FOREX Publication
Open Access | Rapid and quality publishing

International Journal of
Electrical and Electronics Research (IJEER)
Research Article | Volume 14, Issue 2 | Pages 271-278 | e-ISSN: 2347-470X

Intelligent Fault-Tolerant Control for 4-DOF Robotic Manipulator Using Sliding Mode Control and RBFNN Against Lumped Uncertainties

Thien-Quang Nguyen[1], and Vi-Do Tran[2]*

[1]*Faculty of Electrical and Electronics Engineering, Ho Chi Minh City University of Technology and Engineering, Ho Chi Minh City, Vietnam; Email: 2341105@student.hcmute.edu.vn*
[2]*Head of Automatic Control Department, Faculty of Electrical and Electronics Engineering, Ho Chi Minh City University of Technology and Engineering, Ho Chi Minh City, Vietnam; Email: dotv@hcmute.edu.vn*

***Correspondence:** *dotv@hcmute.edu.vn*

ABSTRACT- This paper presents an intelligent fault-tolerant controller to eliminate factors affecting the robot of precise working ability. First, the dynamic mathematical model of a 4 degrees of freedom (DOF) robotic manipulator with uncertainty factors will be presented. Next, the proposed controller is based on a sliding mode controller (SMC) to accurately control the trajectory, and radial basis function neuron networks (RBFNN) estimate the lumped uncertainties occurring in the system. Additionally, some simulations will be performed to validate the performance of the designed controller on MATLAB Simulink. Finally, the effectiveness of the proposed method is quantitatively assessed based on the root mean square error (RMSE).

General Terms: Robotics, Control Algorithms.
Keywords: Sliding Mode Control, Radial Basis Function Neuron Networks, Robotic Manipulator, Lumped Uncertainties, Fault-Tolerant Control.

ARTICLE INFORMATION
Author(s): Thien-Quang Nguyen, Vi-Do Tran;
Received: 27/12/25; **Accepted**: 13/04/26; **Published**: 15/06/26;
E- ISSN: 2347-470X;
Paper Id: IJEER250164;
Citation: 10.37391/ijeer.140203
Webpage-link:
https://ijeer.forexjournal.co.in/archive/volume-14/ijeer-140203.html

Publisher's Note: FOREX Publication stays neutral with regard to jurisdictional claims in Published maps and institutional affiliations.

1. INTRODUCTION

During the working process, humans have realized that there are jobs that are repetitive, hard, require high precision, dangerous, toxic working environments, etc. At that time, manipulator robotics had been researched and developed, one of the machines that had been of interest since the 1950s to replace humans in doing those jobs [1]. To control a manipulator robot to work effectively, manufacturers need to clearly understand the problems they will encounter, such as kinematics, dynamics, and trajectory planning. Accurate calculation of the robotic dynamics remains a significant challenge due to unmodeled components and parametric uncertainties, which render the system highly nonlinear and complex [2]. Therefore, when controlling a manipulator robot, it is necessary to face the challenge of high nonlinearity, dynamic errors, noisy working environments, sensor interference, friction, loss of efficiency, and even design errors by the manufacturer. Moreover, the slow delay time makes the control signal unresponsive when a sudden error occurs.

In order to handle the problems of faults that may occur during the operation of the system, fault-tolerant controllers (FTC) have been interested in and studied in recent years [3]. In essence, FTC is divided into two types: Passive FTC and Active FTC [3-4]. Meanwhile, PFTC is a control algorithm that is often designed for systems that rarely have faults or do not place emphasis on continuous operation. Therefore, this algorithm is not effective when unexpected faults suddenly occur. This is dangerous for the system as well as for people during the operation. PFTC even makes the designer fall into a passive state in the design [3-4].

The development of the control architecture is driven by critical design considerations, specifically focusing on system flexibility, reliability, stability, and safety [5]. Then, the AFTC will be a more suitable direction [4,6]. AFTC has attracted researchers around the world for about two decades. According to the author's experience, there are few studies on AFTC in Vietnam [6]. AFTCs are different from PFTCs in that they have the ability to update the control signal by estimating the uncertainties occurring in the system. But the designers could face the delay time, increase computation volume, and increase data storage capacity even when the system operates without faults. The use of PFTC or AFTC will depend on the system and the type of fault. In manipulator robotic control, AFTC is often evaluated better than PFTC in terms of efficiency when the fault is accurately estimated but does not require reconfiguration of the controller [3]. Therefore, AFTC will be the target of this paper.

