www.czasopisma.pan.pl PAN POLSKA AKADEMIA NAUK www.journals.pan.pl

ARCHIVE OF MECHANICAL ENGINEERING

DOI: 10.24425/ame.2025.154742 2025, Vol. 72, No. 3, pp. 519–544

Ngoc-Hiep TRAN[1], Danh-Huy NGUYEN [1], Thi-Van-Anh NGUYEN [1]

Enhanced robustness of robot manipulators using fixed-time sliding mode control and RBF neural network

Received 28 April 2025, **Revised** 25 July 2025, **Accepted** 30 July 2025, **Published online** 7 August 2025

Keywords: robot manipulator, fixed-time sliding mode control, radial basis function neural network, model uncertainty

This paper presents a control framework that integrates fixed-time sliding mode control (FxTSMC) with a radial basis function neural network (RBF-NN). The novelty of the proposed approach lies in the synergy between the two techniques, leveraging their respective strengths to enhance overall system performance. The FxTSMC component guarantees fast system response within a predefined time, while the RBF-NN offers adaptive capabilities to handle model uncertainties, thereby improving robustness. The stability of the combined approach is theoretically established and validated through its application to a nonlinear system, specifically a 2-DoF robotic manipulator. Comparative simulations with existing methods are conducted, accompanied by discussions on the results.

1. Introduction

Robotic manipulators [1, 2] are pivotal in modern automation, characterized by their multi-jointed, arm-like structures that emulate the dexterity and precision of human movements. These versatile machines have revolutionized various industries, including manufacturing, healthcare, aerospace, and logistics, by executing tasks ranging from assembly and welding to delicate surgical procedures and intricate material handling. Despite their transformative potential, the control of robotic manipulators poses significant challenges. Their dynamic and nonlinear behaviors, coupled with the interdependencies of their multiple joints, complicate

✉ Thi-Van-Anh NGUYEN, email: anh.nguyenthivan1@hust.edu.vn

[1]School of Electrical and Electronic Engineering, Hanoi University of Science and Technology, Hanoi, Vietnam

precise control. Moreover, external uncertainties such as varying payloads, environmental interactions, and mechanical wear further exacerbate these difficulties. Achieving accurate, stable, and efficient control in the face of these challenges requires advanced techniques that can adapt to real-time changes and ensure high performance. As the demand for more sophisticated and capable robotic systems grows, so too does the need for innovative control strategies that can address these complexities effectively.

To address the control challenges faced by robotic manipulators, numerous control strategies have been developed to enhance their performance, accuracy, and stability. One of the most fundamental approaches is the Proportional-Integral-Derivative (PID) controller, as applied in medical robotics to ensure precise trajectory following [3] and further enhanced with adaptive mechanisms for increased robustness [4]. Model Predictive Control (MPC) [5] has also been employed to effectively manage the motion control of robotic arms by optimizing future control actions based on a model of the system. Fuzzy Logic Control (FLC), known for its ability to handle nonlinearities and uncertainties, has been utilized to govern the behavior of robotic manipulators [6, 7], and has been combined with disturbance observers to mitigate noise and boost control performance [8]. Reinforcement learning (RL) has emerged as a powerful approach in the field of robotics by allowing agents to learn optimal control policies through interaction with the environment. Recent studies have applied RL techniques to robot manipulators, demonstrating improved performance in tasks such as trajectory tracking and manipulation under varying conditions [9–11]. In [9], the authors present two reinforcement learning (RL)-based compensation methods that improve industrial robotic manipulators' control by learning a correction signal to counteract unmodeled disturbances. Similarly, in [10], Hu and Si (2018) propose a neural network framework consisting of a state network (predicting manipulator states), an action network (learning optimal control policies), and a critic network (evaluating policy performance) within an RL framework, with a convergence analysis validating its effectiveness. Expanding on this, Hu et al. (2019) introduce [11], which eliminates the need for an explicit dynamic model by incorporating a kernel-based approach within an RL framework, allowing the system to adapt to complex, unknown dynamics in real-time, enhancing tracking accuracy and overall control performance. Additionally, Sliding Mode Control (SMC) [12–17] is favored for its inherent robustness and ability to maintain stability under varying conditions, making it a popular choice for robotic manipulator control. In [13], Rui Li et al. propose an adaptive sliding mode control (SMC) strategy that compensates for actuator faults using a fast terminal nonsingular approach combined with a non-Lipschitz Lyapunov framework, ensuring asymptotic convergence even in the presence of system failures. The study [14] by Saad Jamshed Abbasi et al. introduces a robust control strategy that combines an extended state observer with SMC to accurately track trajectories of multi-DoF robot manipulators under uncertainties and disturbances. Their methods demonstrate improved tracking performance compared to traditional controllers,

with simulation results validating the robustness and precision of the proposed designs. Each of these methods brings unique strengths to the table, addressing the diverse and complex requirements of robotic manipulator systems.

This paper examines the application of an advanced sliding mode control technique known as Fixed-time Sliding Mode Control . Recently, FxTSMC has gained considerable attention due to its ability to drive the error to zero within a predetermined time frame, regardless of initial conditions, enhancing both response speed and control efficiency. Unlike conventional SMC, which ensures error convergence as time approaches infinity, FxTSMC offers a predetermined time-bound for error convergence, making it highly effective for dynamic systems. This control approach has been successfully applied to various domains, including spacecraft attitude stabilization [18, 19], tracking control for robotic manipulators [20, 21] and Mars entry vehicle control [22]. Moreover, the FxTSMC algorithm has also been applied in combination with observers to ensure precise feedback control, as demonstrated in the studies [23–25]. The authors in [24] propose a practically robust fixed-time sliding mode control scheme incorporating a cascaded observer (CFxTSMO), which guarantees rapid convergence and accurate trajectory tracking regardless of initial conditions, as validated through both simulations and experimental results. In [25], an ESO-based adaptive fixed-time integral sliding mode controller is developed using the singular perturbation method to effectively manage high-order dynamics, measurement noise, and unknown disturbances. By decomposing the system and introducing adaptive control laws, the proposed method achieves fixed-time convergence, mitigates chattering, and delivers improved tracking accuracy compared to conventional approaches. Leveraging these advantages, FxTSMC is utilized in this study to provide a robust and efficient control solution for robotic manipulators.

Integrating a Radial Basis Function (RBF) neural network into the control framework greatly improves its capacity to manage uncertainties and address model discrepancies. The neural network structure consists of nodes, each equipped with an activation function. Various types of activation functions are commonly used, including binary step, linear activation, sigmoid, hyperbolic tangent, and Gaussian functions. In the case of RBF networks, the activation function used is the Gaussian function, which is inherently nonlinear. By forming a linear combination of Gaussian functions with associated weights, an RBF network can approximate nonlinear functions. This capability makes the RBF-NN particularly effective for approximating unknown functions and adapting to system variations, a feature that has been widely leveraged in diverse control applications [26–32]. This adaptability is crucial for robotic systems that frequently encounter unpredictable environmental conditions and dynamic changes, such as variations in payload or external disturbances. Incorporating the RBF-NN into the FxTSMC structure allows the control system to dynamically adjust to these uncertainties, ensuring sustained performance and stability. Many studies have leveraged the RBF-NN's capacity to enhance control schemes, especially in scenarios requiring rapid adaptation to non-linearities and disturbances. By combining the precise, finite-time convergence of FxTSMC

with the adaptive capabilities of the RBF-NN, the control system achieves a robust response to external influences and internal variations.

When proposing a control methodology, it is essential to establish a solid theoretical foundation that ensures system stability. Equally important is the validation of theoretical claims through simulations or experiments on representative systems, which serves to demonstrate the effectiveness of the proposed approach. In this research, the integration of the RBF-NN into the FxTSMC framework enables the system to adapt dynamically to uncertainties, thereby preserving both stability and control performance.

The key contribution of this study lies in the strategic combination of the strengths of each algorithm, leveraging their advantages to enhance both response time and tracking accuracy. The fast convergence property of FxTSMC, coupled with the adaptive capability of the RBF-NN under time-varying conditions, necessitates thorough verification. Accordingly, the motivation of this work is to provide both an analytical stability proof and a simulation-based validation of the proposed method using a nonlinear 2-DoF robotic manipulator. The robustness of the control strategy is further assessed through simulation scenarios involving various model uncertainties. The main contributions of this research can be summarized as follows:

- Propose a controller that integrates Fixed-time Sliding Mode Control with Radial Basis Function Neural Network to address the control challenges in uncertain models.
- Enhance the system's response time using the FxTSMC controller, with the stability of the closed system proven through Lyapunov stability theory and validate the approach using simulations on a 2-DoF manipulator robot model.
- Perform comparisons and evaluations to analyze the performance of the proposed method in relation to other control techniques.

