Original Research Article:
full paper

(2024), «EUREKA: Physics and Engineering»
Number 2

# ADAPTIVE FUZZY-NEURAL NETWORK EFFECTIVELY DISTURBANCE COMPENSATE IN SLIDING MODE CONTROL FOR DUAL ARM ROBOT

***Nam Khanh Nguyen***
*Faculty of Electrical Engineering*[1]

***Quang Viet Bach Bui***
*University of Engineering and Technology*[2]

***Quang Huy Nguyen***
*University of Engineering and Technology*[2]

***Truong Giang Luong***
*University of Engineering and Technology*[2]

***Xuan Dang Dang***
*University of Engineering and Technology*[2]

***Hai Xuan Le***
*International School*[2]

***Linh Ngoc Nguyen***✉
*International School*[2]
*nlnguyen@vnu.edu.vn*

[1]*Hanoi University of Industry*
*298 Cau Dien str., Bac Tu Liem District, Hanoi, Vietnam, 100000*

[2]*Vietnam National University*
*144 Xuan Thu str., Hanoi, Vietnam, 100000*

✉**Corresponding author**

**Abstract**

In this study, an Adaptive Backstepping Sliding Mode Controller (ABSMC) is introduced based on the Radial Basis Function (RBF) neural network and a fuzzy logic modifier. The proposed method is used to control a Dual-Arm Robot (DAR) – a nonlinear structure with unstable parameters and external disturbances. The control aims to track the motion trajectory of both arms in the flat surface coordinate within a short time, maintaining stability, and ensuring that the tracking error converges in finite time, especially when influenced by unforeseen external disturbances. The nonlinear Backstepping Sliding Mode Control (BSMC) is effective in trajectory tracking control; however, undesired phenomena may occur if there are uncertain disturbances affecting the system or model parameters change. It is proposed to use a neural network to estimate a nonlinear function to handle unknown uncertainties of the system. The neural network parameters can be adaptively adjusted to optimal values through adaptation rules derived from Lyapunov's theorem. Additionally, fuzzy logic theory is also employed to adjust the controller parameters to accommodate changes or unexpected impacts. The performance of the Fuzzy Neural Network Backstepping Sliding Mode Control (FNN-BSMC) is evaluated through simulation results using Matlab/Simulink software. Two simulation cases are conducted: the first case assumes stable model parameters without uncertain disturbances affecting the joints, while the second case considers a model with changing parameters and disturbances. Simulation results demonstrate the effective adaptability of the proposed method when the system model is affected by various types of uncertainties from the environment.

**Keywords**: dual-arm robots, backstepping sliding mode control, adaptive control, fuzzy logic, neural network.

**DOI: 10.21303/2461-4262.2024.003250**

## 1. Introduction

Dual-Arm Robot (DAR), also known as a dual manipulator, can replace and assist humans in tasks that have heavy load, require high precision, or take place in challenging working environments.

They are widely utilized in industrial, medical, manufacturing, or construction settings. This type of robot has numerous outstanding advantages, such as flexibility, high precision, and suitability for lifting heavy objects when compared to other types of industrial robots. When considering the same task, the joints of a DAR operating in parallel will require lower torque than a single-arm robot. However, DARs pose challenges the much more complex in terms of dynamics and kinetics compared to conventional serial-link systems. Consequently, controlling DARs imposes higher requirements, laying the foundation for our research.

To date, numerous worldwide studies have been published to control the motion trajectory of this model. Sliding Mode Control (SMC) [1] has been applied to the robot's load-carrying movement. Additionally, Adaptive Control (AC) methods have been applied to address disturbances affecting DAR [2–5]. Besides, nonlinear backstepping control algorithm with adaptive fuzzy control laws to adjust controller parameters is proposed in [2]. In [3, 4], the application of neural network learning rules is suggested to control stable systems impacted by disturbances. In [6], the authors designed a Backstepping Sliding Mode Control via Nonlinear Disturbance Observer. However, no method has thoroughly addressed the issue of compensating for uncertain disturbances from the environment and sudden model changes simultaneously.

In other studies, trajectory tracking control problems for robots, including DAR, have been extensively studied, with various algorithms aimed at enhancing control quality. Novel and powerful control algorithms have been increasingly researched and applied based on nonlinear control algorithms and machine learning methods. Moreover, in the broader field of robot control, advanced control methods such as Model Predictive Control (MPC) [7, 8], Reinforcement learning [9, 10], Deep learning [11, 12], or Deep reinforcement learning [12, 13] have been employed. Some studies have explored an adaptive control approach using Radial Basis Function Neural Network (RBF) with online learning capability to estimate unknown nonlinear components of the model, enhancing control performance [14, 15]. In studies on intelligent control methods [16], the use of Recurrent Fuzzy Wavelet Neural Network (RFWNN) has demonstrated powerful learning capabilities. Research on model-independent controllers based on machine learning techniques [9–13] can automatically adjust, improve control quality over time, and automatically find optimal control strategies.

Based on the analyses above, a backstepping sliding mode control algorithm integrated with neural network and fuzzy logic theory is proposed to control dual-arm manipulators, enabling them to operate efficiently in a short time, track trajectories, and maintain system stability. Furthermore, the proposed method can compensate for disturbances, allowing dual-arm manipulators to work and adapt to various types of disturbances affecting joint moments. This method stands out for its robustness due to the advantages of Sliding Mode Control (SMC) and simplicity in design using the recursive Lyapunov function of the Backstepping technique.

The remaining sections of the paper include: Section 2 presenting the construction of the mathematical model of the dual-arm robot, and discussing the adaptive Fuzzy Neural network Backstepping sliding mode algorithm, Section 3 outlining simulation steps and comments, and Section 4 providing conclusions.

## 2. Materials and methods

### 2. 1. Mathematical model of DAR

Considering the dual-arm robot (DAR) model with 2 degrees of freedom applied in the case of controlling an object with mass described in [3] as shown in **Fig. 1**. The structure of each robot arm consists of 2 revolute joints, where $m_i$, $I_i$, $L_i$, $k_i$ respectively representing the mass, moment of inertia, length of the link, and the distance from the center of mass of the link to a joint. Let's assume the load is securely held by the DAR, ensuring no sliding friction occurs between the load and the contact point with the robot. At the same time, let's define $d_1$ and $d_2$, respectively, the parameters representing the length of the load and the distance between the bases of the two robotic arms.

In this study, let's consider activities on a flat coordinate surface $xy$, when the load is securely attached by two robot arms. The object will move by controlling the trajectory grasped by the dual-arms robot so that the object is held firmly as shown in **Fig. 2**.

