Robust sliding mode control based on a new Quasi-sliding mode and adaptive artificial neural networks observer for robot

Phan Nhut Tan, Huynh Dac Son Tien, Pham Thanh Tung*

Vinh Long University of Technology Education, 73 Nguyen Hue, Long Chau, Vinh Long, Vietnam.
*Corresponding author: tungpt@vlute.edu.vn
Received 4 Sep. 2025; Revised 5 Dec. 2025; Accepted 10 Dec. 2025; Published 25 Dec. 2025.
DOI: https://doi.org/10.54939/1859-1043.j.mst.108.2025.21-30

ABSTRACT

This study designs and evaluates the simulation results of a robust sliding mode control based on a new Quasi–sliding mode and adaptive radial basis function neural network (RBFNN) observer applied to single-link robot control. An industrial robot (robot manipulator) is a multifunctional manipulator that can be programmed to perform dangerous and/or repetitive tasks with high precision. The robust adaptive RBFNN neural network observer is used to estimate the states and nonlinear functions in the mathematical description of the robot. The sliding mode controller based on a new Quasi–sliding mode combines with the robust adaptive RBFNN observer for the robot trajectory tracking control with appropriate quality indicators. The weights of the RBFNN are updated online. The stability of the proposed control methods is proven by Lyapunov stability theory. The simulation results in MATLAB/Simulink have shown the effectiveness and sustainability of the proposed method without the steady state error, the rising time achieves 0.4656(s), the settling time is 0.7690(s), the overshoot is 0(%), the values of RMSE (Root Mean Squared Error), IAE (Integral Absolute Error) and ISE (Integral Square Error) are 1.7549e-06, 0.0222 và 0.001124, respectively.

Keywords: Robot; RBF neural network; Observer; Adaptive; Sliding mode control.

1. INTRODUCTION

Robotic manipulators have become increasingly important in modern industries, gradually replacing human labor in hazardous, repetitive, and complex tasks to ensure accuracy, speed, and cost-effectiveness [1, 2]. Robots are capable of autonomously moving and manipulating objects, and technological advancements have significantly enhanced their capabilities, especially through the development of high-performance controllers [3].

In recent years, extensive research has been conducted on robotic control systems. For instance, the dynamic modeling of single-link and R-P manipulators has been studied [1, 2] a review of the published works from 2010 to 2020 was conducted, The application of fractional-order concepts in modeling and control techniques for various robotic manipulators has been comprehensively evaluated [3], a novel controller based on composite control theory was developed to control a single-link robotic arm actuated by pneumatic artificial muscles (PAMs) in an antagonistic biceps/triceps configuration [4]. Trajectory tracking control method for robotic manipulators [5], an adaptive robust impedance controller when interacting with the environment at an unknown intermediate contact point [6], focusing on the dynamic modeling and optimal controller design based on a genetic algorithm (GA) for a robot with rigid prismatic joints [7], studies on the development of adaptive neuro-fuzzy systems [8] to control the position and velocity of the robotic manipulator, deriving the equations of motion for a manipulator system with translational and rotational movements and developing computational codes using the finite element method [9], evaluating the effect of link flexibility on the end-effector position of a robotic arm during a specific motion [10], Recurrent Neural Network (RNN) and Differential Evolution Optimization (DEO) based on Nonlinear Model Predictive Control (NMPC) technique [11], sliding mode control for trajectory tracking at the robot end-effector by properly designing a reference trajectory for the joint angle [12], output feedback control of a single-link robotic manipulator under matched disturbances and parametric uncertainties [13], Radial Basis Function Neural Network Sliding

Mode Control for Ship Path Following Based on Position Prediction [14], Sliding Mode Control with Chattering Attenuation and Hardware Constraints in Spacecraft Applications [15], Adaptive Control Based on Radial Base Function Neural Network Approximation for Quadrotor [16].

The objective of this study is to design and evaluate the simulation results of a robust sliding mode controller based on a new Quasi-sliding mode and an adaptive artificial neural network observer for robotic systems. In the adaptive radial basis function neural network (RBFNN) observer, the RBFNN is employed to estimate the states x_1, x_2, $f(\boldsymbol{x})$ và $g(\boldsymbol{x})$ in the mathematical model of the robot. In the proposed robust sliding mode controller, a new Quasi-sliding mode (also called hyperbolic tangent (tanh) function) combined with an exponential reaching law and an adaptive RBF neural network observer is utilized to achieve trajectory tracking control of the robot, ensuring satisfactory performance indices. With the proposed control method, speedless adaptive control can be implemented without model information. The weights of the neural network are adjusted online. No precise knowledge of nonlinearities in the observed system is required. The stability of the proposed control methods is verified using Lyapunov stability theory.

The content of this paper is organized into four sections: Section 2 presents the design method of the sliding mode controller based on a new Quasi-sliding mode and an adaptive artificial neural network observer for the robot; Section 3 discusses the simulation results and analysis; and section 4 provides the conclusion.

