the network. Each of n components of the input vector x feeds forward to m basis functions whose output are linearly combined with weighs into the network output. Neurons in the hidden layer have Gaussian transfer functions for which outputs are inversely proportional to the distance from the center of the neuron. The Gaussian-type function can be expressed as:

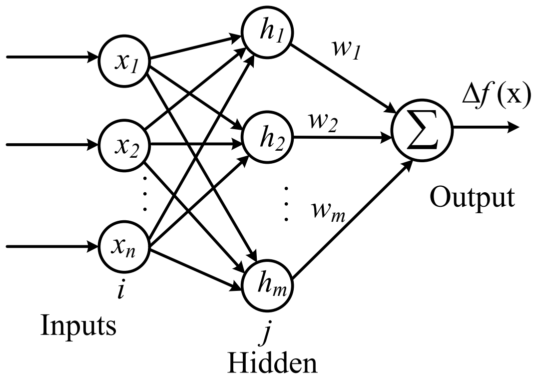


Fig. 3. Typical structure of RBF Neural Networks

$$h_j = \exp\left(-\frac{\left\| x - c_j \right\|^2}{2b_j^2} \right) \qquad j = 1, 2...., m\,, \tag{16}$$

where x is the input signal of RBFNN, m is the number of nodes in the hidden layer, c_j is the center and b_j is the variance of j^{th} basic Gaussian function.

The output of the linear layer RBFNN is

$$\Delta f(x) = \sum_{j=1}^{m} \mathrm{W}_j h_j(x)\,,$$

where W_j is the gradient matrix associated with each node j of RBFNN. By selecting the appropriate weight vector, RBF network can approximate a continuous function with arbitrary precision.

$$\Delta f(x) = W^T h(x) + \varepsilon\,, \tag{17}$$

where W^* is the optimal weight vector and ε is the error that arises from the approximation procedure of

RBF NN. Given an infinitely small constant ε_N there exists ε satisfies a condition $|\varepsilon| < \varepsilon_N$. The optimal weights are bounded $\|W^*\| \le W_{\max}$, therefore, an estimate of $\Delta f(x)$ can be written as follows:

$$\Delta\hat{f}(x) = \hat{W}^T h, \tag{18}$$

where $\hat{W}$ is the estimated weight of the optimal weight of the NN, which is generated by the online weight update algorithm.

According to the theory of RBFN, component $\Delta f(e_p, \dot{e}_p)$ of the dynamics of the system (12) is approximated by using RBFNN. Base on equation (17), $\Delta f(e_p, \dot{e}_p)$ is represented as follows:

$$\Delta f(e_p, \dot{e}_p) = W^T h(x) + \varepsilon. \tag{19}$$

$\Delta\hat{f}(e_p, \dot{e}_p)$ is the estimation of the function $\Delta f(e_p, \dot{e}_p)$, base on (18) is determined:

$$\Delta\hat{f}(e_p, \dot{e}_p) = \hat{W}^T h(x). \tag{20}$$

The weight estimation error as

$$\tilde{W} = W^* - \hat{W}.$$

Then, the error of the function $\Delta f(e_p, \dot{e}_p)$ and $\Delta\hat{f}(e_p, \dot{e}_p)$) is determined:

$$\begin{aligned}\Delta\tilde{f}(e_p, \dot{e}_p) &= \Delta f(e_p, \dot{e}_p) - \Delta\hat{f}(e_p, \dot{e}_p)\\ &= W^T h(x) + \varepsilon - \hat{W}\, h(x) = \tilde{W}\, h(x) + \varepsilon\end{aligned} \tag{21}$$

The inputs of RBFNN are $x = [e_p \;\; \dot{e}_p]^T$.

From (12), with the control in (15) and using (21), the closed dynamics becomes

$$\begin{aligned}H_p \dot{s} &= \Delta f(e_p, \dot{e}_p) - \Delta\hat{f}(e_p, \dot{e}_p) - C_p.s - K_s A.s\\ &= \Delta\tilde{f}(e_p, \dot{e}_p) - C_p.s - K_s A.s\\ &= \tilde{W} h(x) - C_p.s - K_s A.s.\end{aligned} \tag{22}$$

3.3 Update Law for NN Weighs

The candidate of Lyapunov function is chosen as:

$$V = \frac{1}{s}.s^T.H.s + \frac{1}{2} tr(\tilde{W}^T.\Gamma^{-1}\tilde{W}), \tag{23}$$

where $H = A^T H_p$.