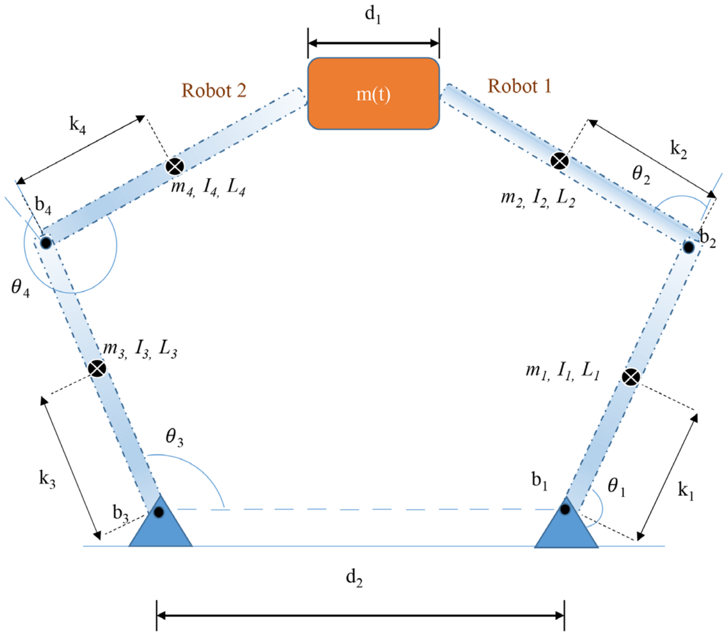



**Fig. 1.** The structure of a Dual-Arm Robot (DAR)

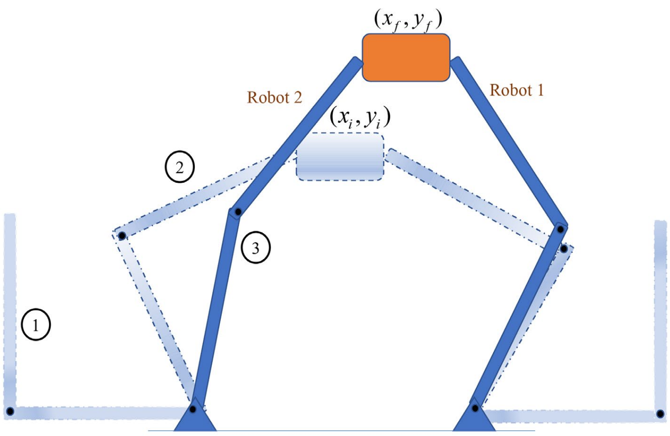



**Fig. 2.** Operational motions of dual-arm robot

Here, $x_m$, $y_m$ represent the coordinates at the center of the load:

$$\begin{cases} x_m = \dfrac{d_2}{2} + l_1\cos(q_1) + l_2\cos(q_1 + q_2) - \dfrac{d_1}{2} = d_2 + l_3\cos(q_3) + l_4 \cos(q_3 + q_4) + d_1, \\ y_m = l_1\sin(q_1) + l_2\sin(q_1 + q_2) = l_3\sin q_3 + l_4\sin(q_3 + q_4). \end{cases} \quad (1)$$

During operation, the robot applies forces $F_1$, $F_2$ to the object as shown in **Fig. 3**. To prevent the object from rotating around the $y$ and $z$ axes, friction forces $F_{s1y} = F_{s2y}$ and $F_{s1z} = F_{s2z}$ are introduced.

In this case, the dynamic equation of the object is written as follows:

$$m\ddot{x}_m = F_2 - F_1, \quad (2)$$

$$m\ddot{y}_m = 2F_{s1y} = F_{s2y}. \quad (3)$$

$$mg = 2F_{s1z} = 2F_{s2z}, \quad (4)$$

where $g = 9.8$ m/s$^2$.

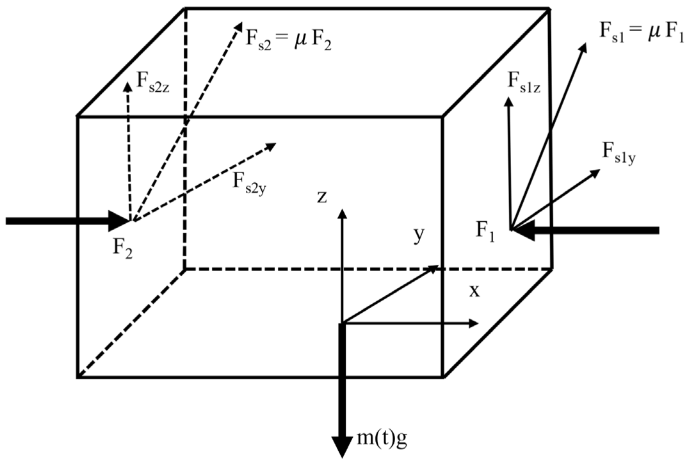



**Fig. 3.** The forces acting on the object

The relationship between the applied forces and the frictional forces is represented by:

$$F_{s1y}^2 + \left(\frac{mg}{2}\right)^2 < \left(\mu F_1\right)^2, \tag{5}$$

$$F_{s2y}^2 + \left(\frac{mg}{2}\right)^2 < \left(\mu F_2\right)^2. \tag{6}$$

When the direction of forces $F_1$ and $F_2$ always points towards the load, the load is effectively held by the two arms, and these forces must be positive. If $\ddot{x}_m(t) \geq 0$, then both applied forces $F_1$ and $F_2$ can be calculated:

$$\begin{cases} F_1 = \dfrac{1}{\mu}\sqrt{\left(\dfrac{m\ddot{y}_m}{2}\right)^2 + \left(\dfrac{mg}{2}\right)^2}, \\ F_2 = \dfrac{1}{\mu}\sqrt{\left(\dfrac{m\ddot{y}_m}{2}\right)^2 + \left(\dfrac{mg}{2}\right)^2} + m\ddot{x}. \end{cases} \tag{7}$$

In the case of $\ddot{x}_m(t) < 0$, these forces can be determined by:

$$\begin{cases} F_1 = \dfrac{1}{\mu}\sqrt{\left(\dfrac{m\ddot{y}_m}{2}\right)^2 + \left(\dfrac{mg}{2}\right)^2} - m\ddot{x}_m, \\ F_2 = \dfrac{1}{\mu}\sqrt{\left(\dfrac{m\ddot{y}_m}{2}\right)^2 + \left(\dfrac{mg}{2}\right)^2}. \end{cases} \tag{8}$$