2. CONTROLLER DESIGN

2.1. Mathematical model of the robot

The single-link robot model considered in this study is described by (1) [17, 18]:

$$\begin{cases} J\dot{x}_2 + \dfrac{1}{2}mgl\sin(x_1) = u \\ y = x_1 \end{cases} \tag{1}$$

The state-space representation of the robot is given as (2):

$$\begin{cases} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix}\left(-\dfrac{1}{2}\dfrac{mgl\sin(x_1)}{J} + \dfrac{1}{J}u \right) \\ y = x_1 \end{cases} \tag{2}$$

where, $x_1 = q$ is the angular position, $x_2 = \dot{q}$ is the angular velocity of the robot, u is the control input torque, J is the moment of inertia, g is the gravitational acceleration, m is the mass and l is the length of the link. Assume $f(\boldsymbol{x}) = -\dfrac{1}{2}\dfrac{mgl\sin(x_1)}{J}$, $g(\boldsymbol{x}) = \dfrac{1}{J}$, the mathematical model of the robot with disturbances can be represented as (3):

$$\begin{cases} \dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}\left(f(\boldsymbol{x}) + g(\boldsymbol{x})u + d(t) \right) \\ y = \boldsymbol{C}^T\boldsymbol{x} \end{cases} \tag{3}$$

where, $\boldsymbol{x} = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T$, $\boldsymbol{A} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$, $\boldsymbol{b} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, $\boldsymbol{C} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $d(t) = 0$, $|d(t)| \le b_d$, $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$ are unknown nonlinear functions.

2.2. Analysis and design of an adaptive artificial neural network observer

This study employs a radial basis function neural network (RBFNN) to estimate the unknown nonlinear functions $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$ in (3). The RBFNN observer [14] for the robot system (3) is given as in (4):

$$\begin{cases} \dot{\hat{\boldsymbol{x}}} = \boldsymbol{A}\hat{\boldsymbol{x}} + \boldsymbol{b}\left(\hat{f}\left(\hat{\boldsymbol{x}}\right) + \hat{g}\left(\hat{\boldsymbol{x}}\right)u + v(t)\right) + \boldsymbol{K}\left(y - \boldsymbol{C}^T\hat{\boldsymbol{x}}\right) \\ \hat{y} = \boldsymbol{C}^T\hat{\boldsymbol{x}} \end{cases} \tag{4}$$

where, $\hat{\boldsymbol{x}}$ is the estimated value of $\boldsymbol{x}$, $\boldsymbol{K}$ is the gain vector, $\boldsymbol{K} = \begin{bmatrix} k_1 & k_2 \end{bmatrix}^T$, $\hat{f}\left(\hat{\boldsymbol{x}}\right)$ and $\hat{g}\left(\hat{\boldsymbol{x}}\right)$ is the estimation of $f\left(\boldsymbol{x}\right)$ and $g\left(\boldsymbol{x}\right)$, $v(t)$ is the robust term.

The unknown continuous nonlinear functions in the system can be approximated by an RBFNN with ideal constant weights $\boldsymbol{W}^*$ and a sufficient number of basis functions $\boldsymbol{h}\left(x\right)$ as shown in (5):

$$\begin{aligned} f\left(\boldsymbol{x}\right) &= \boldsymbol{W}_1^{*T}\boldsymbol{h}_1\left(\boldsymbol{x}\right) + \varepsilon_1\left(\boldsymbol{x}\right),\ \varepsilon_1\left(\boldsymbol{x}\right) \le \varepsilon_{1,N} \\ g\left(\boldsymbol{x}\right) &= \boldsymbol{W}_2^{*T}\boldsymbol{h}_2\left(\boldsymbol{x}\right) + \varepsilon_2\left(\boldsymbol{x}\right),\ \varepsilon_2\left(\boldsymbol{x}\right) \le \varepsilon_{2,N} \end{aligned} \tag{5}$$

where, $\varepsilon_1\left(\boldsymbol{x}\right)$ and $\varepsilon_2\left(\boldsymbol{x}\right)$ are the neural network reconstruction errors.

Assume that the desired weights $\boldsymbol{W}_1^*$ and $\boldsymbol{W}_2^*$ are bounded by known values:

$$\left\|\boldsymbol{W}^*\right\|_{i,F} \le \boldsymbol{W}_{i,M},\quad i = 1,2 \tag{6}$$

The estimated outputs of $f\left(\boldsymbol{x}\right)$ and $g\left(\boldsymbol{x}\right)$ using the RBFNN are given by (7):

$$\begin{aligned} \hat{f}\left(\hat{\boldsymbol{x}}\right) &= \hat{\boldsymbol{W}}_1^T\boldsymbol{h}_1\left(\hat{\boldsymbol{x}}\right) \\ \hat{g}\left(\hat{\boldsymbol{x}}\right) &= \hat{\boldsymbol{W}}_2^T\boldsymbol{h}_2\left(\hat{\boldsymbol{x}}\right) \end{aligned} \tag{7}$$

where, $\hat{\boldsymbol{W}}_1$ và $\hat{\boldsymbol{W}}_2$ is the estimated weight value, $\tilde{\boldsymbol{W}}_i = \boldsymbol{W}_i^* - \hat{\boldsymbol{W}}_i\ \left(i = 1,2\right)$.

Assume that the control input $u\left(t\right)$ is bounded by a positive constant $\left|u\left(t\right)\right| \le u_d$. The observer is designed as in (8):

$$\begin{cases} \dot{\hat{\boldsymbol{x}}} = \boldsymbol{A}\hat{\boldsymbol{x}} + \boldsymbol{b}\left(\hat{\boldsymbol{W}}_1^T\hat{\boldsymbol{h}}_1 + \hat{\boldsymbol{W}}_2^T\hat{\boldsymbol{h}}_2 u - v_1 - v_2\right) + \boldsymbol{K}\left(y - \boldsymbol{C}\hat{\boldsymbol{x}}\right) \\ \hat{y} = \boldsymbol{C}^T\hat{\boldsymbol{x}} \end{cases} \tag{8}$$

where, the robust term is given by (9):

$$v_i\left(t\right) = -D_i\frac{\tilde{y}}{\left|\tilde{y}\right|},\ i = 1,2 \tag{9}$$

with $D_1 \ge \beta_1\delta_M$, $D_2 \ge \beta_2\delta_M u_d$, $\delta_M = \delta_{\max}\left[L^{-1}\left(s\right)\right]$, $\delta_{\max}\left[.\right]$ is the maximum singular value.