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of Electrical and Electronics Research (IJEER)
Research Article | Volume 14, Issue 2 | Pages 271-278 | e-ISSN: 2347-470X

The basic difference between PFTC and AFTC is that AFTC will have an additional fault estimation part [4]. When operating, the fault diagnosis unit will estimate the fault that occurs and feed this information to the main controller. At this point, the controller can eliminate the fault or reconfigure the controller to cope with the fault that is occurring in the system [6]. There are many ways to build an AFTC controller.

Computed torque control (CTC) is a feedback controller of the system of dynamic components to evaluate the nonlinear component [7]. However, when using this controller, it is necessary to know the exact dynamics of the system. Next, the sliding mode controller (SMC) is a famous controller in highly nonlinear systems. However, the chattering phenomenon that occurs in the system reduces efficiency and safety [8]. The fuzzy controller is smart and effective when it can update the controller parameters, but when designing, the designer must have experience and be an expert in that system. Without this factor, this method is difficult to construct [9]. In addition, some famous filters that can be used to filter uncertainties include the Kalman Filter [10], Linear or nonlinear extended state observer [11], etc. These systems are effective but have the disadvantage of taking time to calculate, even when there is no uncertainty. In addition, it requires the designer to know the exact dynamics of the system, because it relies on the factors of dynamics to design. In addition, some popular methods include GA-PID [12], back-stepping control [13], LQG [14], neuron networks [15-16], model-based fault diagnosis algorithms [17], active-assistive control [18], distributed convex optimization [19], adaptive control [20-21], robust output feedback stabilization [22], etc. All methods can evaluate the fault and provide that information to the main controller. In this paper, AFTC is presented based on radial basis function (RBFNN). The RBFNN is a type of intelligent control, which is effective and simple to construct [16].

Different from the author of the previous research on FTC. Linear extend state observer (LESO) is capable of estimating many types of uncertainties, such as friction, model fault, actuator fault, and external uncertainty, by calling the sum of these uncertainties the lumped uncertainties occurring in the system. But it requires the designer to have clear knowledge and understanding of dynamic models of the system. To solve that problem, a radial basis function neuron network (RBFNN) will be presented in this paper. It is also capable of solving multiple aggregations similar to LESO, but is easier to build when it does not need to know information about the system of dynamics [16]. This demonstrates the strong and flexible application ability of the proposed controller.

In this paper, an intelligent fault-tolerant controller based on SMC and RBFNN will be investigated to control a 4-degree-of-freedom robotic, taking into account lumped uncertain factors including friction and loss of performance in the motors. To demonstrate the effectiveness of the proposed controller, we have conducted some AFTC simulations and compared it with PFTC, which is SMC. When there is no uncertainty, and when there is uncertainty, to see the ability of the proposed controller to withstand that factor.

The structure of this paper is arranged as follows. The dynamics of the robotic manipulator considering the uncertainty factor are presented in *section 2*. Next, the proposed controller based on SMC and RBFNN will be presented in *section 3*. Then, *section 4* will show some implementations and simulations to demonstrate its superiority. Additionally, *section 5* provides that the effectiveness of the proposed method is quantitatively assessed using the root mean square error (RMSE) metric. Finally, some conclusions and future directions of the research are presented.

2. PROBLEM FORMULATION

2.1. The Dynamic of Robotic Manipulator

In [4,15], the dynamic of robotic manipulator is represented as the following as *eq. (1)*;

$$M(\theta)\ddot{\theta} + C(\theta,\dot{\theta})\dot{\theta} + G(\theta) = \tau \tag{1}$$

where $\ddot{\theta}, \dot{\theta}, \theta \in \mathbb{R}^{n\times 1}$ are acceleration, velocity, and position; $\tau \in \mathbb{R}^{n\times 1}$ are the control signals, the mean torque at the joints; $M(\theta) \in \mathbb{R}^{n\times n}$ is an inertia term; $C(\theta,\dot{\theta}) \in \mathbb{R}^{n\times n}$ denotes the Coriolis and Centrifugal term; $G(\theta) \in \mathbb{R}^{n\times 1}$ is the gravity term; n is the number of degrees of freedom. In the ideal operating condition of the robotic manipulator, *eq. (1)* has been fully presented. But the operation of the robotic manipulator can have many fault factors that are harmful to the system. For example, the aging of the motor leads to loss of performance, environmental interference, friction, unwanted changes in dynamics, even mistakes from the designer [3][4], etc. As the introduction has mentioned, this paper will solve that type of fault, which is the loss of performance in the motor operating at each rotating joint and friction. For the purpose of robust control, all non-ideal factors, including motor degradation and complex friction, are aggregated into a term called lumped uncertainty, $d_t \in \mathbb{R}^{n\times 1}$. The following *eq. (2)* extends the faults mentioned above based on *eq. (1)*;

