ADAPTIVE FORCE/ POSITION CONTROL FOR DUAL-ARM SYSTEM BASED ON NEURAL NETWORK RADIAL BASIS FUNCTION WITHOUT USING A FORCE SENSOR

Luu Thi Hue[1,2], Duong Minh Duc[1], Nguyen Pham Thuc Anh[1]

[1]Ha Noi University of Science and Technology; anh.nguyenphamthuc@hust.edu.vn; duc.duongminh@hust.edu.vn

[2]Electric Power University; hue.luuthi@gmail.com

(Received September 21, 2020; Accepted December 04, 2020)

Abstract - The paper has developed an adaptive algorithm using neural network for controlling dual-arm robotic system in stable holding a rectangle object and moving it to track the desired trajectories. Firstly, an overall dynamic of the system including the dual-arm robot and the object is derived based on Euler-Lagrangian principle. Then based on the dynamics, a controller has proposed to achieve the desired trajectories of the holding object. A radial basis neural network has been applied to compensate uncertainties of system parameters. The adaptive learning algorithm has been derived owning to Lyapunov stability principle to guarantee asymptotical convergence of the closed loop system. Besides, force control at contact point is implemented without the measurements of forces and moments at contact points. Finally, simulation work on Matlab has been carried out to confirm the accuracy and the effectiveness of the proposed controller.

Key words - Adaptive control; dynamics; Radial Basis Function Neural Network; dual-arm robotic; force/ position control

1. Introduction

Cooperative manipulators have been witnessing an increasing interest due to their versatility within robotic applications such as assembly heavy objects, transporting, and so on. Cooperative manipulators will replace humans to work in hazardous and dangerous environments better than single robots. However, it is much more complicated to design a controller of cooperative manipulation because the dynamics of these systems are complicated and highly nonlinear. Some control algorithms for multiple-robot systems have been announced. Yun and Kumar [1] used nonlinear feedback linearization for coordinated motion control in dual-arm configurations. Robust control algorithm for position and internal forces control in cooperative multiple manipulator systems with the uncertainty dynamics is presented in [2]. An adaptive control scheme based on the inverse dynamics controller structure for the position control problem of multi-arm manipulators is presented in [3]. An adaptive hybrid force/position control scheme of two planar manipulators coordinates moving an object with unknown parameters, but the parameters of known robot model are proposed [4]. [5] presented a robust adaptive control method to cooperative two planar robots holding an object, the parameters uncertainty of the system are estimated using an estimation law to control the system properly.

Most of these kinds of adaptive which is presented above are adaptive controllers based on the dynamic model to build adaptive update algorithm. Then, we estimate parameters of the uncertainty dynamics. The algorithm is relatively complicated, influencing on speed calculation and practical applicability is limited. Recently, the neural network has strongly shown changes to develop controllers. The most useful property of neural network in control is their ability to approximate arbitrary linear or nonlinear mapping through learning. The idea of neural network based control, the neural network is used to compensate unknown nonlinear dynamics and compensate for structured/unstructured existing uncertainties in the dynamic model. A dual neural network approach is applied in [6] to resolve the coordination problem of two redundant robots. An adaptive neural network position/force control approaches have been developed for cooperative robots system with unknown dynamical models [7]. Besides, estimation using neural network is developed [8, 9] to estimate system uncertainty. Using inverse dynamics model to design the combination of Second-Order Sliding Mode Control (SOSMC) and neural networks to estimate dual arm robot uncertainty is presented in [10]. An adaptive robust neural control based on inverse dynamics model, a radial basis function neural network is adopted for dynamical uncertainty estimation is proposed in [11].

Although above adaptive control using the neural network mostly coped with the dynamic model uncertainties, some controllers are designed to adopt model-based inverse dynamics algorithm and approximating of such inverse dynamic algorithm. Then, the controllers are complicated, the condition of contact between the end-effector of robots and the object may not be guaranteed. Moreover, the controllers are used to track the desired internal forces arising in the system. Thus, the force measurement is required.

In this paper, we propose an adaptive controller based on radial basic network (RBF) to control cooperative manipulator system manipulating to hold and move a rigid object according to a design trajectory planning. The controller is designed without using inverse dynamics and force sensor, and the neural network is used to compensate unknown or inaccurate parts of the real model. Using the proposed controller, the computation is reduced, and the condition of contact between the end-effector of robots and objects is guaranteed.