The adaptation law of the RBFNN is designed as in (10):

$$\begin{aligned} \dot{\hat{\boldsymbol{W}}}_1 &= \boldsymbol{F}_1\hat{\bar{\boldsymbol{h}}}_1\tilde{y} - \kappa_1\boldsymbol{F}_1\left|\tilde{y}\right|\hat{\boldsymbol{W}}_1 \\ \dot{\hat{\boldsymbol{W}}}_2 &= \boldsymbol{F}_2\hat{\bar{\boldsymbol{h}}}_2\tilde{y}u - \kappa_2\boldsymbol{F}_2\left|\tilde{y}\right|\hat{\boldsymbol{W}}_2 \end{aligned} \tag{10}$$

where , $\boldsymbol{F}_i = \boldsymbol{F}_i^T$, $\kappa_i > 0$, $i = 1,2$.

The state estimation error $\tilde{\boldsymbol{x}}\left(t\right)$ and the weight estimation errors of the RBFNN $\tilde{\boldsymbol{W}}_1\left(t\right)$ and $\tilde{\boldsymbol{W}}_2\left(t\right)$ is UUB.

Equation (8) can be rewritten as (11):

$$\dot{\tilde{\boldsymbol{x}}} = \left(\boldsymbol{A} - \boldsymbol{K}\boldsymbol{C}^T\right)\tilde{\boldsymbol{x}} + \boldsymbol{b}\tilde{u} \tag{11}$$

The solution of $\dot{\tilde{\boldsymbol{x}}} = \left(\boldsymbol{A} - \boldsymbol{K}\boldsymbol{C}^T\right)\tilde{\boldsymbol{x}}$ is given as in (12):

$$\tilde{\boldsymbol{x}}(t) = \tilde{\boldsymbol{x}}(0)e^{\int_0^t \left(\boldsymbol{A}-\boldsymbol{K}\boldsymbol{C}^T\right)d\tau} \tag{12}$$

Therefore, the solution of (11) is given by (13):

$$\tilde{\boldsymbol{x}}(t) = \tilde{\boldsymbol{x}}(0)e^{\int_0^t \left(\boldsymbol{A}-\boldsymbol{K}\boldsymbol{C}^T\right)d\tau} + e^{\int_0^t \left(\boldsymbol{A}-\boldsymbol{K}\boldsymbol{C}^T\right)d\tau}\int_0^t b\tilde{u}(\tau)e^{-\int_0^\tau \left(\boldsymbol{A}-\boldsymbol{K}\boldsymbol{C}^T\right)d\tau}d\tau \tag{13}$$

By selecting $\Psi(t,0) = e^{\int_0^t \left(\boldsymbol{A}-\boldsymbol{K}\boldsymbol{C}^T\right)d\tau}$, $\Psi(t,\tau) = e^{\int_0^t \left(\boldsymbol{A}-\boldsymbol{K}\boldsymbol{C}^T\right)d\tau - \int_0^\tau \left(\boldsymbol{A}-\boldsymbol{K}\boldsymbol{C}^T\right)d\tau}$, equation (13) becomes (14):

$$\tilde{\boldsymbol{x}}(t) = \Psi(t,0)\tilde{\boldsymbol{x}}(0) + \int_0^t \Psi(t,\tau)b\tilde{u}(\tau)d\tau \tag{14}$$

$$e^{\int_0^t \left(\boldsymbol{A}-\boldsymbol{K}\boldsymbol{C}^T\right)d\tau - \int_0^\tau \left(\boldsymbol{A}-\boldsymbol{K}\boldsymbol{C}^T\right)d\tau} = e^{\left(\boldsymbol{A}-\boldsymbol{K}\boldsymbol{C}^T\right)(t-\tau)} = e^{\boldsymbol{A}(t-\tau)} \times e^{-\boldsymbol{K}\boldsymbol{C}^T(t-\tau)} = m_0 e^{-\alpha(t-\tau)} \tag{15}$$

where, $m_0 = e^{\boldsymbol{A}(t-\tau)}$, $\gamma = \boldsymbol{K}\boldsymbol{C}^T$. Therefore, $\Psi(t,\tau)$ is bounded by $m_0 e^{-\alpha(t-\tau)}$ with m_0 and γ being positive constants.