$$M(\theta)\ddot{\theta} + C(\theta,\dot{\theta})\dot{\theta} + G(\theta) + d_t = \tau \tag{2}$$

where $d_t \in \mathbb{R}^{n\times 1}$ is the lumped uncertainty, the faults in the system, $d_t = \tau_{\text{fric}} - (\alpha - 1)\tau \in \mathbb{R}^{n\times 1}$; $\tau_{\text{fric}} \in \mathbb{R}^{n\times 1}$ are viscous friction and static friction, the friction factors; αdenotes the performance of the motors, $0 \le \alpha \le 1$.

Property 1 [4,6]: $M(\theta) \in \mathbb{R}^{n\times n}$ is defined as a diagonal matrix with positive definite entries and follows as:

$$0 < \delta(M(\theta)) \| M(\theta) \| (M(\theta))_{0_{max_{min}}} \tag{3}$$

where δ_0 is a known positive constant, δ_{min} and δ_{max} are the minimum and maximum values of the inertia.

Property 2 [4,6]: $\dot{M}(\theta) - 2C(\theta,\dot{\theta})$ is a skew-symmetric matrix, which implies that for any vector $A \in \mathbb{R}^{n\times 1}$ follow as below:

$$A^T\big(\dot{M}(\theta) - 2C(\theta,\dot{\theta})\big)A = 0 \tag{4}$$

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of Electrical and Electronics Research (IJEER)
Research Article | Volume 14, Issue 2 | Pages 271-278 | e-ISSN: 2347-470X

Property 3 [4,6]: The matrix $C(\theta,\dot{\theta}) \in \mathbb{R}^{n\times n}$ satisfies the inequality $\|C(\theta,\dot{\theta})\| \leq c_b\|\dot{\theta}\|$, where c_b is a positive constant.

Property 4 [4,6]: The matrix $G(\theta) \in \mathbb{R}^{n\times 1}$ satisfies the inequality $\|G(\theta)\| \leq g_b$, where g_b is a positive constant.

2.2. Friction Factors

Assumption 1 [4,6]: Frictional effects, comprising both viscous and static components, are represented in *Equation (5)*;

$$\tau_{\text{fric}} = b\dot{\theta} + c\,tanh\left(\frac{\dot{\theta}}{\psi}\right) \quad (5)$$

where $b = diag(b_1, b_2, \ldots, b_n) \in \mathbb{R}^{n\times n}$ is a diagonal matrix of viscous friction; $c = diag(c_1, c_2, \ldots, c_n) \in \mathbb{R}^{n\times n}$ is a diagonal matrix of static friction; $\psi = [\psi_1, \psi_2, \ldots, \psi_n]^T \in \mathbb{R}^{n\times 1}$ is a positive vector and $tanh\left(\frac{\dot{\theta}}{\psi}\right) = \left[tanh\left(\frac{\dot{\theta}_1}{\psi_1}\right), tanh\left(\frac{\dot{\theta}_2}{\psi_2}\right), \ldots, tanh\left(\frac{\dot{\theta}_n}{\psi_n}\right)\right]^T \in \mathbb{R}^{n\times 1}$.

Assumption 2 [4,6]: The lumped uncertainty $d_t \in \mathbb{R}^{n\times 1}$ is assumed to be unknown but bounded by a positive constant ε such that $\|d\|_t \leq \varepsilon$.

3. PROPOSED CONTROL

3.1. Control Description

This paper proposes an intelligent control method based on a nonlinear controller, as shown in *figure 1*, which is a SMC combined with RBFNN. While SMC controls the robotic manipulator to work accurately according to the initial requirements, RBFNN performs the estimation of unexpected faults in the system.