The rest of the paper is organized as follows: section 2 formulates the dynamics of dual-arm robot during holding and manipulating an object under geometric constraints. In section 3, a neural network adaptive hybrid force/position control scheme is designed without force measurement. In section 4, simulation results are presented to confirm the effectiveness of the proposed control scheme. Section 5 is for the conclusion.

2. Dynamic Model

2.1. Description of the system

The dual-arm robot under study has been illustrated in Figure 1. Each arm is a planar robot with three degrees of freedom. The dual-arm robot will manipulate a rectangular rigid object in 2D space.

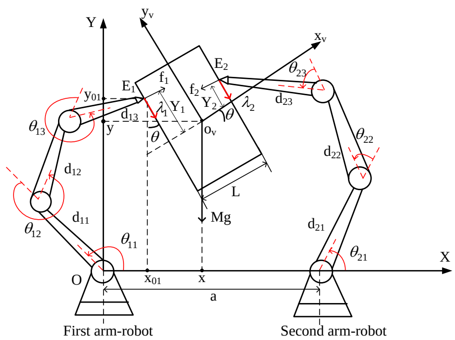

Figure 1. *Model of dual-arm robot in holding an object*

Vector $q_i = \left[\theta_{i1}, \theta_{i2}, \theta_{i3}\right]^T$ is joint angle vector of the i^{th} robot (i =1, 2) and vector $z = \left[x, y, \theta\right]^T$ represents the position and rotational angle of the object in the reference frame OXY.

The end-effector of dual-arm robot impacts on the object, these interactional forces f_1, f_2 arising have the normal direction to the surface of the object and these force λ_1, λ_2 has the parallel direction to the surface of the object. The object-robot system is confined in the vertical plane then gravitational force affects the system.

2.2. Kinematic relationships

Refering to Figure 1, we can represent the position o_v of the mass center (x, y) of the object in the reference frame $\{OXY\}$ as:

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

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

And the distance Y_i from E_i to the X-axis of the object frame can be determined as

$$\begin{cases} Y_1 = (x - x_{01})\sin\theta - (y - y_{01})\cos\theta \\ Y_2 = (x - x_{02})\sin\theta - (y - y_{02})\cos\theta \end{cases}$$

Taking differential of equation (1), it is obtained that,

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

where $A_i = D_i^{+}.\left[\cos\theta \quad \sin\theta\right].J_{0i}$; $D_i^{+} = D_i^T(D_i.D_i^T)^{-1}$ is the pseudo-inverse matrix, $D_i = \left[\cos\theta \quad \sin\theta \quad -Y_i\right]$,

$J_{0i} = \left(\dfrac{\partial x_{0i}}{\partial q_i}, \dfrac{\partial y_{0i}}{\partial q_i}\right)^T$ is Jacobian matrix of the i^{th} robot.

Finally we have

$$\dot{q} = A\dot{z};\ \ddot{q} = A\ddot{z} + B\dot{z} \tag{2}$$

Where $q = [q_1^T, q_2^T]^T$; $A = [(A_1^{-1})^T, (A_2^{-1})^T]^T$,

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

2.3. Dynamics of the dual-arm robot – object system

Dynamics of the system can be formulated on the basis of Euler-Lagrange approach. The Lagrangian function can be defined as:

$$L = K - P$$

where K is kinematic energy and P is the potential energy of the robot-object system [12].

Applying Euler-Lagrange method, dynamic equation system of the whole robot-object system can be formulated.

The dynamic equation of the i^{th} robot.

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

where $H_i(q_i)$ is the inertia matrix, $C_i(q_i, \dot{q}_i)$ is the Coriolis and centrifugal matrix, can be determined from $H_i(q_i)$ matrix as $\dot{H}(q) - 2C(\dot{q}_i, q)$ is skew-symmetric, $G_i(q_i)$ is the gravitational vector, J_{0i} is Jacobian matrix of the i^{th} arm-robot.

The dynamic equation of the object can be described as.

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

Where, H_z is the inertia matrix of the object, $H_z = diag[M, M, J]$ with M is mass of the object and J inertia moment of the object.