From lemma 2 [18] and (11):

$$\left\|\tilde{\boldsymbol{x}}(t)\right\| \le k_1 + k_2\left\|\tilde{\boldsymbol{u}}\right\|_2^\gamma, \ \forall t \ge 0 \tag{16}$$

where, $\left\|\tilde{\boldsymbol{u}}\right\|_2^\gamma = \left\|\tilde{\boldsymbol{W}}_1^T\hat{\boldsymbol{h}}_1 + w_1 + \varepsilon_1 + \left[\tilde{\boldsymbol{W}}_2^T\hat{\boldsymbol{h}}_2 + w_2 + \varepsilon_2\right]u + d + v_1 + v_2\right\|_2^\gamma$

Definition:

$$c = w_1 + \varepsilon_1 + \left[w_2 + \varepsilon_2\right]u + d + v_1 + v_2 \tag{17}$$

Therefore,

$$\left\|\tilde{\boldsymbol{u}}\right\|_2^\gamma = \left\|\tilde{\boldsymbol{W}}_1^T\hat{\boldsymbol{h}}_1 + \tilde{\boldsymbol{W}}_2^T\hat{\boldsymbol{h}}_2 u + c\right\|_2^\gamma \le \left\|\tilde{\boldsymbol{W}}_1^T\hat{\boldsymbol{h}}_1\right\|_2^\gamma + \left\|\tilde{\boldsymbol{W}}_2^T\hat{\boldsymbol{h}}_2 u\right\|_2^\gamma + c_4' \tag{18}$$

with $\left\|c\right\|_2^\gamma \le c_4'$

Since,

$$\left\|\boldsymbol{A}\boldsymbol{x}\right\|_2 \le \left\|\boldsymbol{A}\right\|_F \left\|\boldsymbol{x}\right\|_2 \tag{19}$$

$$\left\|\boldsymbol{x}\right\|_2^\gamma = \sqrt{\int_0^t e^{-\gamma(t-\tau)}\boldsymbol{x}^T(\tau)\boldsymbol{x}(\tau)d\tau} \tag{20}$$

Thus,

$$\begin{aligned}\left\|\tilde{\boldsymbol{W}}_1^T\hat{\boldsymbol{h}}_1\right\|_2^\gamma &\le \left\|\tilde{\boldsymbol{W}}_1^T\right\|_F^\gamma \left\|\hat{\boldsymbol{h}}_1\right\|_2^\gamma = \left\|\tilde{\boldsymbol{W}}_1^T\right\|_F^\gamma \sqrt{\int_0^t e^{-\gamma(t-\tau)}\hat{\boldsymbol{h}}_1\hat{\boldsymbol{h}}_1^T d\tau} \\ &= \left\|\tilde{\boldsymbol{W}}_1^T\right\|_F^\gamma \left\|\hat{\boldsymbol{h}}_1\right\| \sqrt{\int_0^t e^{-\gamma(t-\tau)}d\tau} = \left\|\tilde{\boldsymbol{W}}_1^T\right\|_F^\gamma \left\|\hat{\boldsymbol{h}}_1\right\| \frac{1}{\sqrt{\gamma}} \sqrt{\int_0^t e^{-\gamma(t-\tau)}d\left(-\gamma(t-\tau)\right)} \\ &= \left\|\tilde{\boldsymbol{W}}_1^T\right\|_F^\gamma \left\|\hat{\boldsymbol{h}}_1\right\| \frac{1}{\sqrt{\gamma}} \sqrt{1-e^{-\gamma t}} \le \left\|\tilde{\boldsymbol{W}}_1^T\right\|_F^\gamma \frac{1}{\sqrt{\gamma}} c_5'\end{aligned} \tag{21}$$

Similarly,

$$\left\|\tilde{\boldsymbol{W}}_2^T\hat{\boldsymbol{h}}_2u\right\|_2^\gamma \le \left\|\tilde{\boldsymbol{W}}_2^T\right\|_F^\gamma \frac{1}{\sqrt{\gamma}}c_6^{'} \tag{22}$$

where, $c_5^{'} = \left\|\hat{\boldsymbol{h}}_1\right\|\sqrt{1-e^{-\alpha t}}$, $c_6^{'} = \left\|\hat{\boldsymbol{h}}_2\right\|\sqrt{1-e^{-\alpha t}}u_d$.

Now,

$$\left\|\tilde{\boldsymbol{u}}\right\|_2^\gamma \le \left\|\tilde{\boldsymbol{W}}_1^T\right\|_F^\gamma \frac{1}{\sqrt{\gamma}}c_5^{'} + \left\|\tilde{\boldsymbol{W}}_2^T\right\|_F^\gamma \frac{1}{\sqrt{\gamma}}c_6^{'} + c_4^{'} \tag{23}$$

Substituting (23) into (16), we obtain (24):

$$\left\|\tilde{\boldsymbol{x}}(t)\right\| \le k_1 + k_2\left(\left\|\tilde{\boldsymbol{W}}_1^T\right\|_F^\gamma \frac{1}{\sqrt{\gamma}}c_5^{'} + \left\|\tilde{\boldsymbol{W}}_2^T\right\|_F^\gamma \frac{1}{\sqrt{\gamma}}c_6^{'} + c_4^{'}\right) = c_3 + \left(c_4 + c_5\left\|\tilde{\boldsymbol{W}}_1\right\|_F^\gamma + c_6\left\|\tilde{\boldsymbol{W}}_2\right\|_F^\gamma\right)\frac{1}{\sqrt{\gamma}} \tag{24}$$

where, $c_3 = k_1$, $c_4 = k_2c_4^{'}$, $c_5 = k_2c_5^{'}$ $c_6 = k_2c_6^{'}$, c_4, c_5, c_6 being positive constants.

2.3. Design of a robust sliding mode controller based on a new Quasi-sliding mode and an adaptive artificial neural network observer for robot

The tracking error is defined as (25):

$$e_r = \hat{x}_1 - x_{1d} \tag{25}$$

where, x_{1d} is the desired position.