2. Model of 2-DoF robot manipulator

The 2-DoF robot manipulator model is depicted in Fig. 1. This robot functions by applying torques τ_1 and τ_2 to its first and second rotating joints, respectively. These joints rotate counterclockwise around the z-axis, resulting in angular positions q_1 and q_2. For simplification during transformations, the combined angle $q_1 + q_2$ is referred to as q_{12}. The detailed system parameters, including the mass of each arm, arm lengths, centers of mass, moments of inertia, friction coefficients, and gravitational acceleration used to model the robot's dynamics, are listed in Table 1.

To model the system using the Euler-Lagrange method, the first step involves calculating the kinetic and potential energies of the robot. The kinetic energy of the 2-DoF robot manipulator is determined using the following formula:

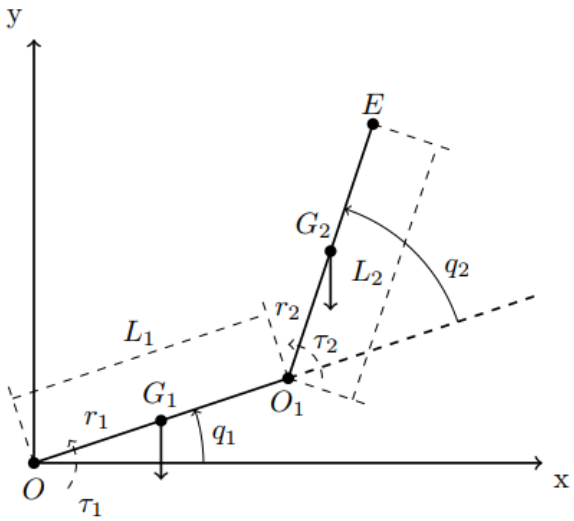

Fig. 1. Description of 2-DoF robot manipulator

Table 1. Robot manipulator parameters

Symbol	Explanation	Value	Unit
m_1, m_2	Masses of the first and second arms	15, 9	kg
L_1, L_2	Lengths of the first and second arms	0.5, 0.5	m
r_1, r_2	Distances from joints to mass center of arms	0.25, 0.25	m
I_1, I_2	Moment of inertia of arms	0.313, 0.313	kg m^2
g	Gravitational acceleration	9.81	m/s^2
f_{v1}, f_{v2}	Friction coefficients of arms	$\frac{1}{7}, \frac{1}{7}$	Nms/rad

$$K = \frac{1}{2}\left(m_1 r_1^2 + m_2 L_1^2 + I_1\right)\dot{q}_1^2 + \frac{1}{2} I_2 \dot{q}_2^2 + m_2 L_1 r_2 \cos q_2 \dot{q}_1 \left(\dot{q}_1 + \dot{q}_2\right) + \frac{1}{2} m_2 r_2^2 \left(\dot{q}_1 + \dot{q}_2\right)^2 . \tag{1}$$

The potential energy of the 2-DoF robot manipulator is derived as follows:

$$P = m_1 g r_1 \sin q_1 + m_2 g \left(L_1 \sin q_1 + r_2 \sin q_{12}\right) . \tag{2}$$

Based on the previously calculated kinetic and potential energies, the Lagrangian of the system is derived as follows:

$$L = K - P = \frac{1}{2}\left(m_1 r_1^2 + m_2 L_1^2 + m_2 r_2^2 + I_1\right)\dot{q}_1^2 + \frac{1}{2}\left(m_2 r_2^2 + I_2\right)\dot{q}_2^2 + m_2 r_2^2 \dot{q}_1 \dot{q}_2 + m_2 L_1 r_2 \cos q_2 \dot{q}_1 \left(\dot{q}_1 + \dot{q}_2\right) - m_1 g r_1 \sin q_1 - m_2 g \left(L_1 \sin q_1 + r_2 \sin q_{12}\right). \tag{3}$$

Euler-Lagrange method is employed to derive the dynamic model of the system. In this model, q_i represents the rotation angles of joint i, and τ_i denotes the torques applied to joint i.

$$\tau_i = \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i}. \tag{4}$$

After detailed calculations and simplifications, the dynamic model is presented:

$$\tau_1 = \left(a_1 + 2a_2 \cos q_2\right)\ddot{q}_1 + \left(a_3 + a_2 \cos q_2\right)\ddot{q}_2 - a_2 \dot{q}_2^2 \sin q_2 + \left(f_{v1} - a_2 \dot{q}_2 \sin q_2\right)\dot{q}_1 + a_4 \cos q_1 + a_5 \cos q_{12}, \tag{5}$$

$$\tau_2 = \left(a_3 + a_2 \cos q_2\right)\ddot{q}_1 + \left(a_3 + a_6\right)\ddot{q}_2 + a_2 \sin q_2 \dot{q}_1^2 + f_{v2}\dot{q}_2 + a_5 \cos q_{12}, \tag{6}$$

with $a_1 = m_1 r_1^2 + m_2 L_1^2 + m_2 r_2^2 + I_1$, $a_2 = m_2 L_1 r_2$, $a_3 = m_2 r_2^2$, $a_4 = m_1 g r_1 + m_2 g L_1$, $a_5 = m_2 g r_2$, $a_6 = I_2$.

The derived dynamic model of the 2-DoF robot manipulator is expressed as:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau, \tag{7}$$

where:

$$q = \begin{bmatrix} q_1 & q_2 \end{bmatrix}^{\mathrm{T}}, \tag{8}$$

$$\tau = \begin{bmatrix} \tau_1 & \tau_2 \end{bmatrix}^{\mathrm{T}}, \tag{9}$$

$$M(q) = \begin{bmatrix} a_1 + 2a_2 \cos q_2 & a_3 + a_2 \cos q_2 \\ a_3 + a_2 \cos q_2 & a_3 + a_6 \end{bmatrix}, \tag{10}$$

$$C(q, \dot{q}) = \begin{bmatrix} f_{v1} - 2a_2 \dot{q}_2 \sin q_2 & -a_2 \dot{q}_2 \sin q_2 \\ a_2 \sin q_2 \dot{q}_1 & f_{v2} \end{bmatrix}, \tag{11}$$

$$G(q) = \begin{bmatrix} a_4 \cos q_1 + a_5 \cos q_{12} \\ a_5 \cos q_{12} \end{bmatrix}. \tag{12}$$

3. Control design

A proposed solution for the trajectory tracking control of a robot manipulator integrates a FxTSMC approach with RBF-NNs. The FxTSMC controller is designed to achieve control within a predetermined settling time, ensuring rapid response for the robot manipulator to track desired trajectories efficiently. However, inherent uncertainties persist between the actual system dynamics and the idealized mathematical model, which can compromise control performance.

To address this challenge, the RBF-NN is incorporated into the control architecture. This neural network serves the crucial role of approximating unknown functions within the control signal. By doing so, it enables the controller to effectively adapt to model uncertainties and variations in system parameters during operation. This adaptive capability enhances the robustness and accuracy of the control system, ensuring precise trajectory tracking despite the presence of uncertainties in the underlying system model.

3.1. Fixed-time sliding mode control

The deviation between the state variables and the reference signal is denoted as:

$$e = [e_i]^{\mathrm{T}}, \; e_i = q_i - q_{id}\,, \tag{13}$$

$$\dot{e} = [\dot{e}_i]^{\mathrm{T}}, \; \dot{e}_i = \dot{q}_i - \dot{q}_{id}\,, \tag{14}$$

where q_{id} is the reference angle of the i^{th} joint. To construct the sliding surface, let's define:

$$\mathrm{sig}^{\delta}(e_i) = |e_i|^{\delta}\mathrm{sign}(e_i). \tag{15}$$

A nonlinear equation is proposed in [21] to construct a fixed-time sliding surface along with its first derivative:

$$f(e_i) = \begin{cases} K_a \mathrm{sig}^{2-\mu}(e_i) + K_b \mu^{|e_i|} e_i, & |e_i| < \mu, \\ \mathrm{sig}^{1-\mu}(e_i)\,, & |e_i| \geqslant \mu, \end{cases} \tag{16}$$

$$h(e_i) = \frac{\partial f(e_i)}{\partial e_i} = \begin{cases} K_a(2-\mu)\,|e_i|^{1-\mu} + K_b\left(|e_i|\ln\mu + 1\right)\mu^{|e_i|}, & |e_i| < \mu, \\ (1-\mu)\,|e_i|^{-\mu}\,, & |e_i| \geqslant \mu, \end{cases} \tag{17}$$

where μ is a positive constant and chosen as $0 < \mu < \exp(-1)$. To ensure $f(e_i)$ and $h(e_i)$ are continuous when $|e_i| = \mu$, K_a and K_b are defined as:

$$K_a = \frac{-1 - \ln\mu}{1 - \mu - \mu\ln\mu}, \quad K_b = \frac{\mu^{-2\mu}}{1 - \mu - \mu\ln\mu}. \tag{18}$$

From the above equations, a fixed-time sliding surface is chosen as follows:

$$s = \dot{e} + C_1 F(e) + C_2 E^{\delta}(e). \tag{19}$$

Differentiating s with respect to time:

$$\dot{s} = \ddot{e} + C_1 H(e)\dot{e} + C_2 N^{\delta-1}(e)\dot{e}\,, \tag{20}$$

where C_1, C_2 are positive definite dianogal matrixes, δ is a constant satisfying $\delta > 1$. The definitions of $F(e)$, $E^{\delta}(e)$, $H(e)$ and $N^{\delta-1}(e)$ are as follows:

$$F(e) = [f(e_i)]^{\mathrm{T}}, \tag{21}$$

$$E^{\delta}(e) = \left[\mathrm{sig}^{\delta}(e_i)\right]^{\mathrm{T}}, \tag{22}$$

$$H(e) = \mathrm{diag}\,\{h(e_i)\}\,, \tag{23}$$

$$N^{\delta-1}(e) = \mathrm{diag}\{|e_i|^{\delta-1}\}, \tag{24}$$

where e_i is the i^{th} component of vector e in (13), $f(e_i)$ and $h(e_i)$ are described in (16) and (17), respectively.