If define $A^T J_B^T = E$ then $A^T J_B^T E^+ = I$, and $H = A^T H_p, N = A^T C_p$ then $\dot{H} - 2N$ is skew-symmetric matrix.

Differentiation of (23) leads to

$$\dot{V} = s^T.H.\dot{s} + \frac{1}{2} s^T.\dot{H}.s + tr(\tilde{W}^T.\Gamma^{-1}\dot{\tilde{W}}).$$

Due to $\dot{H} - 2A^T C_p$ is skew-symmetric matrix then $s^T(\dot{H} - 2A^T C_p)\mathrm{s} = 0$.

Combining with (22) leads to:

$$\dot{V} = -s^T \mathrm{A}^T K_s A s + tr(\tilde{W}^T.\Gamma^{-1}\dot{\tilde{W}}) + \mathrm{s}^T A^T \tilde{\mathrm{W}}^T h(x)$$

Due to $s^T.A^T = (A.s)^T$

$$\dot{V} = -(\mathrm{A.s})^T K_s (A.s) + tr(\tilde{W}^T.\Gamma^{-1}\dot{\tilde{W}}) + (A.s)^T \tilde{\mathrm{W}}^T h.$$

According to the Lyapunov stability principle, the condition for stability of closed dynamics is $\dot{V} \le 0$ then

$$\begin{aligned}&\Rightarrow tr(\tilde{W}^T.\Gamma^{-1}\dot{\tilde{W}}) + (A.s)^T \tilde{\mathrm{W}}^T h = 0\\ &\quad tr(\tilde{W}^T.(\Gamma^{-1}\dot{\tilde{W}}) + h(A.s)^T) = 0\\ &\Rightarrow \quad \Gamma^{-1}\dot{\tilde{W}} + h(A.s)^T = 0\end{aligned}$$

So the updated law for the weights of the neural network may have the form:

$$\dot{\tilde{W}} = \Gamma h(A.s)^T. \tag{24}$$

It is possible to prove that the dynamic system is stable under the control input (15) combining with the updated law (24).

4. Simulation

In order to confirm the effectiveness of the proposed RBFNN control, we carry out the simulation work in MatLab/Simulink. The desired position (x_d, y_d) and rotational angle θ_d of the grasping object are planned in 5 order-polynomial trajectories as follows:

$$x_d = 0{,}54 + 0{,}4845t^3 - 0{,}2907t^4 + 0{,}0465t^5;$$
$$y_d = 1{,}4 + 0{,}3841t^3 - 0{,}2304t^4 + 0{,}0369t^5;$$
$$\theta_d = 0{,}513t^3 - 0{,}1508t^4 + 0{,}0241t^5.$$

- The control parameters are:

$K_s = diag(15,15,15,15,15,15)$;
$\Lambda = diag(350,350,350)$.

- RBFNN has 6 inputs, 6 outputs, and 50 neural nodes in hidden layer.

Initiating value of weights *$W_0 = 6$.*

Center c_j= [-2, 2]$_{50}$.

Width b_j =10; $\Gamma = 2$.

The simulation is carried out under some cases as follows:

4.1. Case 1: The differences between the estimated terms and the actual term are 10% of the actual term as follows:

$$\Delta H_p = 10\% H_0; \Delta C_p = 10\% C_0; \Delta G_p = 10\% G_0.$$

We investigated 2 cases:

a) Without compensation $\Delta f(e_p, \dot{e}_p)$

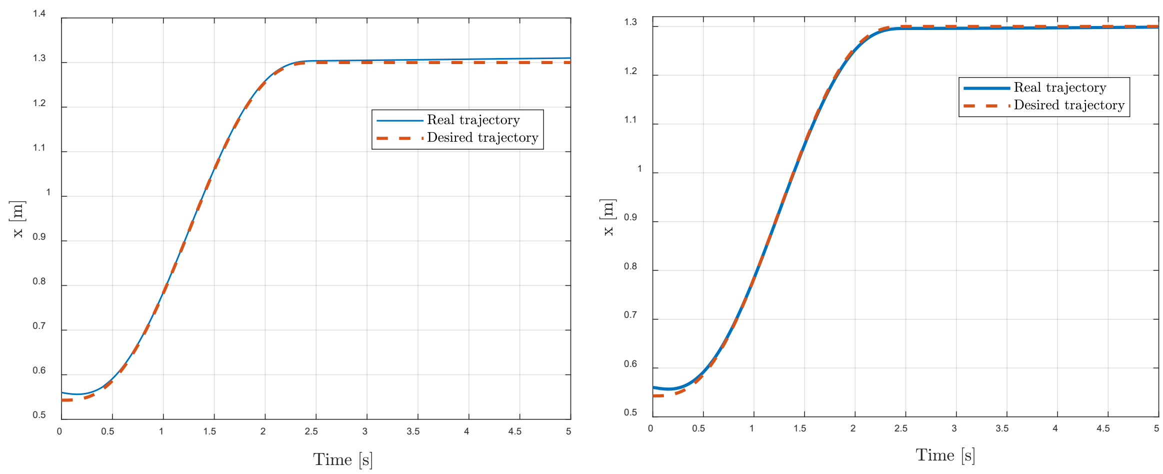
We apply the control input (14) to the dynamics of the dual arm robot – object (9).

It is possible to see the errors between the real trajectories and the desired trajectories in *x*- (Fig.4a), *y*-axis (Fig.5a) and rotation around *z*-axis (Fig.6a) are bigger in time.

b) With compensation $\Delta f(e_p, \dot{e}_p)$

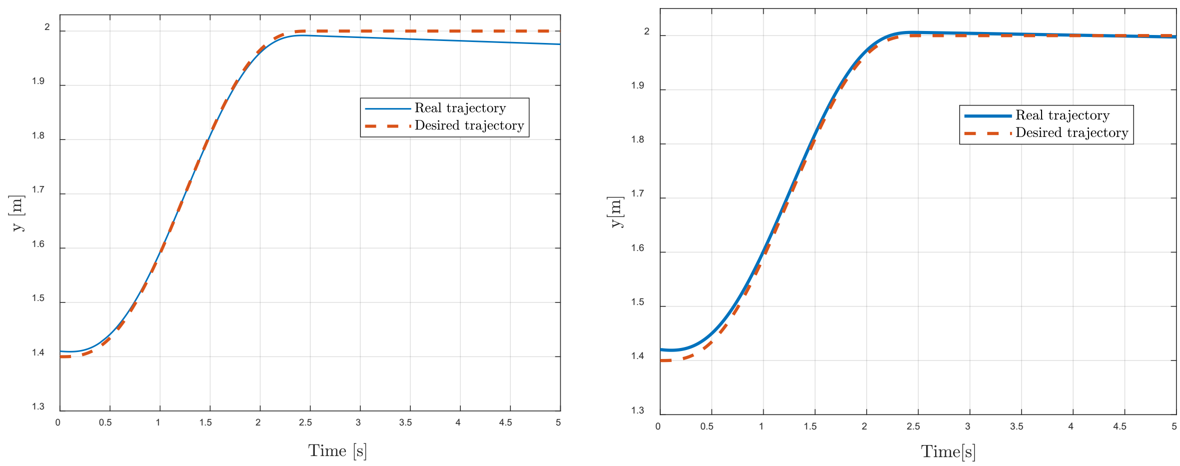
We apply the control input (15) to the dynamics of the dual arm robot – object (9). It is possible to realize that the dual-arm robot can move the object to the desired position of the mass center (x_d, y_d) in (Fig.4b), (Fig.5b) and rotate it to the desired orientation in Fig.6b. The time needed for exact tracking to the trajectories are very short, only 0.5 second for translational move in *x*- and *y*- directions, and 1 second for rotation about *z*- axis.

The simulation results display the effectiveness of the RBFNN controller in tracking trajectories of the object and its rotational angle to the desired trajectories in the case of uncertain level of dynamic parameters is not so much, 10%. For the case, we increase the uncertain level to 20%.



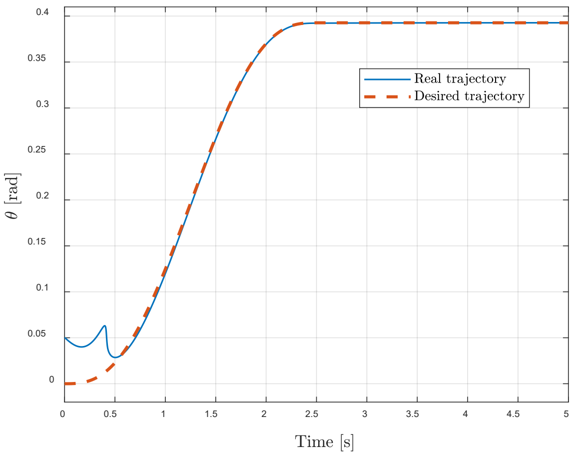
(a) When $\hat{\Delta f}(e_p, \dot{e}_p) = 0$ (b) When $\hat{\Delta f}(e_p, \dot{e}_p) \neq 0$