By taking the first and second derivatives of (25), we obtain (26) and (27):

$$\dot{e}_r = \dot{\hat{x}}_1 - \dot{x}_{1d} = \hat{x}_2 - \dot{x}_{1d} \tag{26}$$

$$\ddot{e}_r = \dot{\hat{x}}_2 - \ddot{x}_{1d} \tag{27}$$

The sliding surface is defined as in (28):

$$s = \lambda e_r + \dot{e}_r \tag{28}$$

with λ required to satisfy the Hurwitz condition, $\lambda > 0$.

The Lyapunov function is defined as in (29):

$$V_c = \frac{1}{2}s^2 \tag{29}$$

Taking the derivative of (28), we obtain (30):

$$\dot{s} = \lambda\dot{e}_r + \ddot{e}_r \tag{30}$$

Substituting (26) and (27) into (30), we obtain (31):

$$\dot{s} = \lambda\left(\dot{\hat{x}}_1 - \dot{x}_{1d}\right) + \left(\dot{\hat{x}}_2 - \ddot{x}_{1d}\right) \tag{31}$$

From the observer (8), we have (32) and (33):

$$\dot{\hat{x}}_1 = \hat{x}_2 + K_1\left(x_1 - \hat{x}_1\right) \tag{32}$$

$$\dot{\hat{x}}_2 = \hat{f}\left(\boldsymbol{x}\right) + \hat{g}\left(\boldsymbol{x}\right)u - v(t) + K_2\left(x_1 - \hat{x}_1\right) \tag{33}$$

Substituting (32) and (33) into (31), we obtain (34):

$$\dot{s} = \lambda\left(\left(\hat{x}_2 + K_1\left(x_1 - \hat{x}_1\right)\right) - \dot{x}_{1d}\right) + \left(\hat{f}\left(\boldsymbol{x}\right) + \hat{g}\left(\boldsymbol{x}\right)u - v(t) + K_2\left(x_1 - \hat{x}_1\right) - \ddot{x}_{1d}\right) \tag{34}$$

With the exponential reaching law as (35):

$$\dot{s} = -\eta sign(s) - \mu s, \qquad \eta > 0, \mu > 0 \tag{35}$$

Now, the sliding mode controller based on an adaptive artificial neural network observer for the robot is (36):

$$u = \frac{1}{\hat{g}(\boldsymbol{x})}\left(-\lambda\left(\hat{x}_2 + K_1\left(x_1 - \hat{x}_1\right) - \dot{x}_{1d}\right) - \hat{f}(\boldsymbol{x}) + v(t) - K_2\left(x_1 - \hat{x}_1\right) + \ddot{x}_{1d} - \eta sign(s) - \mu s\right) \tag{36}$$

In (36), due to the switch term $\eta sign(s)$, the chattering can be caused, especially for a large disturbance, which can damage system components such as actuators. Therefore, the *tanh* function (a new quasi-sliding mode) will be used instead of the *signum* function in this study as (37) follows:

$$\tanh\left(\frac{s}{\beta}\right) = \frac{e^{\frac{s}{\beta}} - e^{-\frac{s}{\beta}}}{e^{\frac{s}{\beta}} + e^{-\frac{s}{\beta}}} \tag{37}$$

where $\beta > 0$, the steepness of the tanh function is determined by β value.

To ensure $\dot{V}_c = s\dot{s} < 0$, the RSMC-NQ-ARBFNN-O controller is designed as in (38):

$$u_{new\_Quasi} = \frac{1}{\hat{g}(\boldsymbol{x})}\begin{pmatrix} -\lambda\left(\hat{x}_2 + K_1\left(x_1 - \hat{x}_1\right) - \dot{x}_{1d}\right) - \hat{f}(\boldsymbol{x}) + v(t) \\ -K_2\left(x_1 - \hat{x}_1\right) + \ddot{x}_{1d} - \eta \tanh\left(\frac{s}{\alpha}\right) - \mu s \end{pmatrix} \tag{38}$$

with $\eta > 0$, $\alpha > 0, \mu > 0$.

Now, we have (39):

$$s\dot{s} = s\left(-\eta \tanh\left(\frac{s}{\alpha}\right) - \mu s - d(t)\right) = -\eta s \tanh\left(\frac{s}{\alpha}\right) - \mu s^2 - sd(t) < 0 \tag{39}$$

Therefore,

$$\dot{V} \le 0 \ (\dot{V} = 0 \ \text{when} \ s = 0) \tag{40}$$

The Lyapunov function is defined as (41):

$$V = V_0 + V_c \tag{41}$$

Therefore,

$$\dot{V} = \dot{V}_0 + \dot{V}_c \le 0 \tag{42}$$

3. RESULTS AND DISCUSSION

The parameters of the RBF neural network as follows: c_1 = (1/3)*[-3 -2 -1 0 1 2 3]; c_2 = (1/3)*[-3 -2 -1 0 1 2 3]; b = 5. The parameters of the RSMC-NQ-ARBFNN-O controller as follows: K_1 = 400; K_2 = 800; λ = 28; D = 0.8; α = 0.1; η = 0.1 and μ = 5.0. These parameters are selected by the trial-and-error method.

The simulation results of the RSMC-NQ-ARBFNN-O controller applied to the robot are done in MATLAB/Simulink. Figure 1 shows the actual response and error q (x_1) of the robot with the RSMC-NQ-ARBFNN-O controller when q_d= 0.1(rad). The actual angular position of the robot (x_1=q) converges to the desired angular position (x_{1d}=q_d) with the rising time achieves 0.4656(s), the settling time is 0.7690 (s), the steady-state error is 0(rad), the overshoot is 0(%). The values of Root Mean Squared Error (RMSE), Integral Absolute Error (IAE), and Integral Square Error (ISE) are 1.7549e-06, 0.0222, and 0.001124, respectively. The performance indices of the RSMC-NQ-

ARBFNN-O controller are presented in table 1 and compared with the traditional sliding mode control. The performance metrics in table 1 highlight the superiority of the RSMC-NQ-ARBFNN-O controller over the methods implemented in the traditional SMC.