3. Controller Law

The control purpose is to provide the joint torques so that the motion trajectory of the object converges to the desired trajectory with the dynamic model uncertainty. In this paper, the adaptive controller is proposed without requiring measurement of the force and moment at the contact points.

3.1. Controller design

3.1.1. Position control

The dynamic of the dual-arm robot represented in Eq. (3) can be concisely reformulated as [13]

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

where $\tau = [\tau_1, \tau_2]^T$; $F = [f_1, \lambda_1, f_2, \lambda_2]^T$;

$$J_B^T = \begin{bmatrix} B_{1J}^T & E_{1T}^T & 0_{3\times1} & 0_{3\times1} \\ 0_{3\times1} & 0_{3\times1} & -B_{2J}^T & E_{2T}^T \end{bmatrix},$$

where $B_{iJ} = \left[\cos\theta, \sin\theta\right].J_i$; $E_{iT} = \left[\sin\theta, -\cos\theta\right].J_i$

$$H(q) = blockdiag[\mathrm{H}_1(q_1), H_2(q_2)];$$
$$C(q,\dot{q}) = blockdiag[\mathrm{C}_1(q_1,\dot{q}_1), C_2(q_2,\dot{q}_2)];$$

$$G(q) = [G_1(q_1),\ G_2(q_2)]^T .$$

The dynamic of the object in Eq. (4) can be rewritten as

$$H_Z\ddot{z} + C_Z(z,\dot{z})\dot{z} + g_Z = F_Z . \quad (6)$$

Force/torque on the object respectively

$$F_Z = E.F .$$

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

where $E^+ = E^T(E.E^T)^{-1}$ is the pseudo-inverse of grasp-matrix *E*. Using these relations along with Eq. (2), Eq. (5), Eq. (6) and Eq. (7), the combined dynamic model of the cooperative robot manipulator system is obtained as follows:

$$H_P\ddot{z} + C_P\dot{z} + G_P = \tau , \quad (8)$$

where $H_p = H(q)A + J_B^T E^+ H_z$;

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

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

The inertia matrix H_p, the centrifugal and Coriolis matrix C_p and the gravity vector G_p in the dynamical model Eq. (8), include the physical parameters of dual-arm robot and object such as links lengths, links masses, moments of inertial, object mass and so on. It is difficult to obtain these actual values because the measuring errors, environment and play load variations. Therefore, we assume that the actual value H_p, C_p, G_p are represented by the nominal parts H_0, C_0, G_0 and the uncertain parts ΔH_p, ΔC_p, ΔG_p respectively, that is,

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

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

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

Then, the robot-object dynamic model Eq. (8) can be represented as.

$$H_0\ddot{z} + C_0\dot{z} + G_0 + Y(z,\dot{z}) = \tau , \quad (9)$$

where $Y(z,\dot{z})$ contains the uncertain parts of the dynamic model and can be expressed as follows:

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

Define the following tracking errors

$$e_p = z_d - z . \quad (10)$$

Now, consider the following standard filtered tracking errors as.

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

where Λ is a positive definite symmetric design parameter matrix. When $s \to 0$ implies that $z \to z_d\ as\ t \to \infty$.

Using Eq. (9) – (11) the robot dynamic can be written in terms of filtered tracking error as

$$H_p\dot{s} = f_0(x) + \Delta f(x) - C_p.s - \tau , \quad (12)$$

where $f_0(x) = H_0(\ddot{z}_d + \Lambda.\dot{e}_p) + C_0.(\dot{z}_d + \Lambda.e_p) + G_0$ is the nominal nonlinear part, and

$$\Delta f(x) = \Delta H_p(\ddot{z}_d + \Lambda.\dot{e}_p) + \Delta C_p.(\dot{z}_d + \Lambda.e_p) + \Delta G_p \quad (13)$$

is the uncertain nonlinear part.

Assume that when the optimal design trajectory runs time for the object: The physical parameters of dynamic model is exactly known, now the dynamic equation of the system has only parts H_0, C_0, G_0, the model of the system is considered as the ideal model.

The following controller is proposed

$$\tau = f_0(x) + K_s\,\xi \quad (14)$$

with K_s is a positive definite gain matrix.