For the system (7), the control signal of the FxTSMC is designed as the following formula:

$$\tau = -K_1 E^{v_1}(s) - K_2 E^{v_2}(s) - K_3 \frac{s}{\|s\|} - \eta\,, \tag{25}$$

where K_1, K_2 denote positive definite matrices, v_1, v_2 are positive constants such that $v_1 > 1$ and $0 < v_2 < 1$, η is defined as follows:

$$\begin{aligned}\eta =& C_1 M(q) H(e)\dot{e} + C_2 M(q) N^{\delta-1}(e)\dot{e} \\ &- C(q,\dot{q})\dot{q} - G(q) - M(q)\ddot{q}_d\,.\end{aligned} \tag{26}$$

3.2. Radial basis function neural network

Radial Basis Function neural networks, described in Fig. 2, are a type of neural network characterized by a three-layer structure: input layer, hidden layer and output layer. The input layer consists of input variables described by the vector $x = [x_i]^{\mathrm{T}}$, which typically contains system information such as state variables, reference variables, and the differences between them. This vector is then passed through the hidden layer, which is activated by the Gaussian function $\phi(x)$. The hidden layer of an RBF-NN is composed of nodes, each of which utilizes a Gaussian function.

The Gaussian function $\phi(x)$ in (27) contain parameters c, b and input x. The parameter vector c_j determines the position of the centers of the Gauss function when expressed in the coordinate system defined by the input vector x. Therefore,

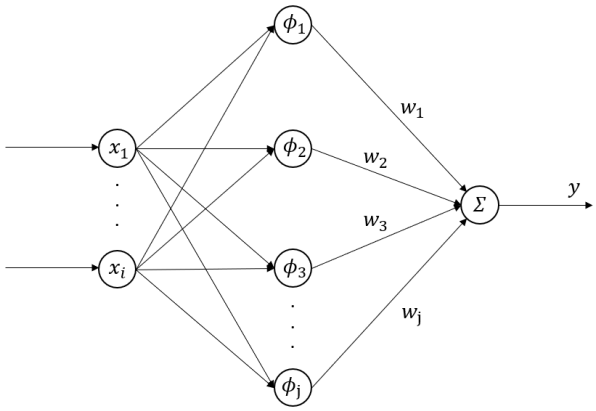

Fig. 2. Radial basis function neural network

the parameter vector c_j is called the center vector and has the same dimensionality as the input vector x. The parameter vector b determines the shape of the Gaussian function, influencing whether it is wide or narrow, steep or gentle. Hence, b is referred to as the radius parameter vector. Each node in the hidden layer will be represented by a center vector c_j and a radius parameter b_j.

An RBF network consists of a matrix c representing the centers, with the number of rows equal to the number of nodes and the number of columns equal to the dimensionality of the input vectors. In other words, each node has a center vector c_j, and these vectors are sequentially arranged into the rows of the matrix c. Similarly, each node is associated with a radius parameter b_j, these parameters are organized into a vector b.

$$\phi_j(x) = \exp\left(-\frac{\|x - c_j\|^2}{2b_j^2}\right). \tag{27}$$

The output value of RBF is calculated by summing the outputs of the hidden layer multiplied by the coefficients. These coefficients are written in vector form with the symbol $\hat{W}$. The formula representing the output y of RBF is as follows:

$$y(t) = \hat{W}^{\mathrm{T}}\phi(x), \tag{28}$$

where

$$\hat{W} = \left[w_1, \ldots, w_j\right]^{\mathrm{T}}. \tag{29}$$

The number of nodes in the RBF-NN is not fixed and can be chosen based on the preferences and expertise of the implementer. Each Gauss function maps a range of input variable values to a range of output values. By employing multiple

Gauss functions, the RBF-NN can cover the entire range of input variable values. Increasing the number of Gauss functions enhances the approximation capability of the network, but it also increases the computational demands.

To utilize the RBF-NN for approximation, identification of the variables in the equation to be approximated is necessary to select them as input for the RBF. This forms the foundation for achieving optimal performance with the RBF. Based on the model and control signals, the inputs to the RBF include the angular error e, angular velocity error $\dot{e}$, reference angle q_d, reference angular velocity $\dot{q}_d$, and reference angular acceleration $\ddot{q}_d$. In vector form, this can be rewritten as:

$$x = [e, \dot{e}, q_d, \dot{q}_d, \ddot{q}_d]^{\mathrm{T}} . \tag{30}$$

Then, the equation that needs to be approximated $D(x)$ is expressed as follows:

$$D(x) = W^{*\mathrm{T}}\phi(x) + \epsilon , \tag{31}$$

where $W^{*\mathrm{T}}$ is the ideal weight matrix, ϵ is the difference in output value of RBF between the using of ideal weight $W^{*\mathrm{T}}$ and estimate weight $\hat{W}^{\mathrm{T}}$. The magnitude of estimation error ϵ impacts the performance of the system when designing the control. The better the approximation provided by the RBF-NN, the less the control signal is influenced by chattering, which is a characteristic phenomenon commonly associated with sliding mode control.

3.3. FxTSMC-RBF-NN controller

During the model calculation, assumptions are introduced to simplify the construction process. Additionally, parameters in the model may not entirely reflect reality. The fixed-time sliding mode controller is designed based on this model, meaning discrepancies in the model can impact control performance. To mitigate uncertainty in the model, the RBF-NN is employed to approximate the functions in the designed control signal. This section introduces a combination of fixed-time SMC controller and RBF-NN.

The structure of the control system is illustrated in Fig. 3. Specifically, the SMC fixed-time controller maintains the joint positions relative to the reference value.

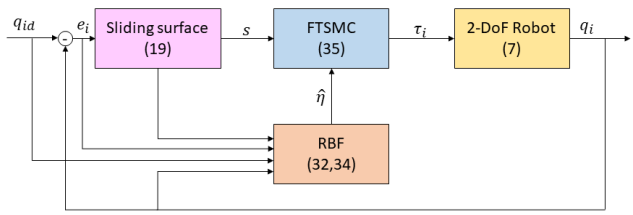

Fig. 3. Control structure

To achieve this, the value η needs to be determined, especially in the presence of model discrepancies and uncertainties. In this context, the RBF-NN plays a crucial role in estimating η. By approximating unknown functions, the RBF-NN predicts the value η and feeds it into the SMC fixed-time controller. Moreover, the RBF-NN possesses the capability to adapt to changes in the system or uncertainties in system parameters through its weight update rule.

The output of the RBF-NN is the estimated value of η and expressed as follows:

$$\hat{\eta} = \hat{W}^{\mathrm{T}}\phi(x). \tag{32}$$

Let's define:

$$\tilde{W} = W^{*} - \hat{W}. \tag{33}$$

To minimize the error ϵ, the update weight rule is used:

$$\dot{\hat{W}} = -\dot{\tilde{W}} = -F_W\phi(x)s^{\mathrm{T}}, \tag{34}$$

where F_W is a positive definite diagonal matrix. Then, the object's control input signal will be rewritten as:

$$\begin{aligned}\tau &= -K_1E^{v_1}(s) - K_2E^{v_2}(s) - K_3\frac{s}{\|s\|} - \hat{\eta}\\ &= -K_1E^{v_1}(s) - K_2E^{v_2}(s) - K_3\frac{s}{\|s\|} - \hat{W}^{\mathrm{T}}\phi(x),\end{aligned} \tag{35}$$

where K_3 is a positive definite diagonal matrix. The parameter matrix K_3 must be selected such that $K_3 > \|\epsilon\|$. This mitigates the impact of estimation errors in the RBF-NN, thereby ensuring the stability of the entire closed-loop system. A detailed proof is provided in the following section.

4. Control design

4.1. Preliminaries

Definition 1: Consider a system:

$$\dot{x} = g(t,x), \quad x(0) = x_0\,, \tag{36}$$

where $x \in R^n$ and $g : R^+ \times R^n \to R^n$ denotes a nonlinear function. The origin of system (36) is considered globally finite-time stable if it is globally asymptotically stable and every solution $x(t)$ of (36) converges to the equilibrium within a finite time interval.