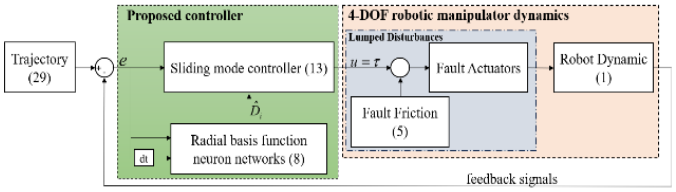


Figure 1. Diagram of the system using SMCRBFNN

The combination of the two factors above helps the system to work more stably and safely; the proposed control belongs to the AFTC.

3.2. Radial Basis Function Neuron Networks

The fault model that occurs in the robotic manipulator system during operation is a challenge that is difficult to identify [16]. Many types of faults and variations are difficult to identify, which are harmful to the system. Therefore, using neural networks to estimate the exact error occurs without knowing the fault model clearly. In this study, we use RBFNN to estimate the lumped uncertainties, which is called an intelligent strategy. The RBFNN consists of three layers: an input layer, a hidden layer, and an output layer.

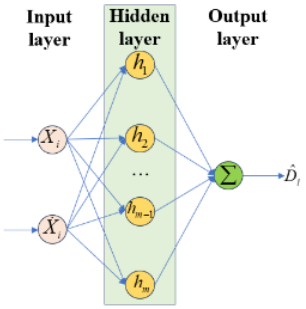


Figure 2. Structure of RBFNN

In *figure 2*, the position vectors X_j and $\dot{X}_j$ are input to the network. At the hidden layer, neurons are activated by a radial basis function. The hidden of the system is presented by *eq. (6);*

$$h_j = exp\left(-\frac{\|X_j-\mu_{1j}\|^2+\|\dot{X}_j-\mu_{2j}\|^2}{b^2}\right) \quad (6)$$

where b is the width of the Gaussian function, μ_{1j}, μ_{2j} are the center points; $H_i = [h_1, h_2, \ldots, h_m]^T$ denotes the output vector of the hidden layer with m the number of neurons. W_i denotes the optimal weight matrix. $\|\sigma\|$ is the approximation error, $\|\sigma\| \leq \bar{\sigma}$. Thus, the lumped uncertainties D_i can be presented:

$$D_i = W_i^T H_i + \sigma_i \quad (7)$$

The output $\widehat{D}_i$ of the RBFNN to estimate D_i as below:

$$\widehat{D}_i = \widehat{W}_i^T H_i \quad (8)$$

where $\widehat{W}_i$ is a trained weight matrix. And $\dot{\widehat{W}}_i$, given by:

$$\dot{\widehat{W}}_i = \Delta\left(H_i s_i^T - \rho\|s_i\|\widehat{W}_i\right) \quad (9)$$

where Δ is a positive definite gain matrix, and ρ is a positive definite gain number.

3.3. Sliding Mode Controller

To design this controller, first set the tracking error as $e = \theta - \theta_d$ with $e \in \mathbb{R}^{n\times 1}$; then $\dot{e} \in \mathbb{R}^{n\times 1}$ is the first derivative of e; $\ddot{e} \in \mathbb{R}^{n\times 1}$ is the second derivative of tracking error; $\theta_d \in \mathbb{R}^{n\times 1}$ is the desired state. The sliding surface $s \in \mathbb{R}^{n\times 1}$ is assumed to be differentiable, and is selected as:

$$s = \dot{e} + \lambda e \quad (10)$$

where $\lambda = diag([\lambda_1, \lambda_2, \ldots, \lambda_n]) \in \mathbb{R}^{n\times n}$ is defined as a diagonal matrix with positive definite entries. The velocity of the sliding surface is calculated as *eq. (11)* below:

$$\dot{s} = \ddot{e} + \lambda\dot{e} \quad (11)$$

A detailed development of the sliding surface velocity is shown below when $\ddot{e}$ is substituted from *equation (2)*;

$$\dot{s} = \left(M(\theta)\right)^{-1}\left(\tau - C(\theta,\dot{\theta})\dot{\theta} - G(\theta) - d_t\right) - \ddot{\theta}_d + \lambda\dot{e} \quad (12)$$

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of Electrical and Electronics Research (IJEER)
Research Article | Volume 14, Issue 2 | Pages 271-278 | e-ISSN: 2347-470X