According to the Lyapunov stability principle, the theoretical analysis may lead to the conclusion that

$$\begin{cases} z & \to z_d \\ \dot{z} & \to \dot{z}_d \end{cases} \quad as\ \ t \to \infty .$$

So, ideal model is controlled stable. But in fact, the dynamic equation of the system always exists uncertainly parts as Eq. (9). The system Eq. (9) using the control law Eq. (14), the system will be unstable. Therefore, the authors propose using the RBF network to compensate uncertainly parts, the errors position/direction of the object is minimum. The controller algorithm using the RBF network to compensate unknowingly parts has been illustrated in Figure 2.

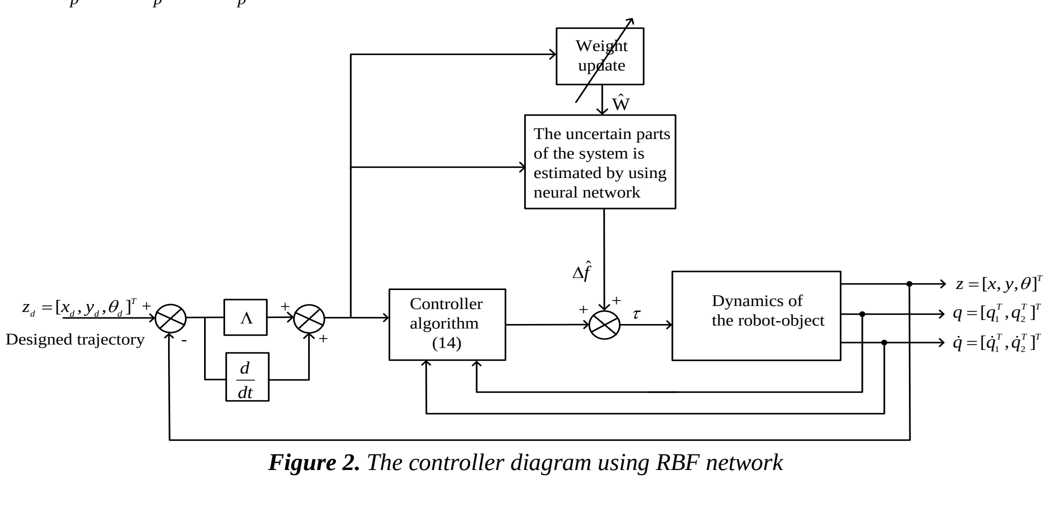

Figure 2. *The controller diagram using RBF network*

Controller algorithm is designed for the real model of the system based on Eq. (14), NN has been add to compensate the uncertainly parts. The following controller of position is proposed

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

3.1.2. Desired force applied to the object

Let us define a reference velocity of the object by

$$\dot{z}_r = (\dot{z}_d + \gamma e_p)$$

where γ is a positive definite gain matrix. The desired force is generated by an estimated reference model of the object as follows [15]:

$$\begin{aligned} F_z^{\,d} &= \hat{H}_z\ddot{z}_r + \hat{C}_z(z,\dot{z}_r)\dot{z}_r + \hat{g}_z \\ &= Y_0(z,\dot{z}_r,\ddot{z}_r)\hat{\sigma}_0, \end{aligned} \tag{16}$$

where $\hat{\sigma}_0$ is a parameter estimate of σ_0 and $\hat{H}_z, \hat{C}_z$ and $\hat{g}_z$ are estimates of M_z, C_z and g_z.

An adaptive law of the dynamic parameter of the object is given by.

$$\dot{\hat{\sigma}} = -\Gamma_0 Y_0^T(z,\dot{z}_r,\ddot{z}_r)s_0\,, \tag{17}$$

where Γ_0 is a positive definite gain matrix and s_0 is a residual error given by.

$$s_0 = \dot{z} - \dot{z}_r = -\dot{e}_p - \gamma e_p$$

The desired force applied to the object is updated based on the estimates of the dynamic parameters of the object.

The desired force at contact points should satisfy the relation given in Eq. (7). Thus, the desired force at a contact point is given by.

$$F^d = E^+ F_z^{\,d} \tag{18}$$

3.1.3. Hybrid force/position control

Now, position control and force control without the measurement of force at contact points is proposed,

$$\tau = f_0(y) + \Delta\hat{f}(y) + J_B^T F^d + K_s\xi\,. \tag{19}$$

3.2. Radial Basis Function Neural Network

As a typical neural network, it has been proved that the radial basis function neural network (RBFNN) can approximate any nonlinear function. The RBFNN with multiple input and single output can be described as in figure 3 to compensate the uncertainly parts $\Delta f(x)$ of the real model as [14].