Definition 2: The origin of system (36) is termed fixed-time stable if it meets two criteria: it is globally finite-time stable, meaning every solution $x(t)$ converges

to the equilibrium within a finite time interval and the settling-time function $T(x_0)$ is bounded. Specifically, there exists a constant $T_{max} > 0$ such that $T(x_0) < T_{max}$ for all initial condition x_0.

Lemma 1 [33]: Consider system (36). If there exists a Lyapunov function $V(x)$ such that:

$$\begin{aligned} &V(x) = 0 \Leftrightarrow x = 0, \\ &\dot{V}(x) = -\alpha V^{\frac{m}{n}}(x) - \beta V^{\frac{p}{k}}(x), \end{aligned} \tag{37}$$

where $\alpha > 0, \beta > 0$ and m, n, p, k are positive odd integers satifying $m > n, p < k$. Then, the origin is fixed-time stable and the settling time function $T(x_0)$ is bounded by [21]:

$$T(x_0) < T_{\max} = \frac{1}{\alpha}\frac{n}{m-n} + \frac{1}{\beta}\frac{k}{k-p}. \tag{38}$$

Lemma 2 [33]: Consider a system as follows:

$$\dot{z} = -\alpha \mathrm{sig}^{\frac{m}{n}}(z) - \beta \mathrm{sig}^{\frac{p}{k}}(z), z(0) = z_0, \tag{39}$$

where $\alpha > 0$, $\beta > 0$ and m, n, p, k are positive odd integers satifying $m > n$, $p < k$. Then, the origin of this system is fixed-time stable and the settling time is bounded as described in (38).

Proof [33]: Proposing a candidate Lyapunov function for the system (39):

$$V(z) = \frac{1}{2}z^2. \tag{40}$$

Taking derivative of $V(z)$ with respect to time:

$$\begin{aligned} \dot{V}(z) &= z\dot{z} \\ &= z\left(-\alpha \mathrm{sig}^{\frac{m}{n}}(z) - \beta \mathrm{sig}^{\frac{p}{k}}(z)\right) \\ &= -\alpha \mathrm{sig}^{\frac{m+n}{n}}(z) - \beta \mathrm{sig}^{\frac{p+k}{k}}(z) \\ &= -2\alpha (z^2)^{\frac{m+n}{2n}} - 2\beta (z^2)^{\frac{p+k}{2k}} \\ &= -2\alpha V^{\frac{m+n}{2n}} - 2\beta V^{\frac{p+k}{2k}}. \end{aligned}$$

Applying Lemma 1, we obtain:

$$\begin{aligned} T(z_0) < T_{\max} &= \frac{1}{2\alpha}\frac{2n}{(m+n)-2n} + \frac{1}{2\beta}\frac{2k}{2k-(p+k)} \\ &= \frac{1}{\alpha}\frac{n}{m-n} + \frac{1}{\beta}\frac{k}{k-p}. \end{aligned} \tag{41}$$

Lemma 3 [21]: $\forall\rho$ such that $0 < \rho < 1$, the following inequality holds:

$$\sum_{i=1}^{n} |x_i|^{1+\rho} \geqslant \left(\sum_{i=1}^{n} |x_i|^2\right)^{\frac{1+\rho}{2}}. \tag{42}$$

Lemma 4 [21]: $\forall\rho$ such that $\rho > 1$, the following inequality holds:

$$\sum_{i=1}^{n} |x_i|^{\rho} \geqslant n^{1-\rho} \left(\sum_{i=1}^{n} |x_i|\right)^{\rho}. \tag{43}$$

4.2. Stability

Theorem 1: When employing a control signal (25) in conjunction with a sliding surface (19), the tracking errors of the system (7) converge to zero within a fixed-time. The settling time for this convergence is determined by:

$$\begin{aligned} T_{\text{set}} = T_r + T_s < & \frac{2}{\lambda_{\min}\{K_1\} n^{\frac{1-v_1}{2}} (v_1 - 1)} + \frac{2}{\lambda_{\min}\{K_2\} (1 - v_2)} \\ & + \frac{2}{c_{1i}(1-\alpha)} + \frac{2}{c_{2i}(\beta - 1)}, \end{aligned} \tag{44}$$

where T_r, T_s represent the time required for the system's state trajectory to reach the sliding surface and the time taken for the system's state trajectory to slide along the sliding surface, respectively.

The stability of the system (7) with control input (25) will be proven through Lyapunov stability theory. Let's choose Lyapunov functions as follows:

$$V_1 = \frac{1}{2} s^{\mathrm{T}} M s\,. \tag{45}$$

Firstly, the stability of the FxTSMC controller is considered. Taking derivative of candidate function V_1:

$$\dot{V}_1 = s^{\mathrm{T}} M \dot{s}\,. \tag{46}$$

By using equations (25) and (26) in place of model (7), the expressions are transformed and simplified as follows to obtain:

$$\begin{aligned}
&M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = -K_1 E^{v_1}(s) - K_2 E^{v_2}(s) \\
&- K_3 \frac{s}{\|s\|} - C_1 M(q) H(e)\dot{e} - C_2 M(q) N^{\delta-1}\dot{e} \\
&+ C(q,\dot{q})\dot{q} + G(q) + M(q)\ddot{q}_d \\
\Leftrightarrow &M(q)\left[\ddot{e} + C_1 H(e)\dot{e} + C_2 N^{\delta-1}(e)\dot{e}\right] \\
&= -K_1 E^{v_1}(s) - K_2 E^{v_2}(s) - K_3 \frac{s}{\|s\|} \\
\Leftrightarrow &M(q)\dot{s} = -K_1 E^{v_1}(s) - K_2 E^{v_2}(s) - K_3 \frac{s}{\|s\|} \\
\Leftrightarrow &\dot{V}_1 = s^{\mathrm{T}} M(q)\dot{s} \\
\Leftrightarrow &\dot{V}_1 = s^{\mathrm{T}}\left[-K_1 E^{v_1}(s) - K_2 E^{v_2}(s) - K_3 \frac{s}{\|s\|}\right].
\end{aligned} \tag{47}$$

According to Lemma 3 and 4, the following inequalities can be established:

$$s^T K_1 E^{v_1}(s) \geqslant \lambda_{\min}\{K_1\} \sum_{i=1}^{n} |s_i|^{1+v_1} \geqslant \lambda_{\min}\{K_1\} n^{\frac{1-v_1}{2}} \left(\sum_{i=1}^{n} |s_i|^2\right)^{\frac{1+v_1}{2}}, \tag{48}$$

$$s^T K_2 E^{v_2}(s) \geqslant \lambda_{\min}\{K_2\} \sum_{i=1}^{n} |s_i|^{1+v_2} \geqslant \lambda_{\min}\{K_2\} \left(\sum_{i=1}^{n} |s_i|^2\right)^{\frac{1+v_2}{2}}. \tag{49}$$

From (47), (48), (49) and following [21], conclude that:

$$\dot{V}_1 \leqslant -\lambda_{\min}\{K_1\} n^{\frac{1-v_1}{2}} V_1^{\frac{1+v_1}{2}} - \lambda_{\min}\{K_2\} V_1^{\frac{1+v_2}{2}}. \tag{50}$$

Follow Lemma 1, the interval time for the system's state trajectory to reach the sliding surface, which denoted as T_r, is bound by the following formula:

$$T_r < \frac{2}{\lambda_{\min}\{K_1\} n^{\frac{1-v_1}{2}} (v_1 - 1)} + \frac{2}{\lambda_{\min}\{K_2\}(1 - v_2)}. \tag{51}$$

During the reaching phase, the state trajectory of the system converges towards the sliding surface within a predetermined time frame, as indicated by the aforementioned inequality. Once the system enters the sliding phase, the tracking errors confine to a small region around the origin, denoted as $D_\mu = \{e_i \,|\, |e_i| \leqslant \mu|\}$. The duration required for the tracking errors to move into this region is finite and bounded. In the sliding phase, where $s = 0$, the following condition holds:

$$\dot{e}_i = \begin{cases} -c_{1i}\mathrm{sig}^{1-\mu}(e_i) - c_{2i}\mathrm{sig}^{\delta}(e_i), & |e_i(T_r)| \geqslant \mu, \\ -c_{1i}\left(K_a \mathrm{sig}^{2-\mu}(e_i) + K_b \mu^{|e_i|} e_i\right) - c_{2i}\mathrm{sig}^{\delta}(e_i), & |e_i(T_r)| < \mu. \end{cases} \tag{52}$$

When $|e_i(T_r)| < \delta$, considering the Lyapunov candidate function:

$$V_2 = \frac{1}{2}e_i^2 . \tag{53}$$

Taking the derivative of V_2 with respect to time:

$$V_2 = -c_{1i}K_a|e_i|^{r+1} - c_{1i}K_b\delta^{|e_i|} - c_{2i}|e_i|^{\beta+1} < 0 . \tag{54}$$

Therefore, the system exhibits asymptotic stability at the origin. When the condition $|e_i(T_r)| \geqslant \delta$ is met, as stated in Lemma 1, the duration for which the tracking errors reach the sliding surface is bounded, as described by the following formula:

$$T_s < \frac{2}{c_{1i}(1-\alpha)} + \frac{2}{c_{2i}(\beta-1)}. \tag{55}$$

The total time for the tracking errors to converge to an arbitrary small domain of the origin, $D_\mu = \{e_i \,|\, |e_i \leqslant \mu|\}$, is bounded. This time comprises two components: T_r, the time it takes for the system's trajectory to reach the sliding surface and T_s, the sliding time on the surface. Therefore, the settling time T_{set} of the system is bounded as shown in (44).