Applying the Euler-Lagrange equation with the form $F_i = \partial/\partial t\ \left(\partial L/\partial \dot{q}_i\right) - \partial L/\partial q_i$, the dynamic model of a dual-arm robot controlling a load can be described as follows:

$$M(q)\ddot{q} + C(q,\dot{q}) + G(q) = \tau + J^T(q)F(q,\dot{q},\ddot{q}) - T_d - \beta, \tag{9}$$

where $\tau$ being a control input vector representing torque in a 4×1 dimension, $T_d$ being a 4×1 vector indicating the influence of noise on the robot's arms, and $\beta$ representing the viscous friction force on all joints, described as follows:

$$q = \left[q_1\ q_2\ q_3\ q_4\right]^T,$$

$$u = \left[u_1\ u_2\ u_3\ u_4\right]^T,$$

$$F=\left[F_1\ F_{s1y}\ F_2\ F_{s2y}\right]^T,$$

$$T_d=\ \left[T_{d1}\ T_{d2}\ T_{d3}\ T_{d4}\right]^T,$$

$$G(q)=\ \left[0\ 0\ 0\ 0\right]^T,$$

$$\beta=\left[b_1\ b_2\dot{q}_2\ b_3\dot{q}_3\ b_4\dot{q}_4\right]^T.$$

$M(q)$ is a 4×4 matrix of mass dynamics, and its components are determined by:

$$m_{11}=A_1+A_2+2A_3\cos q_2,$$

$$m_{22}=A_2,$$

$$m_{13}=m_{14}=m_{23}=\ m_{24}=0,$$

$$m_{33}=A_4+A_5+2A_6\cos q_4,$$

$$m_{34}=m_{43}=A_5+A_6\cos q_4,$$

$$m_{44}=A_5,$$

$$m_{31}=m_{32}=m_{41}=m_{42}=0.$$

With:

$$A_1=m_1k_1^2+m_2l_1^2+I_1,$$

$$A_2=m_2k_2^2+I_2,$$

$$A_3=m_2l_1k_2,$$

$$A_4=m_3k_3^2+m_4l_3^2+I_3,$$

$$A_5=m_4k_4^2+I_4,$$

$$A_6=m_4l_3k_4.$$

$C(q\ ,\dot{q})$ is a 4×1 Coriolis matrix, its components are determined by:

$$c_{11}=-A_3\ \sin q_2\left(\dot{q}_2^2+\dot{q}_1\dot{q}_2\right)+b_1\dot{q}_1,$$

$$c_{21}=A_3\dot{q}_1^2\sin q_2+b_2q_2,$$

$$c_{31}=-A_6\sin q_4\left(\dot{q}_4^2+\dot{q}_3\dot{q}_4\right)+b_3\dot{q}_3,$$

$$c_{41}=A_6\dot{q}_3^2\sin q_4+b_2\dot{q}_4.$$

Additionally, $J$ is a Jacobian matrix with a size of 4×4, and its components are calculated as follows:

$$J_{11}=-L_1\sin\ q_1-L_2\sin\left(q_1+q_2\right),$$

$$J_{12}=-L_1\cos\ q_1-L_2\cos\left(q_1+q_2\right),$$

$$J_{13} = J_{14} = 0,$$

$$J_{21} = -L_2 \sin(q_1 + q_2),$$

$$J_{22} = -L_2 \cos(q_1 + q_2),$$

$$J_{23} = J_{24} = 0,$$

$$J_{31} = J_{32} = 0,$$

$$J_{33} = L_3 \sin q_3 + L_4 \sin(q_3 + q_4),$$

$$J_{34} = -L_3 \cos q_3 - L_4 \cos(q_3 + q_4),$$

$$J_{41} = J_{42} = 0,$$

$$J_{43} = L_4 \sin(q_3 + q_4),$$

$$J_{44} = -L_4 \cos(q_3 + q_4).$$

**2. 2. Controller design**

To design a closed-loop control system for effectively controlling the motion trajectory of the DAR, it is firstly possible to implement the design of a nonlinear controller presented in [1] based on the backstepping algorithm and sliding mode control.

However, during operation, the parameters of the controller are uncertain, and simultaneously, the joints of the robot may be influenced by various unknown external forces. Therefore, the authors propose a method to design an adaptive controller with parameter adaptation using fuzzy logic and an RBF neural network to compensate all the disturbances acting on the controller. The proposed control structure is shown in **Fig. 4.**

First, the dynamic model (9) of the system is rewritten as follows: $x_1 = (q_1, q_2, q_3, q_4)^T$ is the state vector of the DAR model. Thus, the state equation is described by (10):

$$\begin{cases} \dot{x}_1 = x_2, \\ \dot{x}_2 = M^{-1}(q) \cdot \tau + M^{-1}(q) \cdot K(q, \dot{q}, \ddot{q}), \end{cases} \tag{10}$$

where $K(q,\dot{q},\ddot{q}) = J^T(q) F(q,\dot{q},\ddot{q}) - C(q,\dot{q}) - G(q) - \beta - T_d$.

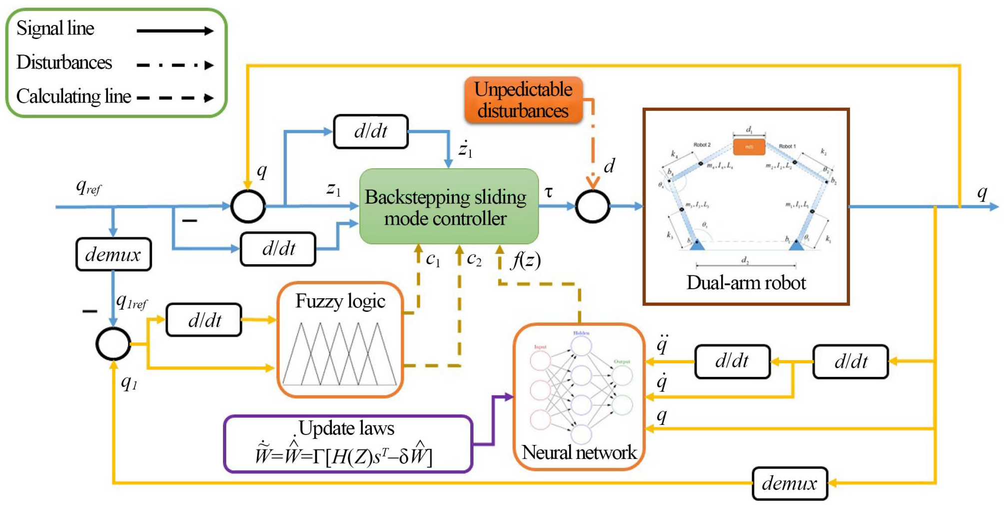



**Fig. 4.** General diagram of the adaptive fuzzy neural network SMC system

#### 2. 2. 1. Backstepping Sliding Mode Control (BSMC) for DAR

The ultimate goal in controlling a DAR is to track the motion of the end-effectors along specified trajectories. In other words, when designing a Backstepping Sliding Mode Controller (BSMC) to control the motion of the two arms along desired trajectories. $z_1 = x_1 - x_{1ref}$ is the error between the current joint angle value and its desired value. Where $x_{1ref} = q_{ref}$ as the desired joint angle value. The nonlinear controller design steps are as follows:

Step 1: consider α as the virtual control signal described as (11):

$$\alpha = -c_1 z_1 + \dot{x}_{1ref}. \tag{11}$$

With a positive constant $c_1$ for:

$$\lim_{t\to\infty} z_1(t) = 0.$$

If $z_2 = x_2 - \alpha$ and the time derivative of $z_1$ is taken:

$$\dot{z}_1 = \dot{x}_1 - \dot{x}_{1ref} = z_2 + \alpha - \dot{x}_{1ref} = z_2 - c_1 z_1. \tag{12}$$

The first Lyapunov function can be defined as:

$$V_1 = \frac{1}{2} z_1^T z_1. \tag{13}$$

So, the derivative of $V_1$ can be calculated as:

$$\dot{V}_1 = z_1^T \dot{z}_1 = -z_1^T c_1 z_1 + z_1^T z_2. \tag{14}$$

Step 2: to design the sliding control algorithm, the sliding surface can be formulated as follows:

$$s = \lambda z_1 + M z_2. \tag{15}$$

Where $\lambda = \mathrm{diag}(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$ is a diagonal matrix with positive diagonal coefficients. If to derivative $s$ with respect to time, the result will be:

$$\dot{s} = \lambda \dot{z}_1 + M\dot{z}_2 = \lambda \dot{z}_1 + M(\dot{x}_2 - \dot{\alpha}) = \lambda \dot{z}_1 + M\left(M^{-1}K + M^{-1}\tau - \dot{\alpha}\right) = \lambda \dot{z}_1 + K + \tau - M\dot{\alpha}. \tag{16}$$

The second Lyapunov function can be defined as:

$$V_2 = V_1 + \frac{1}{2} s^T s. \tag{17}$$

So, the derivative of $V_2$ can be calculated as:

$$\dot{V}_2 = \dot{V}_1 + s^T \dot{s} = -z_1^T c_1 z_1 + z_1^T z_2 + s^T(\lambda \dot{z}_1 + K + \tau - M\dot{\alpha}). \tag{18}$$

$\dot{V}_2$ can be rearranged by adding the signum function as follows:

$$\dot{V}_2 = -z_1^T c_1 z_1 - s^T c_2 \mathrm{sign}(s) + s^T\left(\frac{s z_1^T z_2}{s^T s} + c_2 \mathrm{sign}(s) + \lambda \dot{z}_1 + K + \tau - M\dot{\alpha}\right). \tag{19}$$

With a positive constant $c_2$. If the control input is selected as:

$$\tau = -\left(\frac{s z_1^T z_2}{s^T s} + c_2 \operatorname{sign}(s) + \lambda \dot{z}_1 + K - M\dot{\alpha}\right). \tag{20}$$

So,

$$\dot{V}_2 = -z_1^T c_1 z_1 - s^T c_2 \operatorname{sign}(s) < 0.$$

As a result, the error between the output value and the desired value tends to approach zero. Let's note that when the sliding surface $e_s \to 0$, $\tau \to -\infty$. Therefore, in practice, the control input for the BSMC is proposed as follows:

$$\tau = -\left(\frac{s z_1^T z_2}{s^T s + \sigma} + c_2 \operatorname{sign}(s) + \lambda \dot{z}_1 + K - M\dot{\alpha}\right), \tag{21}$$

where $\sigma$ is a very small positive constant.

#### 2. 2. 2. Design a fuzzy logic modifier for updating model parameters

In this study, it is possible to add an adaptive fuzzy logic modifier to the BSMC. Let's use Mamdani fuzzy logic structure. The inputs of fuzzy logic modifier are the error between the actual and desired values of joint angle 1, as well as their derivatives. Through this, it adjusts two controller parameters.

The definition and fuzzification of the input and output fuzzy sets are represented in **Fig. 5, 6**, respectively. The fuzzy inference table is constructed in **Table 1**. This directly influences the effectiveness of the fuzzy logic algorithm and the stability of the system under various levels of noise.

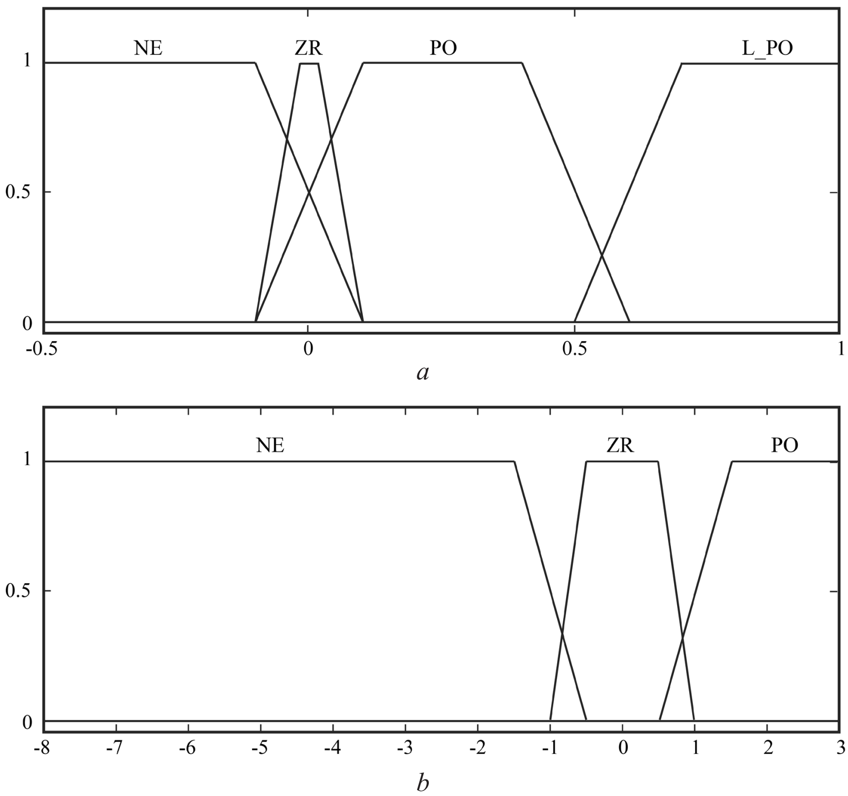
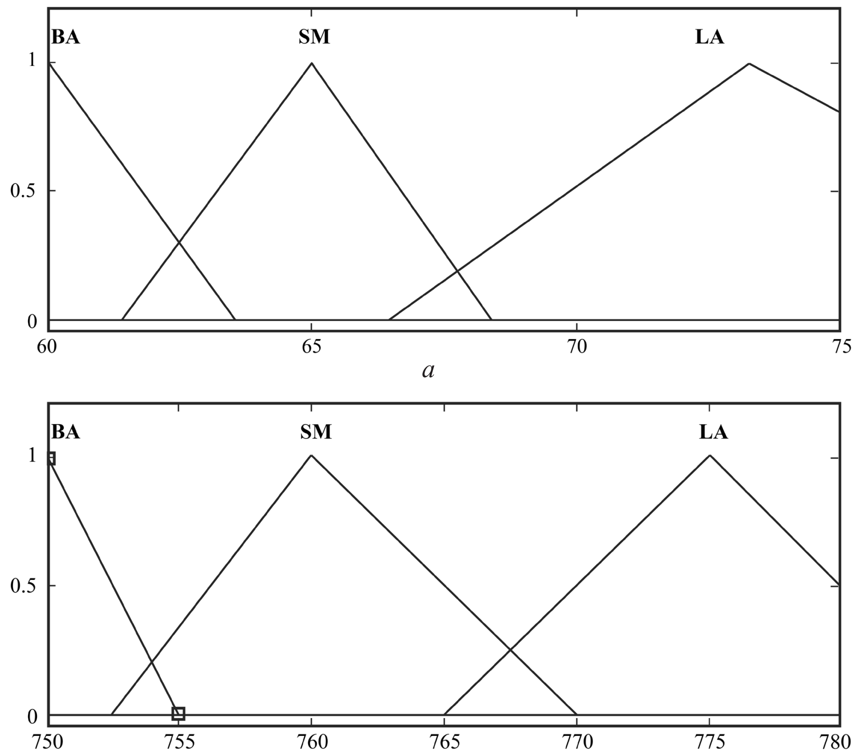