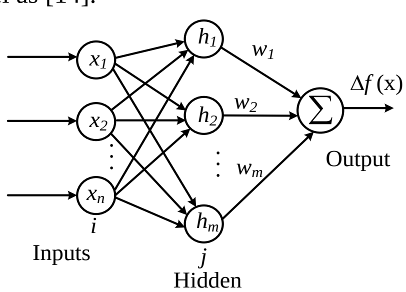

Figure 3. *RBF network*

The structure of a typical three-layer RBF neural network is shown as Figure 3. In RBF neural network has *n* inputs $x = [x_1, x_2, ..., x_n]^T$, *m* is the number of nodes in the hidden layer, the Gaussian function is usually chosen as the hidden layer output function $h_j(x)$

$$h_j(x) = \exp\left(-\frac{\left\|x - c_j\right\|^2}{2b_j^2}\right) \qquad j = 1, 2...., m\,, \tag{20}$$

where, c_j represents the coordinate value of center point of the Gaussian function, b_j represents the width value of Gaussian function for neural net *j*. The output of the RBF network is the linear superposition of the hidden layer.

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

W_j is the weight vector of the neural network. By selecting the appropriate weight vector, RBF network can approximate a continuous function with arbitrary precision.

$$\Delta f(x) = \mathrm{W}^T h(x) + \varepsilon\,, \tag{21}$$

W is the optimal weight vector, ε is approximation error. In fact, for any choice of a positive number ε_N , one can find a NN such that $|\varepsilon| < \varepsilon_N$. For a specified value of ε_N the ideal approximating NN weight exist. Then, an estimate of $\Delta f(x)$ can be given by.

$$\Delta\hat{f}(x) = \hat{\mathrm{W}}^T h\,, \tag{22}$$

$\hat{\mathrm{W}}$ is estimate of the ideal NN weight that is provided by on-line weight tuning algorithm.

The weight estimation error is as

$$\tilde{W} = \mathrm{W} - \hat{\mathrm{W}}\,.$$

Then, the functional estimation error is $\Delta\tilde{f}(x)$, which is determined.

$$\Delta\tilde{f} = \Delta f - \Delta\hat{f} = \mathrm{W}^T h + \varepsilon - \hat{\mathrm{W}}\mathrm{h} = \tilde{\mathrm{W}}h + \varepsilon \tag{23}$$

From Eq. (13) may choose inputs of RBF $x = [e_p \;\; \dot{e}_p]^T$

From Eq. (12), with the control in Eq. (19) and using Eq. (18), eq. (23) the closed dynamics becomes

$$\begin{aligned} H_p\dot{s} &= \Delta f(x) - \Delta\hat{f}(x) - C_p.s - K_s A.s - J_B F^d \\ &= \Delta\tilde{f}(x) - C_p.s - K_s A.s - J_B F^d \\ &= \tilde{W}h - C_p.s - K_s A.s - J_B^T E^+ F_z^d\,. \end{aligned} \tag{24}$$

3.3. Adaptive Law Design

Let an estimate error vector of the object's dynamic parameters be defined as $\Delta\sigma_0 = \hat{\sigma}_0 - \sigma_0$; The following is then obtained:

$$\begin{aligned} & Y_0(z,\dot{z},\ddot{z})\sigma_0 - Y_0(z,\dot{z}_r,\ddot{z}_r)\hat{\sigma}_0 \\ &= H_z\dot{s}_0 + C_0 s_0 - Y_0(z,\dot{z}_r,\ddot{z}_r)\Delta\sigma_0 \end{aligned} \tag{25}$$

The system is considered stable according to the Lyapunov stability principle, the candidate of Lyapunov function can be selected as

$$L = \frac{1}{2}s^T.H.s + \frac{1}{2}tr(\tilde{W}^T.\Gamma^{-1}\tilde{W}) + \frac{1}{2}s_0^T H_z s_0 + \frac{1}{2}\Delta\sigma_0^T\Gamma_0^{-1}\Delta\sigma_0 \tag{26}$$

where $H = A^T H_p$.