Remark 1: The FxTSMC guarantees the convergence of the tracking error through the control signal given in (26), within a predefined time interval determined by the formula in (44). However, a portion of the control input depends on the system model, which can compromise performance when the model contains uncertainties. This limitation highlights the need to incorporate an RBF-NN to address the issue effectively.

Theorem 2: When employing a control signal (35) in conjunction with a sliding surface (19) and update law (34), the tracking errors of the system (7) converge to zero within a fixed-time. The settling time for this convergence is constrained by (44).

A candidate Lyapunov function is considered:

$$V_3 = \frac{1}{2}s^{\mathrm{T}}Ms + \frac{1}{2}\text{trace}\left(\tilde{W}^{\mathrm{T}}F_W^{-1}\tilde{W}\right). \tag{56}$$

Taking derivative of V_3 with respect to time:

$$\dot{V}_3 = s^{\mathrm{T}}M\dot{s} + \text{trace}\left(\tilde{W}^{\mathrm{T}}F_W^{-1}\dot{\tilde{W}}\right). \tag{57}$$

Based on (7) and (35), the following relationship is obtained:

$$\begin{aligned}
&M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = -K_1 E^{\nu_1}(s) \\
&\quad - K_2 E^{\nu_2}(s) - K_3 \frac{s}{\|s\|} - \hat{\eta} \\
\Leftrightarrow &M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = -K_1 E^{\nu_1}(s) \\
&\quad - K_2 E^{\nu_2}(s) - K_3 \frac{s}{\|s\|} - \eta + (\eta - \hat{\eta}) \\
\Leftrightarrow &M(q)\left[\ddot{e} + C_1 H(e)\dot{e} + C_2 N^{\delta-1}(e)\dot{e}\right] \\
&\quad = -K_1 E^{\nu_1}(s) - K_2 E^{\nu_2}(s) - K_3 \frac{s}{\|s\|} + (\eta - \hat{\eta}) \\
\Leftrightarrow &M(q)\dot{s} = -K_1 E^{\nu_1}(s) - K_2 E^{\nu_2}(s) - K_3 \frac{s}{\|s\|} \\
&\quad + (W^{*\top}\phi(x) + \epsilon - \hat{W}^{\mathrm{T}}\phi(x)) \\
\Leftrightarrow &s^{\mathrm{T}} M(q)\dot{s} = s^{\mathrm{T}}[-K_1 E^{\nu_1}(s) - K_2 E^{\nu_2}(s) \\
&\quad - K_3 \frac{s}{\|s\|}] + s^{\mathrm{T}}\epsilon + s^{\mathrm{T}}[W^{*\top} - \hat{W}^{\mathrm{T}}]\phi(x).
\end{aligned} \tag{58}$$

Substituting (34) and (58) into (57), we get:

$$\begin{aligned}
&\dot{V}_3 = s^{\mathrm{T}}\left[-K_1 E^{\nu_1}(s) - K_2 E^{\nu_2}(s) - K_3 \frac{s}{\|s\|}\right] \\
&\quad + s^{\mathrm{T}}\epsilon + s^{\mathrm{T}}(W^{*\top} - \hat{W}^{\mathrm{T}})\phi(x) + \mathrm{trace}\left(\tilde{W}^{\mathrm{T}} F_W^{-1} F_W \phi(x) s^{\mathrm{T}}\right) \\
\Leftrightarrow &\dot{V}_3 = s^{\mathrm{T}}[-K_1 E^{\nu_1}(s) - K_2 E^{\nu_2}(s)] - K_3 \|s\| \\
&\quad + s^{\mathrm{T}}\epsilon + s^{\mathrm{T}}\tilde{W}^{\mathrm{T}}\phi(x) + \mathrm{trace}\left(\tilde{W}^{\mathrm{T}}\phi(x)s^{\mathrm{T}}\right) \\
\Leftrightarrow &\dot{V}_3 = s^{\mathrm{T}}[-K_1 E^{\nu_1}(s) - K_2 E^{\nu_2}(s)] - K_3 \|s\| + s^{\mathrm{T}}\epsilon\,.
\end{aligned} \tag{59}$$

Choosing K_3 such that $K_3 > \|\epsilon\|$, therefore, the following holds:

$$\begin{aligned}
&\dot{V}_3 \leqslant s^{\mathrm{T}}[-K_1 E^{\nu_1}(s) - K_2 E^{\nu_2}(s)] - K_3 \|s\| + \epsilon \|s\| \\
\Leftrightarrow &\dot{V}_3 \leqslant s^{\mathrm{T}}[-K_1 E^{\nu_1}(s) - K_2 E^{\nu_2}(s)] - (K_3 - \epsilon)\|s\| \\
\Rightarrow &\dot{V}_3 \leqslant s^{\mathrm{T}}[-K_1 E^{\nu_1}(s) - K_2 E^{\nu_2}(s)]\,.
\end{aligned} \tag{60}$$

Similar to Theorem 1, using Lemma 3 and 4, this concludes the proof.

5. Simulation results

Assumption: The uncertainties and disturbances affecting the system are time-varying and bounded.

5.1. Trajectory tracking simulation

During the simulation, the scenario involves controlling the 2-DoF robot manipulator to track a trajectory starting from the 5th second and completing the task within 20 seconds, while systematically altering the mass of the second link. This variation in mass serves to represent the uncertainty in the model throughout the simulation. Specifically, the mass of the second link is adjusted to increase within a range from 5.5% to 11%. Fig. 4 illustrates how this mass varies over the course of the simulation.

Fig. 5 displays the reference trajectory and the response trajectory of the end effector, both represented as circles. In Figs. 6 and 7, the desired trajectory is depicted by the red solid line, while the actual trajectory of the end effector is shown by the blue dashed line. It's evident that the FxTSMC-RBF-NN controller