Fig. 4. Response of move in *x*-axis of the object

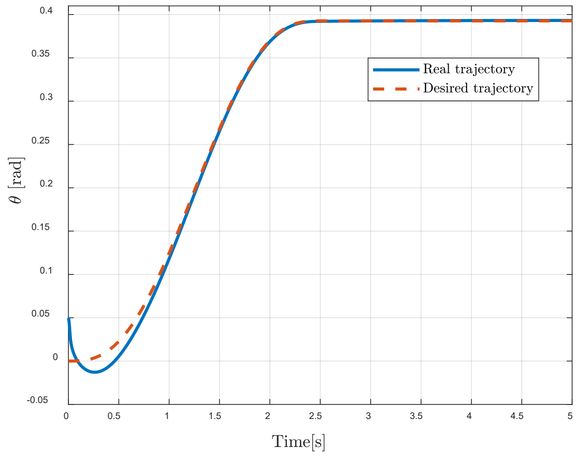


(a) When $\hat{\Delta f}(e_p, \dot{e}_p) = 0$ (b) When $\hat{\Delta f}(e_p, \dot{e}_p) \neq 0$

Fig. 5. Response of move in *y*-axis of the object

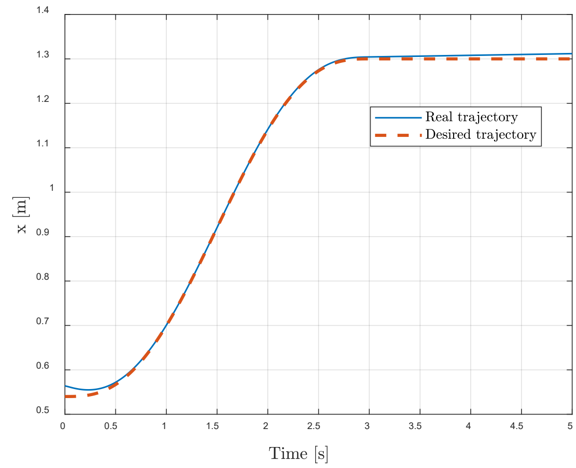


(a) When $\hat{\Delta f}(e_p, \dot{e}_p) = 0$

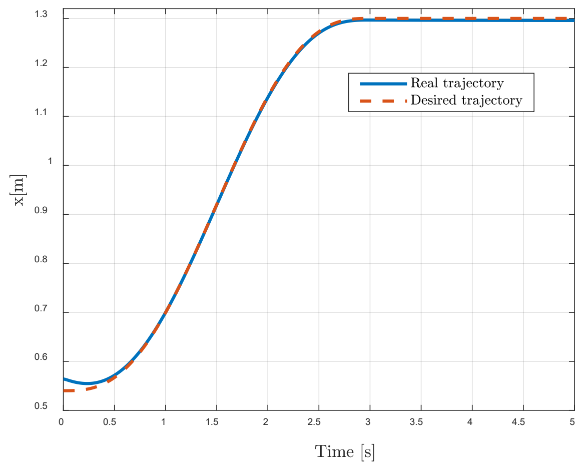


(b) When $\hat{\Delta f}(e_p, \dot{e}_p) \neq 0$

Fig. 6. Response of rotational angle of the object

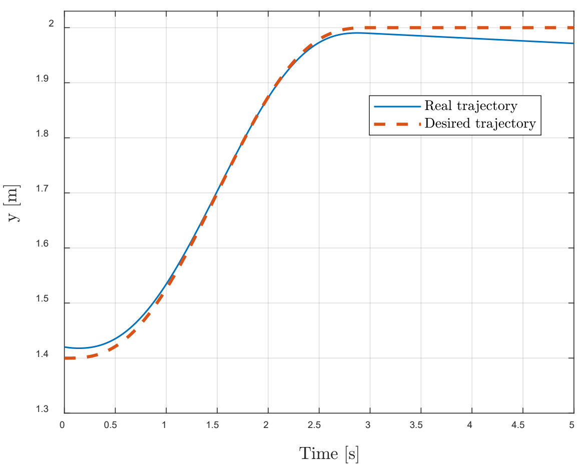


(a) When $\hat{\Delta f}(e_p, \dot{e}_p) = 0$

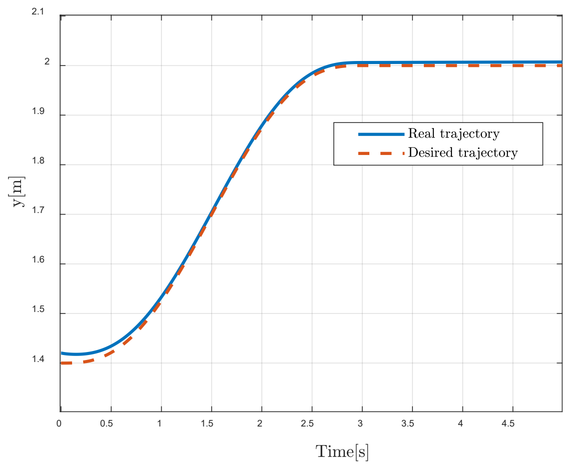


(b) When $\hat{\Delta f}(e_p, \dot{e}_p) \neq 0$

Fig. 7. Response of move in *x*-axis of the object

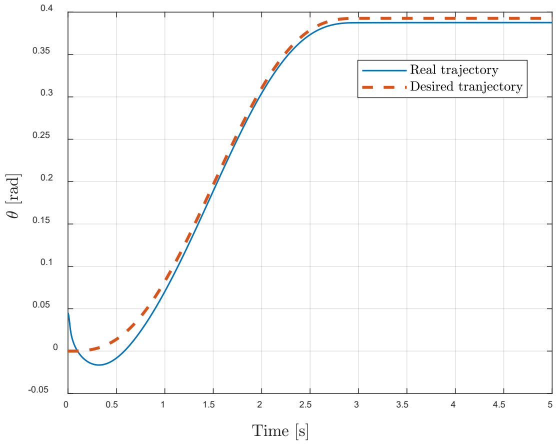


(a) When $\hat{\Delta f}(e_p, \dot{e}_p) = 0$

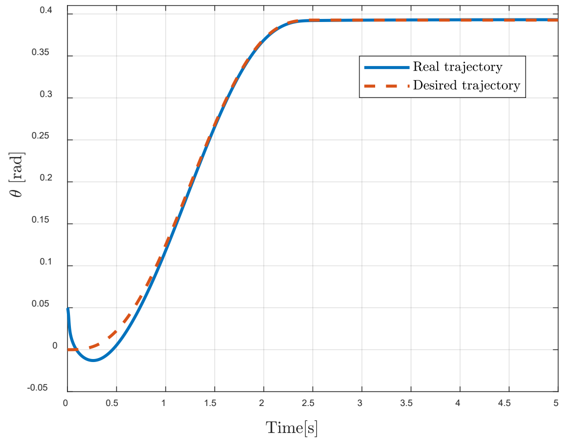


(b) When $\hat{\Delta f}(e_p, \dot{e}_p) \neq 0$

Fig. 8. Response of move in *y*-axis of the object



a) When $\hat{\Delta}f(e_p,\dot{e}_p)=0$



(b) When $\hat{\Delta}f(e_p,\dot{e}_p)\neq 0$

Fig. 9. Response of rotational angle of the object

4.2. Case 2: The differences between the estimated terms and the actual term are 20% of the actual term as follows:

$$\Delta H_p = 20\% H_0;\ \Delta C_p = 20\% C_0; \Delta G_p = 20\% G_0\,.$$

We also investigated 2 cases:

a) Without compensation $\Delta f(e_p,\dot{e}_p)$.

It is easy to realize the divergence between the desired trajectories and actual trajectories of object movement in *x*- axis (Fig.7a), in *y*-axis (Fig.8a) and rotation around *z*-axis (Fig.9a).

b) With compensation $\Delta f(e_p,\dot{e}_p)$.

The desired trajectories and actual trajectories of object in *x*- axis, *y*-axis and rotation around *z*-axis respectively are shown in Fig.7b, Fig.8b and Fig.9b

The simulation results show that the adaptive controller that based on RBFNN works effectively in compensation the dynamic uncertainties of the whole system. It is possible to conclude that when the uncertain level of dynamic parameters is in a small range, the responses of systems in tracking the desired trajectories are reasonable.