The following lemmas as [7]:

Lemma 1: $A^T J_B^T = E$ and $A^T J_B^T E^+ = I$, where I is the identity matrix.

Lemma 2: Let $H = A^T H_p, N = A^T C_p$ then $\dot{H} - 2N$ is skew symmetrix.

The time derivative L of the Lyapunov function becomes.

$$\dot{L} = s^T.H.\dot{s} + \frac{1}{2}s^T.\dot{H}.s + tr(\tilde{W}^T.\Gamma^{-1}\dot{\tilde{W}}) + s_0^T H_z \dot{s}_0 + \frac{1}{2}s_0^T \dot{H}_z s_0 + \Delta\sigma_0^T \Gamma_0^{-1} \Delta\dot{\sigma}_0.$$

Using lemma 2 then $\dot{H} - 2A^T C_p$ is skew symmetric, so $s^T(\dot{H} - 2A^T C_p)s = 0$.

Then, using Eq. (17), (24) and Eq. (25) the derivative L becomes

$$\dot{L} = -s^T A^T K_s A s + tr(\tilde{W}^T.\Gamma^{-1}\dot{\tilde{W}}) + s^T A^T \tilde{W}^T h - s^T A^T J_B^T E^+ F_z^d + s_0^T(F_z - F_z^d)$$

Due to properties of matrices $s^T.A^T = (A.s)^T$ and using lemma 1, we have.

$$\dot{L} = -(A.s)^T K_s (A.s) + tr(\tilde{W}^T.\Gamma^{-1}\dot{\tilde{W}}) + (A.s)^T \tilde{W}^T h - s^T F_z^d + s_o^T(F_z - F_z^d)$$

To stabilize the system then $\dot{L} \le 0$

$$\Rightarrow tr(\tilde{W}^T.\Gamma^{-1}\dot{\tilde{W}}) + (A.s)^T \tilde{W}^T h = 0$$

$$tr(\tilde{W}^T.(\Gamma^{-1}\dot{\tilde{W}}) + h(A.s)^T) = 0$$

$$\Rightarrow \quad \Gamma^{-1}\dot{\tilde{W}} + h(A.s)^T = 0.$$

The adaptive NN weight update law is as

$$\dot{\hat{W}} = \Gamma h(A.s)^T. \tag{27}$$

Control law Eq. (19) with the integrated adaptation law Eq. (17) and Eq. (27). According to the Lyapunov stability principle, the system is stable.

4. Simulation Work

In order to verify the above conclusion of stable controlling the object by means of dual-arm robot, simulation of the closed dynamics of the whole system has been carried out in Matlab/Simulink. The effectiveness of the proposed controller will be surveyed. The parameters of the dual-arm robot and the object are the same as in [12].

Trajectory planning is designed by fifth-order polynomial trajectories for the position and rotational angle of an object, which are given by the equation

$$x = 0,54 + 0,7561t^3 - 0,5508t^4 + 0,107t^5;$$

$$y = 1,4 + 0,5728t^3 - 0,4173t^4 + 0,0811t^5;$$

$$\theta = 0,3999t^3 - 0,2913t^4 + 0,0566t^5.$$

- The parameter of controller

K_s = *diag* (15,15,15,15,15,15)

$\Lambda = diag(350,350,350)$.

$\gamma = diag(0.3, 5, 0.5)$

$\Gamma_0 = diag(10,10)$

- The structures and parameter of neural network

NN RBF is composed of 6 input units, 6 output units and 50 hidden layer nodes. The initial values of the adjustable weight are selected as $W_0 = 6$.

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

The width b_j =10;

The learning parameter as $\Gamma = 2$.

❖ Simulation results with the model deviation

$$\Delta H_p = 30\% H_0;\ \Delta C_p = 30\% C_0; \Delta G_p = 30\% G_0.$$

The simulation results for hybrid force/position control of the object according to the planed trajectory and desired force with the 30% deviation of the model are shown in Figure 4 – Figure 7.