**Fig. 5.** Membership function: $a - e_{\theta_1}$; $b - \dot{e}_{\theta_1}$




*b*

**Fig. 6.** Membership function: $a - c_1$; $b - c_2$

**Table 1**
Fuzzy rule table

| ***e*** | | $\boldsymbol{e}_{\theta_1}$ | | | |
|---|---|---|---|---|---|
| | | ***NE*** | ***ZR*** | ***PO*** | ***LPO*** |
| $\dot{\boldsymbol{e}}_{\theta_1}$ | *NE* | *LA* | *SM* | *SM* | *SM* |
| | *ZR* | *LA* | *BA* | *LA* | *LA* |
| | *PO* | *SM* | *SM* | *LA* | *LA* |

**2. 2. 3. Design a neural network estimator and system stability**

As mentioned earlier, the intricate nonlinear dynamics represented by $K$ in equation (21) lack a complete analytical model. Thus, in this study, let's suggest utilizing a neural network with radial basis function to provide an approximate estimation of the unknown dynamic parameters.

Let $f(Z)$: $R^a \rightarrow R^b$ represent the radial basis function network:

$$f(Z) = W^T H(Z), \tag{22}$$

where $W = [W_1, W_2, \ldots, W_l]^T \in R^{bxl}$ is the ideal weight matrix, and $l$ is the number of neurons in a hidden layer. $H(Z) = [h_1(Z), h_2(Z), \ldots, h_l(Z)]^T$, where $h_i(Z)$ is an activation function. The activation function is described in detail in [14]:

$$h_i(Z) = \frac{\exp\left[\dfrac{-\|Z-\mu_i\|^2 + \|\dot{Z}-\mu_i\|^2}{\eta_i^2}\right]}{\sum_{j=1}^{l} \exp\left[\dfrac{-\|Z-\mu_j\|^2 + \|\dot{Z}-\mu_j\|^2}{\eta_j^2}\right]}, \tag{23}$$

and

$$0 < h_l(Z) \le 1, \tag{24}$$

where $\mu_i = [\mu_{i1}, \mu_{i2}, \ldots, \mu_{ia}]^T$ is the center vector of the receptive field, and $\eta_i$ is the width of the Gaussian function. $\ddot{e}$ is a matrix of the neural inputs. In this design, let's define:

$$Z = \left[x_1^T; \dot{x}_1^T; \ddot{x}_1^T\right]^T \in \Omega_Z \subset R^{12}. \tag{25}$$

If $\hat{W}$ denotes estimation of the weight matrix, the output of the radial basis function $f(Z)$ is approximated by:

$$\tau = -\left(\frac{s{z_1}^T z_2}{s^T s + \sigma} + c_2 \text{sign}(s) + \lambda \dot{z}_1 + \hat{f}(Z) - M\dot{\alpha}\right). \tag{26}$$

It is noted that from now onward it is possible to define the control approach with a control input presented in (26) as the radial basis function network based backstepping sliding mode control (RBFN-BSMC). Now let's check the stability of proposed approach with the Lyapunov function which is formulated by the equation (27):

$$V_2 = V_1 + \frac{1}{2}s^T s + \frac{1}{2}\sum_{i=1}^{4} \tilde{W}_i^T \Gamma^{-1} \tilde{W}_i, \tag{27}$$

where $\Gamma = \text{diag}(\Gamma_1, \Gamma_2, \ldots, \Gamma_4)$ is a positive definite diagonal matrix of the adaptation gains. $\tilde{W} = \hat{W} - W$ is error between the estimated weights $\widehat{W}$ and the ideal weights $W$. Then, derivative of $V_2$ can be computed by:

$$\dot{V}_2 = \dot{V}_1 + s^T \dot{s} + \sum_{i=1}^{4} \tilde{W}_i^T \Gamma^{-1} \tilde{W}_i = -{z_1}^T c_1 z_1 + {z_1}^T z_2 + s^T\left(\lambda \dot{z}_1 + K + \tau - M\dot{\alpha}\right) + \sum_{i=1}^{4} \tilde{W}_i^T \Gamma^{-1} \dot{\tilde{W}}_i. \tag{28}$$

Substituting the control law in (26) into (28) leads to:

$$\dot{V}_2 = -{z_1}^T c_1 z_1 - s^T c_2 \text{sign}(s) + \sum_{i=1}^{4} \tilde{W}_i^T \left[\Gamma^{-1} \dot{\tilde{W}}_i - s^T H(z) - \delta \widehat{W}\right], \tag{29}$$

where σ is a positive number. If the adaptation mechanism is chosen by:

$$\dot{\tilde{W}} = \dot{\widehat{W}} = \Gamma\left[H(Z)s^T - \delta \widehat{W}\right]. \tag{30}$$

When the derivative of $V_2$ can be rewritten as:

$$\dot{V}_2 = -{z_1}^T c_1 z_1 - s^T c_2 \text{sign}(s) < 0. \tag{31}$$

In other words, the system stability holds if the estimated weights $\hat{W}$ are adaptively computed by (30).