5. Conclusion

In this paper, the adaptive control using Radial Basis Function Neural Network for controlling dual-arm robotic system in manipulating a rectangle object tracking to the desired trajectories has been applied. The overall dynamics of the manipulators-object system have been formulated based on the Euler-Lagrangian principle. Radial basis function neural network has been applied to compensate uncertainties of dynamic parameters. The adaptive algorithm has been derived owning to Lyapunov stability principle to guarantee asymptotical convergence of the closed dynamic system. Simulation work on MatLab has been carried out to reconfirm the accuracy and the effectiveness of the proposed controller.

Acknowledgements

This research is funded by Hanoi University of Science and Technology (HUST) under project number T2020-PC-021

References

[1]. V. K. X. Yun, An approach to simultaneous control of trajectory and interaction forces in dual-arm configurations, IEEE Transactions on Robotics and Automation.7 (5) (1991) 618–625. https://doi.org/10.1109/70.97873

[2]. S. S. W. Gueaieba, M. Bolica, Robust computationally efficient control of cooperative closed-chain manipulators with uncertain dynamics, Automatica. 43 (2007) 842–851. https://doi.org/10.1016/j.automatica.2006.10.025

[3]. V. M. Tuneski AI, Dimirovski GM, Adaptive control of multiple robots manipulation on a dynamical environment, Proc Inst Mech Eng I: J Syst Control Eng (2001) 385–404 https://doi.org/10.1177/095965180121500411

[4]. B. R. R. Uzmay I I, Sarikaya H H, Application of robust and adaptive control techniques to cooperative manipulation, Control Eng Pract. (2004). https://doi.org/10.1016/S0967-0661(03)00017-0

[5]. M. R. R. Mohajerpoor, A. Talebi, M. Noorhosseini, R. Monfaredi, A robust adaptive hybrid force/position control scheme of two planar manipulators handling an unknown object interacting with an environment, Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering. (2011). https://doi.org/10.1177/0959651811424251

[6]. B. R. Uzmay I, and Sarikaya H, Application of robust and adaptive control techniques to cooperative manipulation, Control Eng Pract. (2004) 139–148. https://doi.org/10.1016/S0967-0661(03)00017-0

[7]. L. C. Z. Hou, M. Tan, Coordination of two redundant robots using a dual neural network, Proceedings of the International Joint Conference on Neural Network. (2006) 4187–4192.

[8]. N. K. Vikas Panwar, N. Sukavanam, Jin-Hwan Borm, Adaptive neural controller for cooperative multiple robot manipulator system manipulating a single rigid object, Applied Soft Computing. (2012) 216–227. https://doi.org/10.1016/j.asoc.2011.08.051

[9]. N. W. Zhao D, and Zhu Q, A framework of neural networks based consensus control for multiple robotic manipulators, Neurocomputing (2014). https://doi.org/10.1016/j.neucom.2014.03.041

[10]. Z. Q. Zhao D, Li N, Synchronized control with neuroagents for leader follower based multiple robotic manipulators, Neurocomputing. (2014) 149–161. https://doi.org/10.1016/j.neucom.2013.07.016

[11]. Young Hoon Joo, Le Anh Tuan, Le Quoc Tien and Pham Xuan Duong, Adaptive Neural Network Second-order Sliding Mode Control of Dual Arm Robots, International Journal of Control, Automation and Systems 15(X) (2017) https://doi.org/10.1007/s12555-017-0026-1

[12]. J. W. Fan XU, Guo-dong LU, Adaptive robust neural control of a two-manipulator system holding a rigid object with inaccurate base frame parameters, Frontiers of Information Technology & Electronic Engineering. (2018) 1316-1327. https://doi.org/10.1631/FITEE.1601707

[13]. Luu Thi Hue, Nguyen Pham Thuc Anh, Application of force observer in force control of dual-arm robot, Special issue on measurement, control and automation, vol. 21, No 3, December (2018).

[14]. Z. R. Barmak Baigzadehnoe, Alireza Khosravi, Behrooz Rezaie, On position/force tracking control problem of cooperative robot manipulators using adaptive fuzzy backstepping approach, ISA Transactions. (2017). https://doi.org/10.1016/j.isatra.2017.07.029

[15]. Liu, Jinkun, Intelligent control design and MATLAB simulation, Tsinghua University Press, Beijing and Springer Nature Singapore Pte Ltd. (2018) https://doi.org/10.1007/978-981-10-5263-7