From the results in Figure 4 to Figure 7, we can see that the adaptive hybrid position/force control scheme can work effectively. The tracking performances of the position (Figure 4 – Figure 6) of the object can be achieved in *x*, *y* and rotational angle direction. In the contact space, the total force and torque tracking is shown in Figure 7. It is obvious that the adaptive hybrid force/position control scheme can achieve the good convergence speed with small steady-state errors.

a)

b)

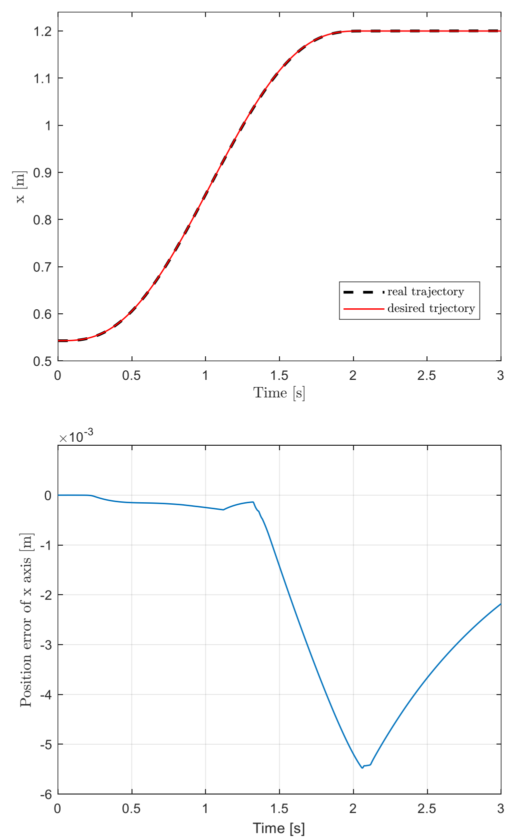

***Figure 4**. The object motion trajectory along x-axis, a) Object position, b) Position error*

a)

b)

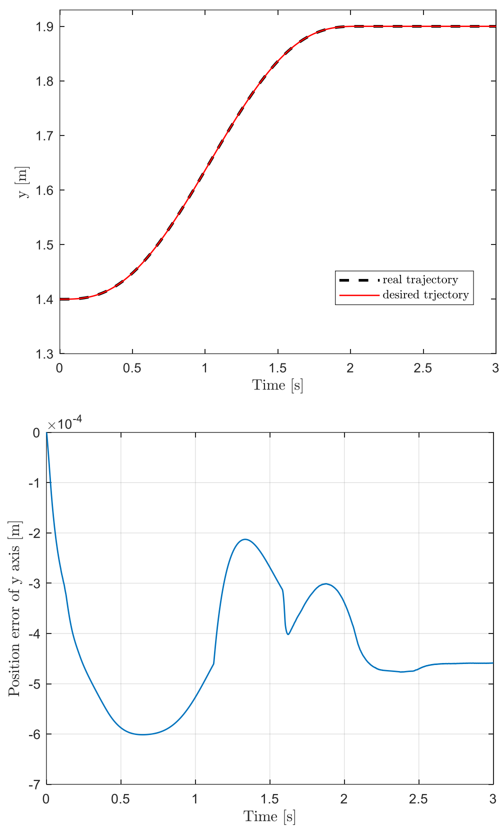

Figure 5. *The object motion trajectory along y-axis; a) Object position; b) Position error*

a)

b)

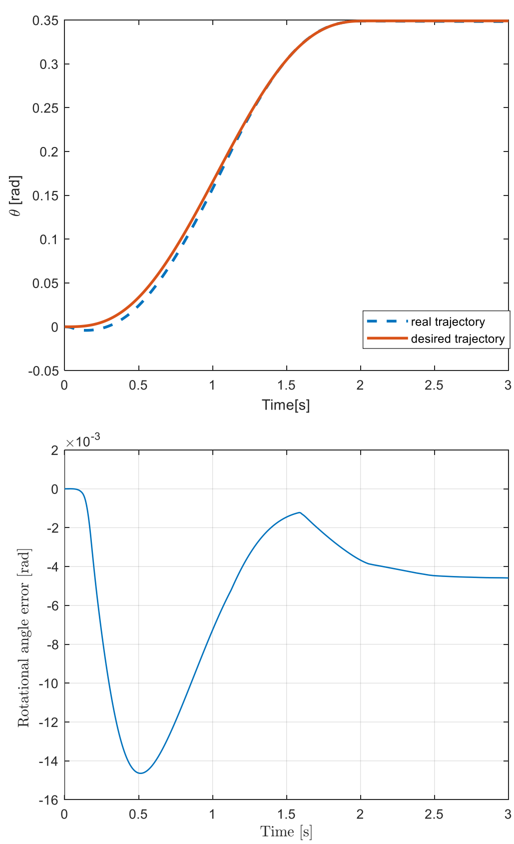

Figure 6. *The object motion trajectory rotational angle, a) Rotational angle of object; b) Rotational angle error*