## 3. Results and discussion

### 3. 1. Trajectory and model parameters

To evaluate the proposed method, it is possible to perform the simulation under Matlab/Simulink software environment. In simulation, the manipulators initially follow reference trajectories to reach the payload. The reference trajectories for the first 2 seconds are specified by:

$$x_{a1}(t) = x_{f1} + \left(x_{i1} - x_{f1}\right)e^{-10t^2}, \tag{32}$$

$$y_{a1}(t)=y_{f1}+(y_{i1}-y_{f1})e^{-10t^2}, \tag{33}$$

$$x_{a2}(t)=x_{f2}+(x_{i2}-x_{f2})e^{-10t^2}, \tag{34}$$

$$y_{a2}(t)=y_{f2}+(y_{i2}-y_{f2})e^{-10t^2}, \tag{35}$$

where $x_{a1}$, $y_{a1}$, $x_{a2}$, $y_{a2}$ are the trajectories of the manipulators. ($x_{i1}$, $y_{i1}$, $x_{i2}$, $y_{i2}$) and ($x_{f1}$, $y_{f1}$, $x_{f2}$, $y_{f2}$) are the initial and final positions of the end-effectors, respectively. Upon securely holding the object, the robot moves the payload along half of a circle to prevent collisions with obstacles. The anticipated path for the center of the object is described by the following curve,

$$x_{mr}(t)=x_0+r_m\cos(\varphi t), \tag{36}$$

$$y_{mr}(t)=y_0+r_m\sin(\varphi t), \tag{37}$$

where ($x_0$, $y_0$) is the position of the obstacle, this curve represents the center of the circle along which the object is in motion. $r_m$ denotes the radius of the circle, and θ is a polar angle ranging from $-\pi$ to 0. It is essential to highlight that the joint angles between the link and the base, or its preceding link, were initially known at the starting point $t=0$, $q_1(0)=\pi/6$, $q_2(0)=\pi/2$, $q_3(0)=\pi$ and $q_4(0)=-2\pi/3$.

Proceeding with simulations based on two cases as follows:

Case 1: all parameters of a dual-arm robot are accurately determined. Concurrently, the system is not affected by uncertain disturbances from the environment.

Case 2: the dual-arm robot will face a huge challenge when the model is subjected to substantial disturbances, which change over time and directly impact all four joint moments of the system. The model parameters used in this case are provided in **Table 2** and the trajectory parameters are shown in **Table 3.**

**Table 2**
Parameters of the dual arm robot system [3]

| Parameters | Symbols | Values |
|---|---|---|
| Moment of inertia | $I_i$ ($i=1, 2, 3, 4$) | 0.18 kgm$^2$ |
| Length of the link | $L_i$ ($i=1, 2, 3, 4$) | 1.2 m |
| The distance from the center of the link to a joint | $k_i$ ($i=1, 2, 3, 4$) | 0.48 m |
| Friction coefficient | $b_i$ ($i=1, 2, 3, 4$) | 110 Nm/s |
| The length of the object | $d_1$ | 0.25 m |
| The distance between the bases of the two arms | $d_2$ | 1.2 m |
| Confidence | μ | 0.35 |

**Table 3**
Parameter of reference trajectory

| Parameter | Symbols | Values |
|---|---|---|
| Initial coordinates of the arm | ($x_{i1}$; $y_{i1}$; $x_{i2}$; $y_{i2}$) | (0.76; 0.6; −0.76; 0.6) |
| Desired value of the arm | ($x_{f1}$; $y_{f1}$; $x_{f2}$; $y_{f2}$) | (−0.275; 1.4; −0.525; 1.4) |
| Center of orbit | ($x_0$; $y_0$) | (0; 1.4) |
| Orbit radius | $r_m$ | 0.4 m |
| Initial values of the angles | $q(0)$ | $(\pi/6;\pi/2;\pi;-2\pi/3)$ |
| Initial angular velocity | $\dot{q}(0)$ | (0; 0; 0; 0) |

*Note: in simulation results shown in 3.2 and 3.3, we use «ABSP-SMC» to illustrate for controller system with adaptive fuzzy neural network backstepping sliding mode control, and «BSP-SMC» is only non-linear backstepping sliding mode control*

**3. 2. Case 1: parameters of DAR model are constant and the system is not affected by disturbances**

In this case, both the fuzzy neural network backstepping sliding mode controller and the nonlinear controller exhibit excellent control performance. The simulation results depicted in **Fig. 7** show that the proposed method achieves fast convergence times, approximately 0.25 seconds for joints 1, 2, and 4, while the convergence time for joint 3 is nearly instant. The system stability is maintained throughout the entire operating time. This outcome is similarly observed with the nonlinear Backstepping sliding mode control. Thus, it is evident that the adaptive Backstepping sliding mode controller based on fuzzy logic and neural network not only provides excellent control performance and rapid response times but also ensures stability throughout the entire operation.

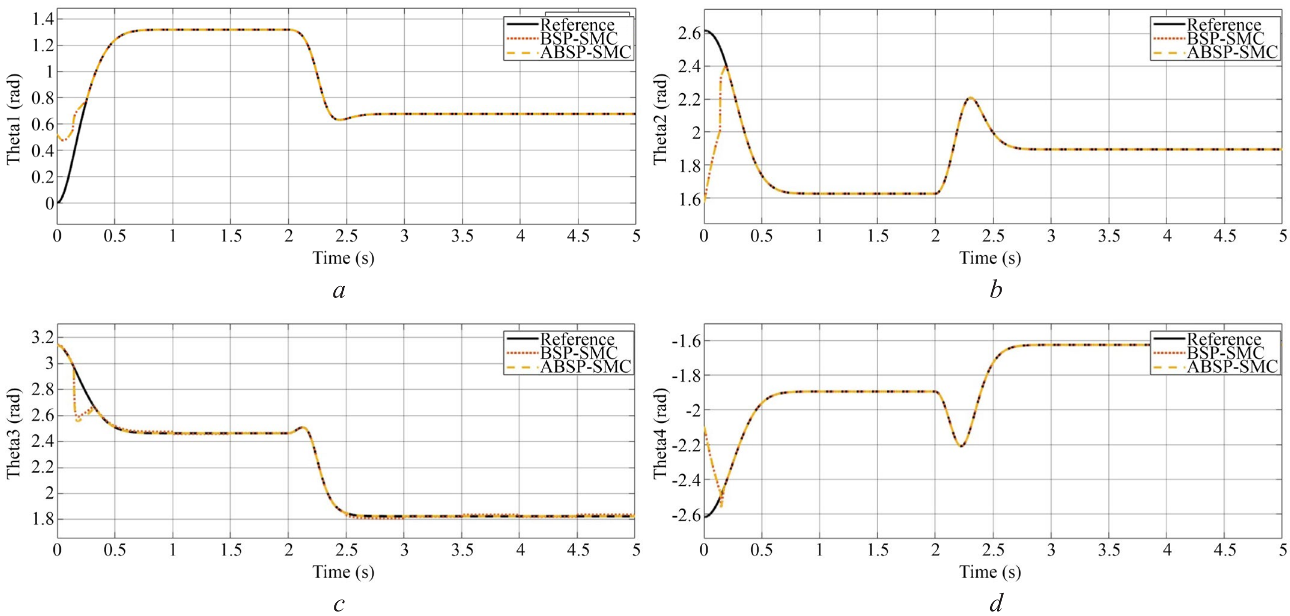



**Fig. 7.** Joint angle trajectory response in the absence of disturbing influences: *a* – Joint angle 1, $q_1$; *b* – Joint angle 2, $q_2$; *c* – Joint angle 3, $q_3$; *d* – Joint angle 4, $q_4$

**3. 3. Case 2: the dual-arm robot model has unstable model parameters, and uncertain disturbances affect the system**

In this case, the system is influenced by uncertain noise signals from the environment as depicted in **Fig. 8** and also gets the sudden change in mass as shown in **Fig. 9**.