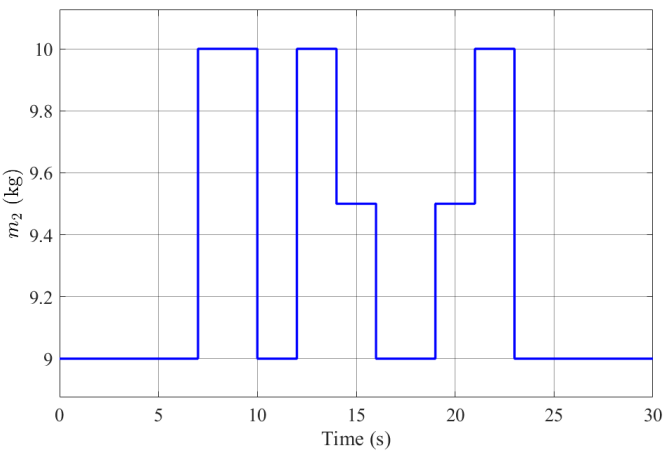

Fig. 4. The variation in mass of second arm

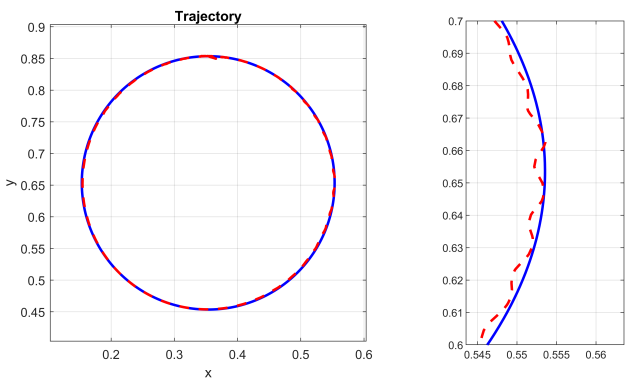

Fig. 5. Trajectory of end effector

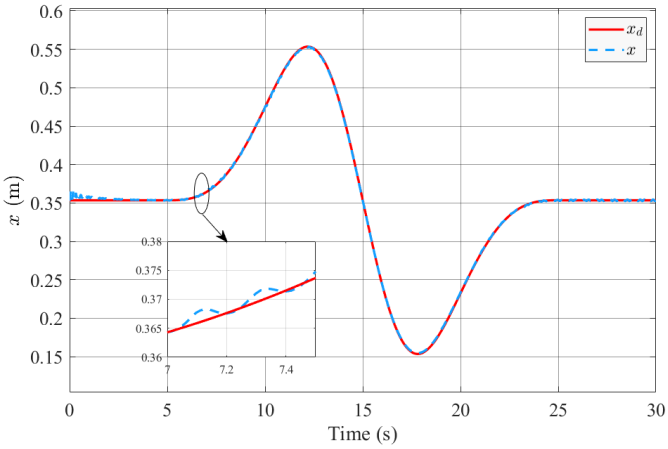

Fig. 6. X-axis position of the end-effector

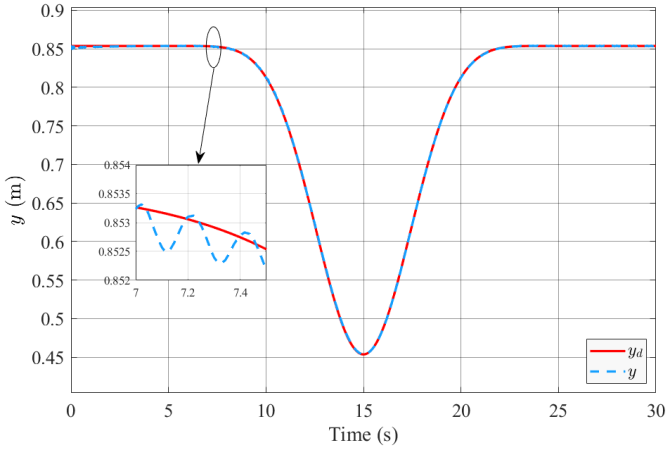

Fig. 7. Y-axis position of the end-effector

effectively maintains tracking control even amidst variations in model parameters. The deviation between the reference position and the feedback position remains minimal, demonstrating robust operation in the presence of uncertainty.

The angular positions of the first arm and the second arm are visually represented in Figs. 8 and 9, respectively, showcasing their effective tracking performance with minimal deviations from their desired trajectories. This highlights the controller's capability to maintain precision in tracking angular positions under various operating conditions. Furthermore, the angular velocities of both links closely approximate their respective reference values, as depicted in Figs. 10 and 11. Despite this, noticeable fluctuations around these desired velocities occur due to suboptimal control parameter settings within the FxTSMC-RBF-NN controller. The complexity arises from the challenge of tuning multiple control parameters,

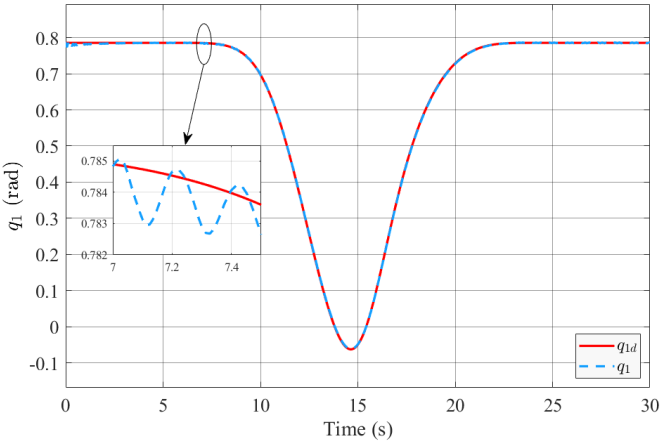

Fig. 8. First arm - angular position

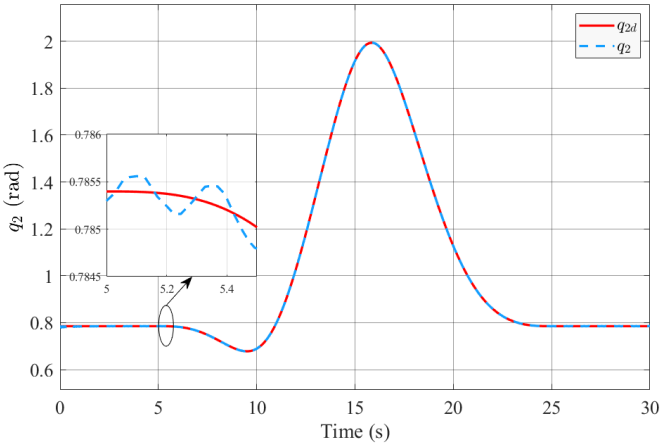

Fig. 9. Second arm - angular position

which makes the adjustment process particularly intricate. While the controller excels in providing robust tracking control in the presence of uncertainties, the difficulty in optimizing its parameters remains a notable drawback. This complexity underscores the need for further refinement in parameter detection processes to fully leverage the controller's strengths in dynamic and uncertain environments.

The control signal is shown in Fig. 12. For varying initial conditions, the control law is computed based on the error between the set point and the system response. As the error increases, the required control effort also increases. Based on the stability guarantees provided in Theorems 1 and 2, adjusting the controller gains allows for modification and improvement of the state response time, as described in equation (44). However, the oscillatory behavior of the control signal is primarily influenced by the weights of the RBF-NN. When these weights are large, the system

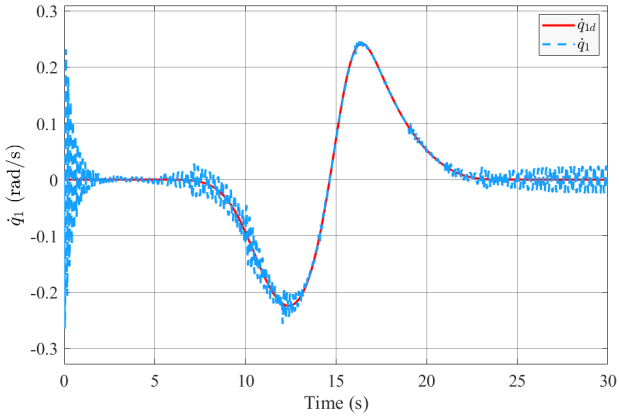

Fig. 10. First arm - angular velocity

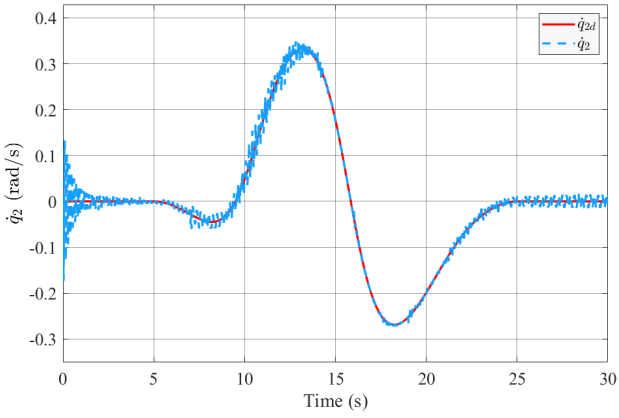

Fig. 11. Second arm - angular velocity

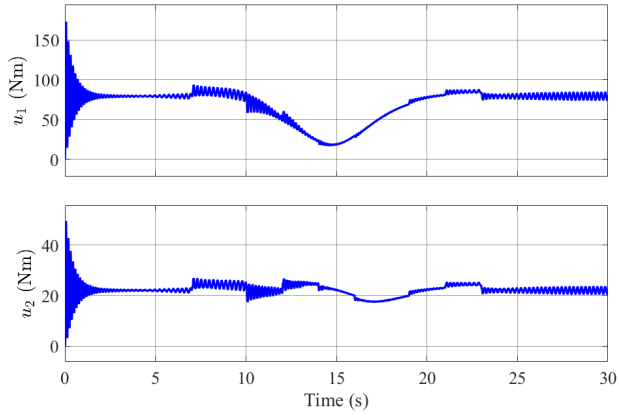

Fig. 12. Control signal

exhibits a rapid response accompanied by oscillations, as illustrated in Fig. 12. Conversely, smaller weight values result in slower adaptation, causing delayed information transmission to the FxTSMC controller. This delay can gradually degrade system performance and potentially lead to instability.

Fig. 13 illustrates the convergence behavior of the sliding surface under the combined FxTSMC-RBF-NN controller. The initial state of the trajectory is marked by a red circle. Each joint's sliding surface gradually approaches the origin and oscillates around it. In contrast to conventional sliding surfaces, where trajectories typically oscillate along a straight line, the FxTSMC sliding surfaces follow curved paths. This behavior arises from the nonlinear structure of the fixed-time sliding surface, which no longer maintains a linear relationship between the state error and its derivative, as is typical in traditional designs.