Table 1. *Performance indices of the RSMC-NQ-ARBFNN-O controller.*

Quality criteria	Settling time (s)	Rising time (s)	Steady-state error (rad)	Overshoot (%)	RMSE	IAE	ISE
RSMC-NQ-ARBFNN-O	0.7690	0.4656	0	0	1.7549e-06	0.0222	0.001124
SMC	1.0953	0.8742	0	0	6.772e-06	0.0464	0.00573

The control signal of the robot with the proposed controller applied to the robot is presented in figure 2. The amplitude of the chattering phenomenon in the RSMC-NQ-ARBFNN-O controller with a power reaching law and the tanh function has been reduced.

Figure 3 presents the sliding surface s of the proposed controller. This sliding surface at start-up according to the sliding coefficient value. Then, s rapidly reaches the convergence point (stabilization point) and keeps sliding around $s = 0$. This proves the finite time problem in sliding mode control, which is applied in this study.

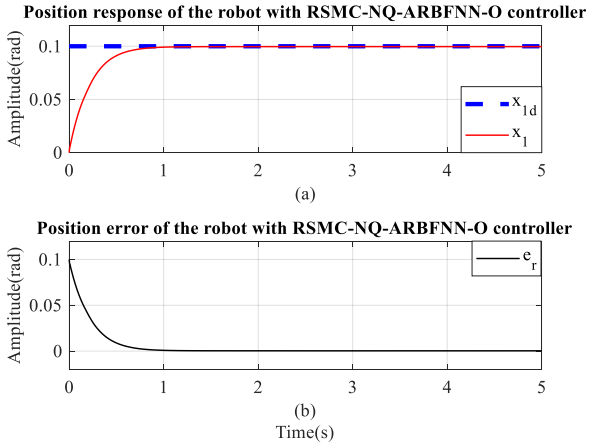

Figure 1. *Response and error of the robot with the RSMC-NQ-ARBFNN-O controller when $x_{1d} = 0.1$ (rad).*

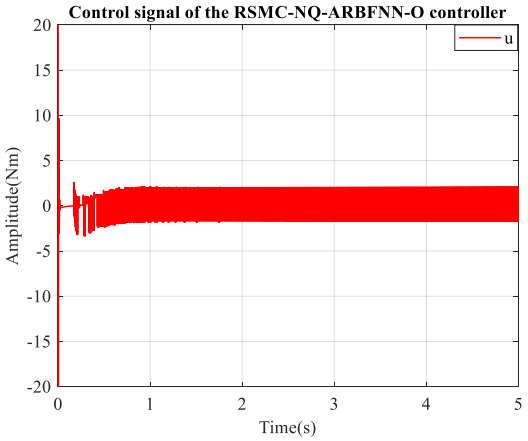

Figure 2. *Control signal of the RSMC-NQ-ARBFNN-O controller.*

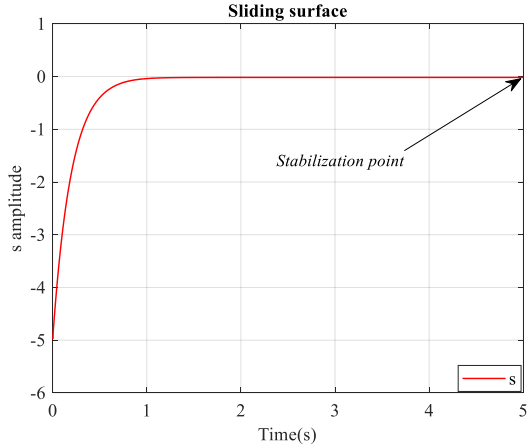

Figure 3. *Sliding surface s of the proposed controller.*

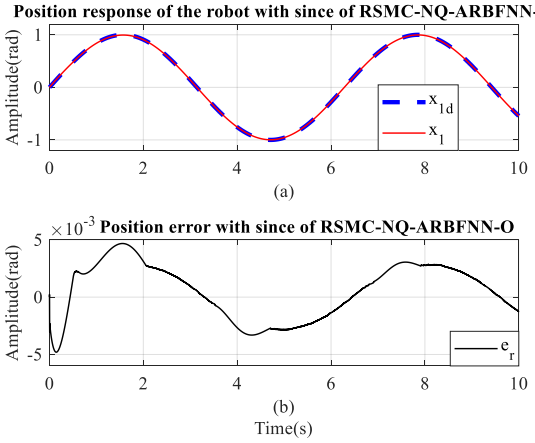

Figure 4. *Position response and error of the robot using the RSMC-NQ-ARBFNN-O controller with x_{1d} = sin(t) (rad).*

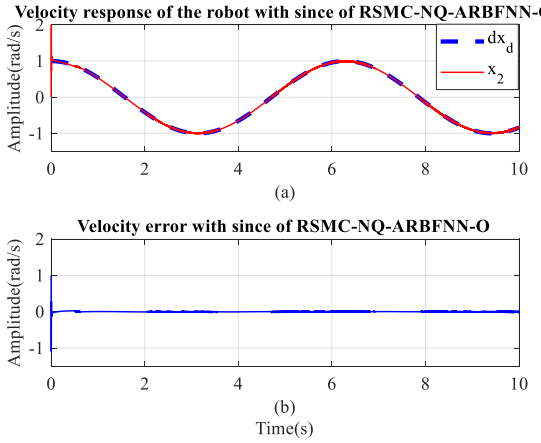

Figure 5. *Velocity response and error of the RSMC-NQ-ARBFNN-O controller with since of RSMC-NQ-ARBFNN-O.*

The actual position and tracking velocity responses of the robot with the RSMC-NQ-ARBFNN-O controller under a sine input are presented in figure 4a and figure 5a. The actual position and velocity of the robot converge to the desired values with the tracking errors approaching zero (figure 4b và figure 5b), moreover, the proposed RSMC-NQ-ARBFNN-O controller effectively eliminates the chattering phenomenon in the control signal.