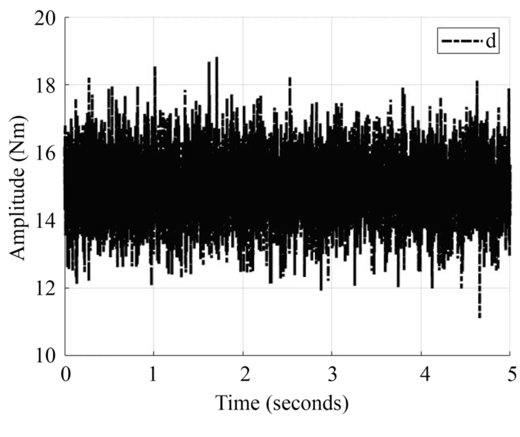



**Fig. 8.** External disturbance affects the dual-arm robot system

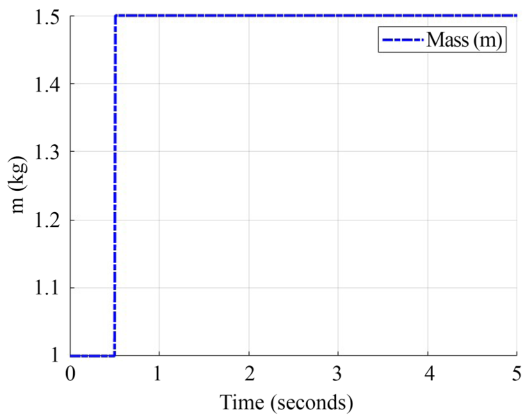
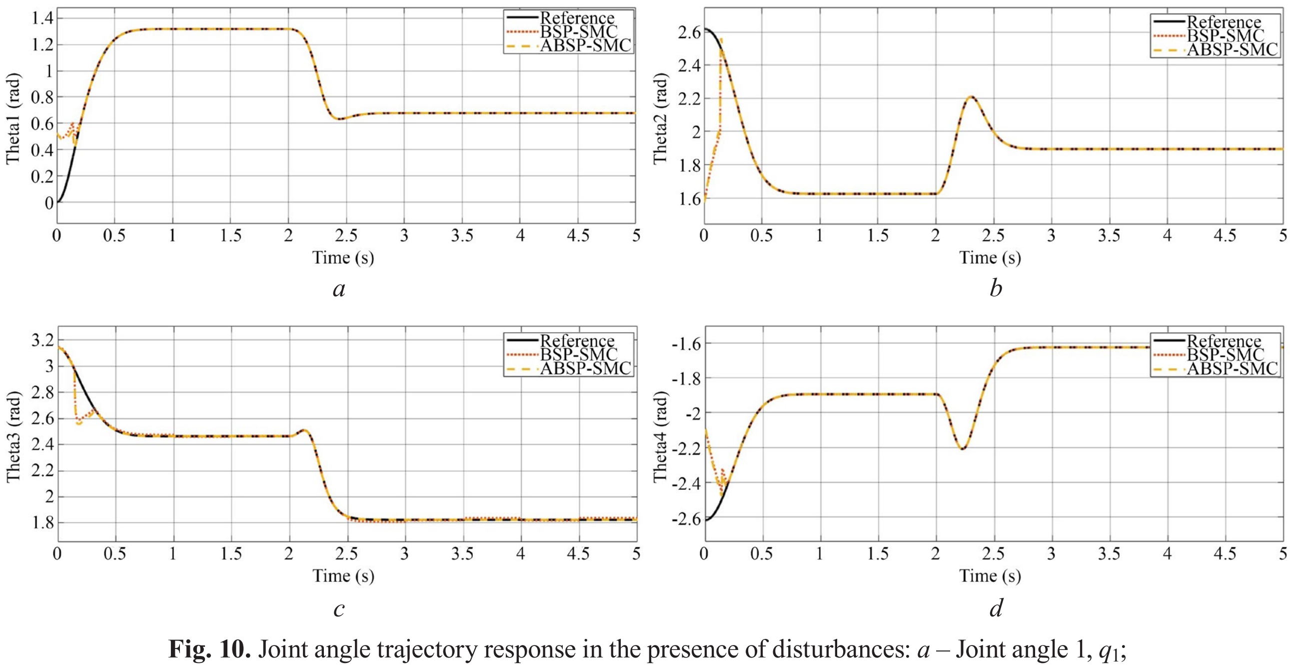



**Fig. 9.** Change of mass

From the simulation results, it is possible to see in **Fig. 10** that the proposed method demonstrates excellent control quality when faced with strong disturbances affecting joint moments and variations in model parameters, as compared to the nonlinear controller**.** In detail, the system shows the shorter convergence time at joint 1 compared to the BSMC controller, approximately 0.1 second; with joint 2, the convergence time of the disturbances one is nearly 0.2 second approaching the BSMC controller then becomes stable. The difference is observed at joint 3, where the nonlinear controller shows a slight deviation during operation, which does not occur with the adaptive controller designed. Considering joint 4, it is shown that the difference between BSMC controller and the ones with disturbance signal at joint 4 is approximately 0.3 second to be stable.

It also can be seen that the control quality of the adaptive fuzzy neural network sliding mode control is excellent at 0.25 seconds, similar to the convergence time in Case 1, while addressing the drawbacks of the sliding mode control, where reluctance and errors persist due to continuous influences from uncertain disturbances on the joint moments of the system.

The high stability performance of closed-loop control systems forms the basis for practical applications. When a dual-arm robot is capable of meeting job requirements almost instantly, it enhances its practical utility.




*a* *b* *c* *d*

**Fig. 10.** Joint angle trajectory response in the presence of disturbances: *a* – Joint angle 1, $q_1$; *b* – Joint angle 2, $q_2$; *c* – Joint angle 3; $q_3$; *d* – Joint angle 4, $q_{4sz}$

**3. 4. Limitations and directions for development of the study**

To apply this algorithm in real-world conditions, it is necessary to face the challenges about mechanics, ensuring the accuracy of model parameters. Actual DAR model errors will lead to a change in the parameter set of the nonlinear controller, which in turn requires re-adjusting the parameters. Additionally, external disturbances in the environment can affect the practical performance of the model during operation. However, reproducing this method and applying it to real-world scenarios for this robot is absolute possible.