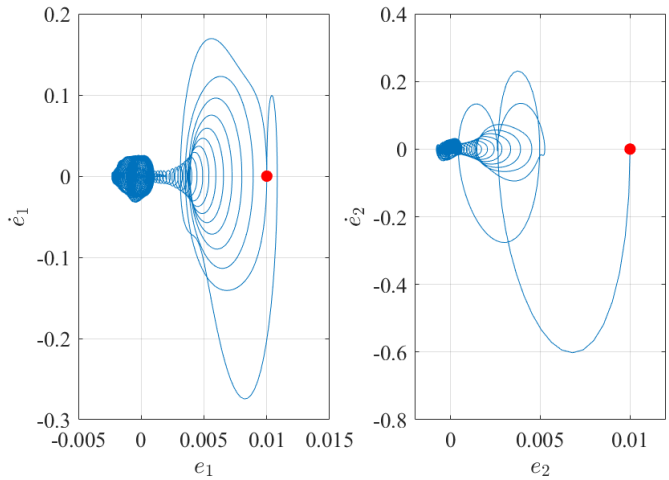

Fig. 13. FxTSMC-RBF-NN sliding surface

5.2. Comparison results

The control performance of the proposed FxTSMC-RBF controller is compared with two other controllers, FxTSMC and SMC, by calculating the position errors of the joints. The root mean square error (RMSE) is utilized to evaluate the performance across methods. For this simulation, the model parameters listed in Table 1 are used, along with percentage variations to facilitate comparisons. Simulations are conducted under scenarios involving changes in mass and/or arm length, and the results are summarized in following tables.

Tables 2 and 3 present the results for a 10% increase and a 10% decrease in the initial parameter values, respectively. Each scenario represents a unique combination of parameter changes. Despite the variations, one consistent observation across

Table 2. RMSE when parameters decreases by 10%

Decrease 10%		FxTSMC-RBF (10^{-3})		FxTSMC (10^{-3})		SMC (10^{-3})	
		e_1	e_2	e_1	e_2	e_1	e_2
m_1, m_2(kg)		1.3	0.47	5.29	3.07	3.89	13.5
	l_1, l_2(m)	1.29	0.46	0.53	3.07	3.94	14.1
m_1(kg)	l_1(m)	1.28	0.51	5.3	3.31	5.04	6.6
m_2(kg)	l_2(m)	1.32	0.43	5.31	2.86	3.9	14.7
m_1, m_2(kg)	l_1, l_2(m)	1.19	0.43	4.92	2.86	4.78	16.9

Table 3. RMSE when parameters increases by 10%

Increase 10%		FxTSMC-RBF (10^{-3})		FxTSMC (10^{-3})		SMC (10^{-3})	
		e_1	e_2	e_1	e_2	e_1	e_2
m_1, m_2(kg)		1.54	0.54	6.1	3.54	3.38	11
	l_1, l_2(m)	1.55	0.56	6.09	3.54	3.23	10.8
m_1(kg)	l_1(m)	1.57	0.51	6.11	3.31	3.71	6.28
m_2(kg)	l_2(m)	1.53	0.6	6.1	3.79	2.69	12.4
m_1, m_2(kg)	l_1, l_2(m)	1.68	0.59	6.52	3.79	3.44	10.8

all cases is the superior performance of the FxTSMC-RBF controller. The position errors of the two arms are consistently smaller when using the proposed controller, highlighting the effectiveness of the RBF-NN in addressing model uncertainties. Furthermore, as shown in Figures 14, 15, 16 and 17, the position trajectories of the two arm angles with the FxTSMC-RBF controller consistently track the desired values more closely compared to the other two controllers, further demonstrating its advantage.

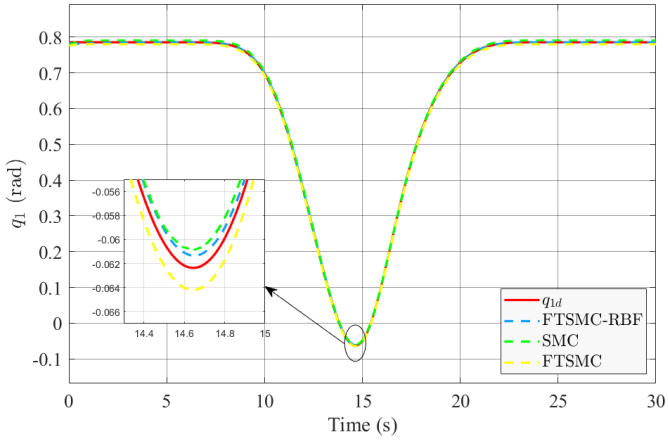

Fig. 14. A decrease of 10% in m_1, m_2, l_1, l_2: response comparison for q_1

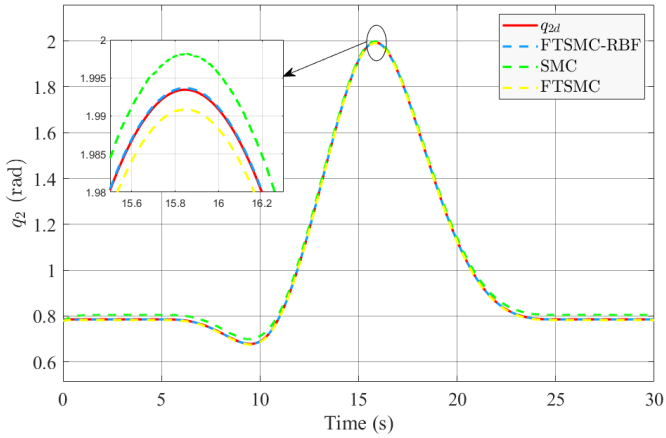

Fig. 15. A decrease of 10% in m_1, m_2, l_1, l_2: response comparison for q_2

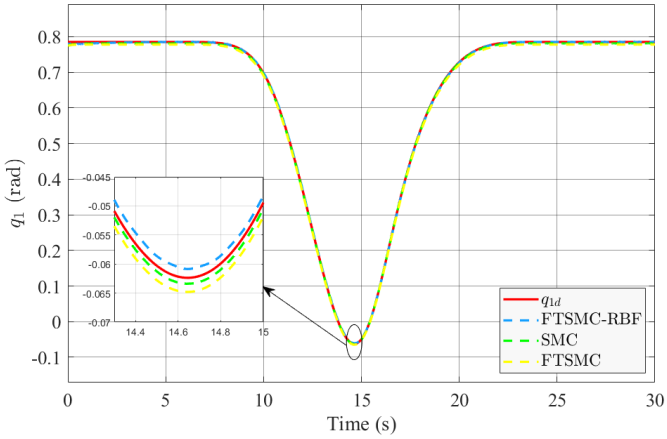

Fig. 16. An increase of 10% in m_1, m_2, l_1, l_2: response comparison for q_1

Remark 2: The comparison results presented in the figures and tables demonstrate the effectiveness of the proposed combined controller. Under conditions of model uncertainty, the controller achieves superior tracking performance, exhibiting reduced tracking errors compared to other control strategies.

The FxTSMC-RBF-NN controller has demonstrated fast tracking capability and improved control performance compared to the other controllers. Its robustness is also confirmed through simulation scenarios involving parameter variations in the system model. However, this increased accuracy comes with a computational cost. Simulation results reveal that the proposed controller exhibits drawbacks in terms of execution time. Specifically, in comparison to the SMC and FxTSMC controllers, which complete the same computational task in 8.28 seconds and 21.56 seconds respectively, the FxTSMC-RBF-NN requires 35.86 seconds. Given the current

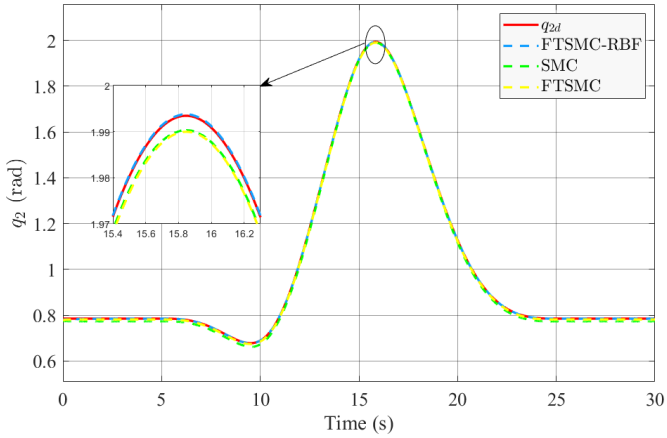

Fig. 17. An increase of 10% in m_1, m_2, l_1, l_2: response comparison for q_2

computation times observed in the simulations, implementing the proposed method on real-world devices is likely to encounter significant challenges in optimizing their operational performance. This represents a significant limitation that must be addressed. Future work will focus on optimizing the computational efficiency of the proposed control strategy.

6. Conclusion

This study presents a novel control approach that integrates fixed-time sliding mode control with a radial basis function neural network, aiming to enhance the performance and robustness of robotic manipulators. The FxTSMC component ensures rapid convergence of system errors within a finite period, significantly improving response speed and control efficiency. The RBF-NN complements this by adapting to dynamic changes and uncertainties, offering robust performance even under varying conditions. Stability analysis, performed through the Lyapunov method, confirms the robustness and reliability of the combined control system. Simulation results on a two-link robotic manipulator highlight the controller's effectiveness in managing uncertainties, maintaining trajectory tracking accuracy, and ensuring robust operation despite changes in system parameters. These findings underscore the potential of the proposed hybrid control strategy to meet the demanding requirements of advanced robotic applications, paving the way for future research and development in this field.