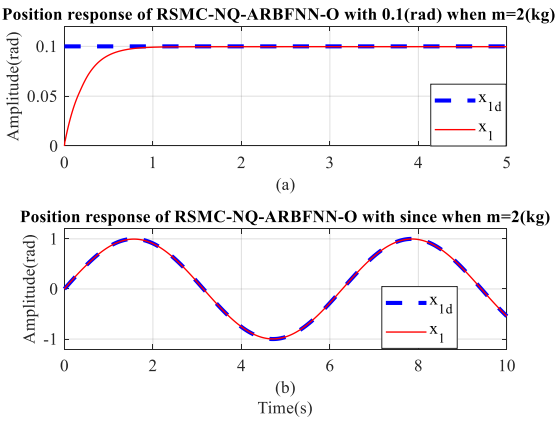

Figure 6. *Position responses of the RSMC-NQ-ARBFNN-O controller with x_{1d} = 0.1(rad) and since when m = 2 (kg).*

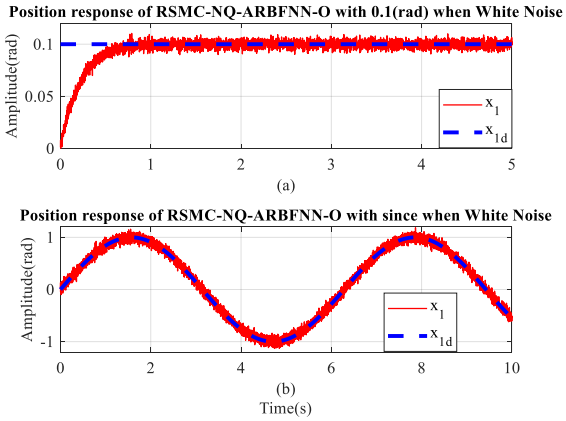

Figure 7. *Position responses of the RSMC-NQ-ARBFNN-O controller with x_{1d} = 0.1(rad) and since with white noise.*

Figure 6 illustrates the actual position response q of the robot with the RSMC-NQ-ARBFNN-O controller in the case of m = 2 (kg) when qd = 0.1 (rad) and qd = sin(t)(rad). The actual position of the robot still tracks the desired trajectory, with the rising time is 0.8001 (s) (which is greater than in the case of m = 1 (kg)), the settling time is 1.2808 (s), while the steady-state error and overshoot are eliminated.

The actual position response of the robot using the proposed controller when the reference inputs are qd = 0.1 (rad) and qd = sin(t)(rad) in the case where white noise is applied at the robot's output (assumed to be sensor noise), the actual position response is shown in figure 7. The actual positions of the robot still converge to the desired reference positions.

4. CONCLUSIONS

The study has designed, analyzed, and evaluated the simulation results of a sliding mode controller based on a new Quasi-sliding mode and a robust adaptive artificial neural network observer for robotic systems. The robust adaptive RBFNN observer has effectively estimated the states x_1, x_2, $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$ in the mathematical model of the robot. The sliding mode controller based on the new Quasi-sliding mode combined with the power reaching law and the adaptive robust RBFNN observer has been designed to ensure Lyapunov stability and achieved trajectory tracking control of the robot with the rising time achieves 0.4656(s), the settling time is 0.7690(s), the steady-state error is 0(rad), the overshoot is 0(%). The values of RMSE, IAE, and ISE are 1.7549e-06, 0.0222 and 0.001124, respectively. The simulation results with a payload mass of m = 2 (kg) and white noise acting on the robot's output also demonstrate the effectiveness and robustness of the proposed control method, with the tracking error converging to zero. In the future, the study will use optimization algorithms to determine the optimal values of the parameters of the proposed control method, find the optimal number of neurons in the hidden layer of the RBFNN neural network and experiment with the actual model.

REFERENCES

[1]. P. Bhavagna Sai Rajeev et al., *"Dynamic analysis of single link and R-P manipulators"*, International Journal of Science, Technology and Management (IJSTM), vol. 11, no. 10, pp. 30–39, (2022).

[2]. T. S. Lee, E. A. Alandoli, *"A critical review of modelling methods for flexible and rigid link manipulators"*, Journal of the Brazilian Society of Mechanical Sciences and Engineering, vol. 42, no. 10, pp. 1–14, (2020).

[3]. K. Bingi, B. Rajanarayan Prusty, A. Pal Singh, *"A review on fractional-order modelling and control of robotic manipulators"*, Fractal and Fractional, vol. 7, no. 1, pp. 1–29, (2023).

[4]. J. Humaidi, I. K. Ibraheem, A. T. Azar, M. E. Sadiq, *"A new adaptive synergetic control design for single link robot arm actuated by pneumatic muscles"*, Entropy, vol. 22, no. 7, pp. 1–24, (2020).