In the future, it is possible to implement and test this algorithm on an experimental system and continue researching and developing this algorithm for various types of industrial robots currently available.

## 4. Conclusions

The study proposed adaptive backstepping sliding mode control algorithm combined with neural network and fuzzy logic theory for controlling the DARs, tracking different trajectories in a short time, maintain system stability, and overcome the limitations of sliding mode control when the system is influenced by unknown disturbances and variations in model parameters. Simulation results demonstrate that the control performance of the system is excellent. The motion of the DARs absolutely follow the desired trajectory and maintaining stability with a rapid establishment time of approximately 0.25 seconds for all 4 joints. With the help of Fuzzy Logic and Neural Network as shown above, the results of 4 joints with and without disturbances become nearly unchanged, if not faster, to compare the ones with disturbances to the ones without it. Clearly, even the ones with disturbance might be reluctant a little from the beginning, it still approaches the original trajectory faster than normal Backstepping for about 0.02 to 0.04 second.

**Conflict of interest**

The authors declare that they have no conflict of interest in relation to this research, whether financial, personal, authorship or otherwise, that could affect the research and its results presented in this paper.

**Financing**

This research if funded by International School, Vietnam National University, Hanoi (VNU-IS) under project number CS.2023-06.

**Data availability**

The manuscript has no associated data.

**Use of artificial intelligence**

The authors confirm that they did not use artificial intelligence technologies when creating the current work.

**References**

[1] Yagiz, N., Hacioglu, Y., Arslan, Y. Z. (2010). Load transportation by dual arm robot using sliding mode control. Journal of Mechanical Science and Technology, 24 (5), 1177–1184. https://doi.org/10.1007/s12206-010-0312-9

[2] Bai, K., Jiang, G., Jiang, G., Liu, Z. (2019). Based on fuzzy-approximation adaptive backstepping control method for dual-arm of humanoid robot with trajectory tracking. International Journal of Advanced Robotic Systems, 16 (3), 172988141983190. https://doi.org/10.1177/1729881419831904

[3] Nguyen, T. V., Thai, N. H., Pham, H. T., Phan, T. A., Nguyen, L., Le, H. X., Nguyen, H. D. (2019). Adaptive Neural Network-Based Backstepping Sliding Mode Control Approach for Dual-Arm Robots. Journal of Control, Automation and Electrical Systems, 30 (4), 512–521. https://doi.org/10.1007/s40313-019-00472-z

[4] Wang, J., Xu, F., Lu, G. (2017). A model weighted adaptive neural backstepping sliding-mode controller for cooperative manipulator system. International Journal of Advanced Robotic Systems, 14 (6), 172988141774302. https://doi.org/10.1177/1729881417743028

[5] Bai, K., Gong, X., Chen, S., Wang, Y., Liu, Z. (2018). Sliding mode nonlinear disturbance observer-based adaptive back-stepping control of a humanoid robotic dual manipulator. Robotica, 36 (11), 1728–1742. https://doi.org/10.1017/s026357471800067x

[6] Huang, Y., Cheng, L., Li, Z., Gao, W., Lu, H., Wei, L. (2019). Backstepping Sliding Mode Control for Robot Manipulator via Nonlinear Disturbance Observer. 2019 Chinese Control Conference (CCC). https://doi.org/10.23919/chicc.2019.8865316

[7] Santos, J. C., Gouttefarde, M., Chemori, A. (2022). A Nonlinear Model Predictive Control for the Position Tracking of Cable-Driven Parallel Robots. IEEE Transactions on Robotics, 38 (4), 2597–2616. https://doi.org/10.1109/tro.2022.3152705

[8] Gold, T., Völz, A., Graichen, K. (2020). Model Predictive Interaction Control for Industrial Robots. IFAC-PapersOnLine, 53 (2), 9891–9898. https://doi.org/10.1016/j.ifacol.2020.12.2696

[9] Hu, Y., Wang, W., Liu, H., Liu, L. (2020). Reinforcement Learning Tracking Control for Robotic Manipulator With Kernel-Based Dynamic Model. IEEE Transactions on Neural Networks and Learning Systems, 31 (9), 3570–3578. https://doi.org/10.1109/tnnls.2019.2945019

[10] Trung, D. N., Van, T. N., Le, H. X., Manh, D. D., Hoang, D. (2023). Reinforcement Control for Planar Robot Based on Neural Network and Extended State Observer. Lecture Notes in Networks and Systems, 516–525. https://doi.org/10.1007/978-981-99-4725-6_62

[11] Chen, X., Guhl, J. (2018). Industrial Robot Control with Object Recognition based on Deep Learning. Procedia CIRP, 76, 149–154. https://doi.org/10.1016/j.procir.2018.01.021

[12] Liu, R., Nageotte, F., Zanne, P., de Mathelin, M., Dresp-Langley, B. (2021). Deep Reinforcement Learning for the Control of Robotic Manipulation: A Focussed Mini-Review. Robotics, 10 (1), 22. https://doi.org/10.3390/robotics10010022

[13] Zhang, X., Ji, Z., Zhang, H., Xiong, R. (2023). A Deep Reinforcement Learning Control Method for a Four-Link Brachiation Robot. 2023 2nd International Conference on Machine Learning, Cloud Computing and Intelligent Mining (MLCCIM). https://doi.org/10.1109/mlccim60412.2023.00085

[14] Dang, S. T., Dinh, X. M., Kim, T. D., Xuan, H. L., Ha, M.-H. (2023). Adaptive Backstepping Hierarchical Sliding Mode Control for 3-Wheeled Mobile Robots Based on RBF Neural Networks. Electronics, 12 (11), 2345. https://doi.org/10.3390/electronics12112345

[15] Tian, Z., Zhou, Y. (2021). Adaptive dynamic surface control of UAV based on RBF neural network. 2021 40th Chinese Control Conference (CCC). https://doi.org/10.23919/ccc52363.2021.9550746

[16] Yen, V. T., Nan, W. Y., Van Cuong, P. (2018). Recurrent fuzzy wavelet neural networks based on robust adaptive sliding mode control for industrial robot manipulators. Neural Computing and Applications, 31 (11), 6945–6958. https://doi.org/10.1007/s00521-018-3520-3

*Received date 06.01.2024*
*Accepted date 17.03.2024*
*Published date 29.03.2024*



***How to cite:*** *Nguyen, N. K., Bui, Q. V. B., Nguyen, Q. H., Luong, T. G., Dang, X. D., Le, H. X., Nguyen, L. N. (2024). Adaptive fuzzy-neural network effectively disturbance compensate in sliding mode control for dual arm robot. EUREKA: Physics and Engineering, 2, 79–93. https://doi.org/10.21303/2461-4262.2024.003250*