Acknowledgements

This research is funded by Hanoi University of Science and Technology (HUST) under project name T2024-TD-015.

References

[1] J. Cai, J. Deng, W. Zhang, and W. Zhao. Modeling method of autonomous robot manipulator based on D-H algorithm. *Mobile Information Systems*, 2021:4448648, 2021. doi: 10.1155/2021/4448648.

[2] V.A. Nguyen, L. Vermeiren, A. Dequidt, A.T. Nguyen, M. Dambrine, and L. Cung. Takagi-Sugeno fuzzy descriptor approach for trajectory control of a 2-DOF serial manipulator. In *2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pages 1284–1289. 2018. doi: 10.1109/ICIEA.2018.8397907.

[3] K. Jayaswal, D.K. Palwalia, and S. Kumar. Performance investigation of PID controller in trajectory control of two-link robotic manipulator in medical robots. *Journal of Interdisciplinary Mathematics*, 24(2):467–478, 2021. doi: 10.1080/09720502.2021.1893444.

[4] J. Lee, P.H. Chang, B. Yu, and M. Jin. An adaptive PID control for robot manipulators under substantial payload variations. *IEEE Access*, 8:162261–162270, 2020. doi: 10.1109/ACCESS.2020.3014348.

[5] Y. Chen, X. Luo, B. Han, Q. Luo, and L. Qiao. Model predictive control with integral compensation for motion control of robot manipulator in joint and task spaces. *IEEE Access*, 8:107063–107075, 2020. doi: 10.1109/ACCESS.2020.3001044.

[6] A.M. Abdul-Sadah, K.M.H. Raheem, and M. Mahdi S. Altufaili. A fuzzy logic controller for a three links robotic manipulator. *AIP Conference Proceedings*, 2386(1):050026, 2022. doi: 10.1063/5.0066871.

[7] A.T. Nguyen, A. Dequidt, V.A. Nguyen, L. Vermeiren, and M. Dambrine. Fuzzy descriptor tracking control with guaranteed $\mathscr{L}_\infty$ error-bound for robot manipulators. *Transactions of the Institute of Measurement and Control*, 43(6):1404–1415, 2021. doi: 10.1177/0142331220979262.

[8] X. Yu, S. Zhang, Q. Fu, C. Xue, and W. Sun. Fuzzy logic control of an uncertain manipulator with full-state constraints and disturbance observer. *IEEE Access*, 8:24284–24295, 2020. doi: 10.1109/ACCESS.2020.2968925.

[9] Y.P. Pane, S.P. Nageshrao, J. Kober, and R. Babuška. Reinforcement learning based compensation methods for robot manipulators. *Engineering Applications of Artificial Intelligence*, 78:236–247, 2019. doi: 10.1016/j.engappai.2018.11.006.

[10] Y. Hu and B. Si. A reinforcement learning neural network for robotic manipulator control. *Neural Computation*, 30(7):1983–2004, 2018. doi: 10.1162/neco_a_01079.

[11] Y. Hu, W. Wang, H. Liu, and L. Liu. Reinforcement learning tracking control for robotic manipulator with kernel-based dynamic model. *IEEE Transactions on Neural Networks and Learning Systems*, 31(9):3570–3578, 2020. doi: 10.1109/TNNLS.2019.2945019.

[12] D.B. Pham, Q.T. Dao, and T.V.A Nguyen. Optimized hierarchical sliding mode control for the swing-up and stabilization of a rotary inverted pendulum. *Automation*, 5(3):2673–4052, 2024. doi: 10.3390/automation5030017.

[13] R. Li, L. Yang, Y. Chen, and G. Lai. Adaptive sliding mode control of robot manipulators with system failures. *Mathematics*, 10(3):339, 2022. doi: 10.3390/math10030339.

[14] S.J. Abbasi, H. Khan, J.W. Lee, M. Salman, and M.C. Lee. Robust control design for accurate trajectory tracking of multi-degree-of-freedom robot manipulator in virtual simulator. *IEEE Access*, 10:17155–17168, 2022. doi: 10.1109/ACCESS.2022.3149298.

[15] J. Liu and X. Wang. *Advanced Sliding Mode Control for Mechanical Systems*. Springer, 2011.

[16] T.V.A. Nguyen, Q.T. Dao, and N.T. Bui. Optimized fuzzy logic and sliding mode control for stability and disturbance rejection in rotary inverted pendulum. *Scientific Reports*, 14(1):31116, 2024. doi: 10.1038/s41598-024-82471-y.

[17] H.N. Le, M.K. Pham, D.H. Pham, and T.V.A. Nguyen. Trajectory tracking and stabilization of two-wheeled balancing mobile robot with hierarchical and sliding mode control. *International Journal of Dynamics and Control*, 13(1):1–14, 2025. doi: 10.1007/s40435-024-01518-0.

[18] L. Cao, B. Xiao, M. Golestani, and D. Ran. Faster fixed-time control of flexible spacecraft attitude stabilization. *IEEE Transactions on Industrial Informatics*, 16(2):1281–1290, 2020. doi: 10.1109/TII.2019.2949588.

[19] L. Cao, B. Xiao, and M. Golestani. Robust fixed-time attitude stabilization control of flexible spacecraft with actuator uncertainty. *Nonlinear Dynamics*, 100:2505–2519, 2020. doi: 10.1007/s11071-020-05596-5.

[20] Z. Anjum, H. Zhou, S. Ahmed, and Y. Guo. Fixed time sliding mode control for disturbed robotic manipulator. *Journal of Vibration and Control*, 30(7-8):1580–1593, 2024. doi: 10.1177/10775463231165094.

[21] L. Zhang, Y. Wang, Y. Hou, and H. Li. Fixed-time sliding mode control for uncertain robot manipulators. *IEEE Access*, 7:149750–149763, 2019. doi: 10.1109/ACCESS.2019.2946866.

[22] G. Shen, Y. Xia, J. Zhang, and B. Cui. Adaptive fixed-time trajectory tracking control for Mars entry vehicle. *Nonlinear Dynamics*, 102:2687–2698, 2020. doi: 10.1007/s11071-020-06088-2.

[23] H. Hou, X. Yu, L. Xu, K. Rsetam, and Z. Cao. Finite-time continuous terminal sliding mode control of servo motor systems. *IEEE Transactions on Industrial Electronics*, 67(7):5647–5656, 2020. doi: 10.1109/TIE.2019.2931517.

[24] K. Rsetam, Z. Cao, L. Wang, M. Al-Rawi, and Z. Man. Practically robust fixed-time convergent sliding mode control for underactuated aerial flexible jointrobots manipulators. *Drones*, 6(12):428, 2022. doi: 10.3390/drones6120428.

[25] R.F.A. Khan, K. Rsetam, Z. Cao, and Z. Man. ESO based adaptive fixed-time integral sliding mode control for flexible joint robots using singular perturbation method. *Nonlinear Dynamics*, pages 1–20, 2025. doi: 10.1007/s11071-025-11421-8.

[26] Q. Song, S. Li, Q. Bai, J. Yang, A. Zhang, X. Zhang, and L. Zhe. Trajectory planning of robot manipulator based on RBF neural network. *Entropy*, 23(9):1207, 2021. doi: 10.3390/e23091207.

[27] M.D. Duong, V.T. Nguyen, and Q.T. Dao. Adaptive control using radial basis function neural networks for pneumatic artificial muscle systems. *International Journal of Online & Biomedical Engineering*, 20(12):109, 2024. doi: 10.3991/ijoe.v20i12.49159.

[28] Y.K. Lee, M.J.and Choi. An adaptive neurocontroller using rbfn for robot manipulators. *IEEE Transactions on Industrial Electronics*, 51(3):711–717, 2004. doi: 10.1109/TIE.2004.824878.

[29] C. He, F. Zhang, and J. Jiang. Adaptive boundary control of flexible manipulators with parameter uncertainty based on RBF neural network. *Shock and Vibration*, 2020(1):8261423, 2020. doi: 10.1155/2020/8261423.

[30] Viet-Thanh Nguyen, Bao-Long Pham, Thi-Van-Anh Nguyen, Ngoc-Tam Bui, and Quy-Thinh Dao. Sliding mode control of antagonistically coupled pneumatic artificial muscles using radial basis neural network function. *SN Applied Sciences*, 5(9):246, 2023. doi: 10.1007/s42452-023-05475-9.

[31] D. Shang, X. Li, M. Yin, and F. Li. Dynamic modeling and RBF neural network compensation control for space flexible manipulator with an underactuated hand. *Chinese Journal of Aeronautics*, 37(3):417–439, 2024. doi: 10.1016/j.cja.2023.08.003.

[32] J. Liu. *Radial Basis Function (RBF) Neural Network Control for Mechanical Systems*. Springer Science & Business Media, 2013.

[33] X. Wang, J. Guo, and S. Tang. Neural network-based multivariable fixed-time terminal sliding mode control for re-entry vehicles. *IET Control Theory & Applications*, 12(12):1763–1772, 2018. doi: 10.1049/iet-cta.2017.1309.