[5]. Z. Yan, X. Lai, Q. Meng, P. Zhang, M. Wu, *"Tracking control of single-link flexible-joint manipulator with unmodeled dynamics and dead zone"*, International Journal of Robust and Nonlinear Control, vol. 31, no. 4, pp. 1270–1287, (2021).

[6]. A. Fayazi, N. Pariz, A. Karimpour, V. Feliu-Batlle, S. H. HosseinNia, *"Adaptive sliding mode impedance control of single-link flexible manipulators interacting with the environment at an unknown intermediate point"*, Robotica, vol. 38, no. 9, pp. 1642–1664, (2020).

[7]. Duong Xuan, *"Dynamics and control analysis of a single flexible link robot with translational joints"*, Science and Technology Development Journal – Engineering and Technology, vol. 3, no. 4, pp. 588–595, (2020).

[8]. Johnson Antony A., *"Motion control of single link flexible joint robot manipulator using ANFIS MATLAB simulation"*, Middle East Journal of Applied Science and Technology (MEJAST), vol. 2, no. 4, pp. 26–35, (2019).

[9]. D. Dermawan, H. Abbas, R. Syam, Z. Djafar, A. K. Muhammad, *"Dynamic modeling of a single-link flexible manipulator robot with translational and rotational motions"*, IIUM Engineering Journal, vol. 21, no. 1, pp. 228–239, (2020).

[10]. E. M. Raju, L. S. R. Krishna, Y. S. C. Mouli, V. N. Rao, *"Effect of link flexibility on tip position of a single link robotic arm"*, Journal of Physics: Conference Series, vol. 662, pp. 1–7, (2015).

[11]. A. Zhang, Z. Lin, B. Wang, Z. Han, *"Nonlinear model predictive control of single-link flexible-joint robot using recurrent neural network and differential evolution optimization"*, Electronics, vol. 10, no. 19, pp. 1–19, (2021).

[12]. J. F. Peza-Solís, G. Silva-Navarro, N. R. Castro-Linares, *"Trajectory tracking control in a single flexible-link robot using finite differences and sliding modes"*, Journal of Applied Research and Technology, vol. 13, no. 1, pp. 70–78, (2015).

[13]. H. Ullah, et al., *"Robust output feedback control of single-link flexible-joint robot manipulator with matched disturbances using high gain observer"*, Sensors, vol. 21, no. 9, pp. 1–22, (2021).

[14]. H. Zhang et al., *"Radial basis function neural network sliding mode control for ship path following based on position prediction"*, Journal of Marine Science and Engineering, vol. 9, p. 1055, (2021).
[15]. M. Mancini, E. Capello, E. Punta, *"Sliding mode control with chattering attenuation and hardware constraints in spacecraft applications"*, IFAC PapersOnLine, vol. 53, no. 2, pp. 5147–5152, (2020).
[16]. W. Alqaisi, C. El-Bayeh, *"Adaptive control based on radial basis function neural network approximation for quadrotor"*, Proceedings of the Annual System of Systems Engineering Conference (SOSE), pp. 214–219, (2022).
[17]. Y. H. Kim, F. L. Lewis, C. T. Abdallah, *"A dynamic recurrent neural-network-based adaptive observer for a class of nonlinear systems"*, Automatica, vol. 33, no. 8, pp. 1539–1543, (1997).
[18]. J. Liu, *"Radial basis function (RBF) neural network control for mechanical systems: design, analysis and MATLAB simulation"*, Springer, (2013).

TÓM TẮT

Điều khiển trượt bền vững dựa trên chế độ Quasi mới và bộ quan sát mạng nơ-ron nhân tạo thích ứng cho robot

Nghiên cứu này thiết kế và đánh giá kết quả mô phỏng bộ điều khiển trượt bền vững dựa vào chế độ Quasi mới và bộ quan sát mạng nơ-ron nhân tạo (RBFNN: Radial basis function neural network) thích nghi cho robot. Robot công nghiệp (tay máy robot) là một tay máy đa chức năng, có thể được lập trình để thực hiện các nhiệm vụ nguy hiểm và/hoặc lặp đi lặp lại với độ chính xác cao. Bộ quan sát mạng nơ-ron RBFNN bền vững thích nghi được sử dụng để ước lượng các trạng thái và hàm phi tuyến trong mô tả toán học của robot. Bộ điều khiển trượt dựa vào chế độ Quasi mới kết hợp với bộ quan sát mạng nơ-ron RBFNN bền vững thích nghi để điều khiển bám quỹ đạo robot với các chỉ tiêu chất lượng đạt được phù hợp. Các trọng số của mạng nơ-ron RBFNN được cập nhật trực tuyến. Tính ổn định của các phương pháp điều khiển đề xuất đều được chứng minh bằng lý thuyết ổn định Lyapunov. Các kết quả mô phỏng trong MATLAB/Simulink đã cho thấy hiệu quả, tính bền vững của phương pháp đề xuất với thời gian tăng đạt 0,4656(s), thời gian xác lập là 0,7690(s), sai số xác lập là 0(rad), độ vọt lố là 0(%), các giá trị của RMSE (Root Mean Squared Error), IAE (Integral Absolute Error) và ISE (Integral Square Error) lần lượt là 1,7549e-06, 0,0222 và 0,001124.

Từ khóa: Robot; Mạng nơ-ron RBF; Bộ quan sát; Thích nghi; Điều khiển trượt.