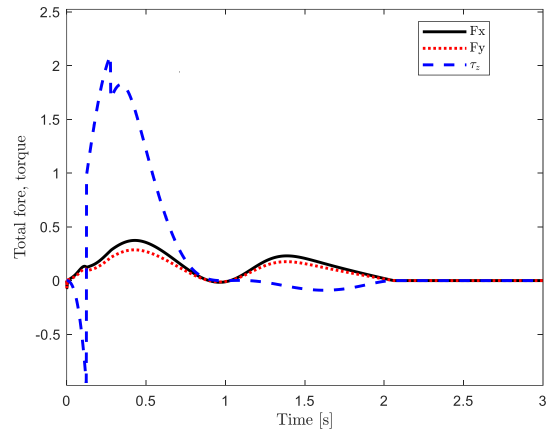

Figure 7. *Total force and torque at the center of mass of the object*

The simulation results show that the controller is working well and stably, the RBF network has compensated completely uncertainly parts of the real model. The position and rotational angle of the object converge to the desired trajectory, the degree of adjustment is zero and the static deviation is zero. As shown in Figure 4 – Figure 6, the maximum error between desired positions and real positions along x axis, y axis and rotational angle direction are only 5 [mm], 0.6 [mm] and 0.015 [rad]. As shown in Figure 7, the equilibrium position, the total force and torque applied to the object are zero, it means that the object stays at the fixed desired position. The object is held assuredly and stably at contact points. The adaptation in control is working successfully and effectively.

5. Conclusion

In this paper, an adaptive controller based on online radial basis neural network has been proposed for coordinated control of dual-arm robots manipulating a single rigid object. The dynamic model of the manipulators and the object has been derived based on Euler-Lagrangian principle. Based on this model, the controller is proposed to achieve the desired trajectory of the object. The RBF network is used to compensate uncertainties of system parameters as well as track the desired forces in the system impact on the object by estimated reference model of the object. The neural network is learned online without requirement of preliminary offline learning. The NN weights may be simply initialized to zero or randomized and errors may be kept arbitrarily small. The proposed control method does not require measurement of forces at contact points. Forces and moments at the contact points impact on the object, so it has been ensured that the object is stably held. The stability of the system is proved using Lyapunov function, through which adaptive learning law is built. The simulation results show that the RBF neural network with online update law can effectively compensate the uncertain dynamic components of the system.

Acknowledgements: This research is funded by the 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, pp. 618–625.

[2] S. S. W. Gueaieba, M. Bolica, "Robust computationally efficient control of cooperative closed-chain manipulators with uncertain dynamics", *Automatica*, 43, 2007, pp. 842–851.

[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, pp. 385–404.

[4] 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*, Vol 226, Issue 4, 2012, pp. 509-522.

[5] B. R. Uzmay I, and Sarikaya H, "Application of robust and adaptive control techniques to cooperative manipulation", *Control Eng Pract*, 2004, pp. 139–148.

[6] 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, pp. 4187–4192.

[7] 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, pp. 216–227.

[8] N. W. Zhao D, and Zhu Q, "A framework of neural networks based consensus control for multiple robotic manipulators", *Neurocomputing*, 140, 2014, pp. 8–18.

[9] Z. Q. Zhao D, Li N, "Synchronized control with neuroagents for leader follower based multiple robotic manipulators", *Neurocomputing*, 2014, pp. 149–161.

[10] Y. H. J. 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.

[11] 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, pp. 1316-1327.

[12] L.T.Hue, N. P. T. 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.

[13] 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*, Vol. 70, 2017, pp. 432-446.

[14] Liu, Jinkun, *Intelligent control design and MATLAB simulation*, Singapore: Springer Singapore, 2018, pp. 122-124.

[15] Haruhisa Kawasaki, Satoshi Ueki, Satoshi Ito, "Decentralized adaptive coordinated control of multiple robot arms without using a force sensor", *Automatica*, 42, 2006, pp. 481–488.