In [6], the law control for sliding mode control will be chosen as *equation (13)*, which includes two factors, such as equilibrium and robust terms:

$$u = u_{eq} + u_r \tag{13}$$

where $u = \tau \in \mathbb{R}^{n\times 1}$; u_{eq} is the equilibrium part; u_r is the robust part. There are similar *equations (14)* and *(15)* below:

$$u_{eq} = M(\theta)(\ddot{\theta}_d - \lambda\dot{e}) + C(\theta,\dot{\theta})(\dot{\theta}_d - \lambda e) - ks + G(\theta) \tag{14}$$

$$u_r = -\eta \text{sign}(s) \tag{15}$$

where $\eta = diag([\eta_1, \eta_2, \dots, \eta_n]) \in \mathbb{R}^{n\times n}$ represents a diagonal matrix that is positive definite. The control will be satisfied when the inequality $\eta_i \geq \|d_t\|$ is chosen.

The sign function is presented as follows $sign(x) = [sign(x_1), sign(x_2), \dots, sign(x_n)]^T \in \mathbb{R}^{n\times 1}$; k is a positive constant.

Proof. A positive definite Lyapunov function is considered and is chosen to follow as *equation (16)* below:

$$V_1 = \frac{1}{2}s^T M(\theta)s \tag{16}$$

Taking the derivative of the above Lyapunov function yields *equation (17)*

$$\dot{V}_1 = s^T M(\theta)\dot{s} + \frac{1}{2}s^T\dot{M}(\theta)s \tag{17}$$

Substituting *eq. (12)* into *eq. (17)* will give the details of the above Lyapunov derivative as follows

$$\dot{V}_1 = s^T M(\theta)\begin{pmatrix}(M(\theta))^{-1}(\tau - C(\theta,\dot{\theta})\dot{\theta} - G(\theta)\\ -d_t) - \ddot{\theta}_d + \lambda\dot{e}\end{pmatrix} + \frac{1}{2}s^T\dot{M}(\theta)s \tag{18}$$

Then, substituting *equations (13)-(15)* into *equations (18)*

$$\dot{V}_1 = s^T M(\theta)\begin{pmatrix}(M(\theta))^{-1}(M(\theta)(\ddot{\theta}_d - \lambda\dot{e}) + C(\theta,\dot{\theta})(\dot{\theta}_d - \lambda e) - ks +\\ G(\theta) - \eta\text{sign}(s) - C(\theta,\dot{\theta})\dot{\theta} - G(\theta) - d_t) - \ddot{\theta}_d + \lambda\dot{e}\end{pmatrix} + \frac{1}{2}s^T\dot{M}(\theta)s$$

$$= s^T M(\theta)\big((M(\theta))^{-1}(-C(\theta,\dot{\theta})s - ks - \eta\text{sign}(s) - d_t)\big) + \frac{1}{2}s^T\dot{M}(\theta)s$$

$$= s^T(-ks - \eta sign(s) - d_t) + \tfrac{1}{2}s^T\big(\dot{M}(\theta) - 2C(\theta,\dot{\theta})\big)s \tag{19}$$

Based on *Property 2* and $\eta_i \geq \|d_t\|$, *equation (19)* will be transformed as *equation (13)*.

$$\dot{V}_1 = s^T(-ks - \eta sign(s) - d_t) \leq -k\|s\|^2 \leq 0 \tag{20}$$

The stability of the controller has been demonstrated according to Lyapunov theory.

3.4. Proposed Controller

The combination of SMC and RBFNN above will be the proposed controller in this paper, while SMC helps the system to perform tasks according to the setpoint signal, RBFNN will estimate the fault occurring in the system during operation. The proposed control $u \in \mathbb{R}^{n\times 1}$ law is shown as:

$$u = u_{eq} + u_r + \widehat{D}_i = M(\theta)(\ddot{\theta}_d - \lambda\dot{e}) + C(\theta,\dot{\theta})(\dot{\theta}_d - \lambda e) - ks + G(\theta) - \eta sign(s) + \widehat{W}_i^T H_i \tag{21}$$

Assumption 3: Lumped uncertainties D_i affect the robotic manipulator, at this point RBFNN will handle the parameter W_i; $\|W_i\| \leq \bar{W}_i$ is bound, then the $\widehat{D}_i$ is estimated. If the sliding surface $\|s_i\| \leq \bar{s}_i$ is bound and follows *equation (22)*, the proposed controller is stable.

$$-\frac{\rho s_i\|\tilde{W}_i\|^2}{4} \leq -s_i\bar{\sigma} - 2s_i\widetilde{W}_i^T H_i \tag{22}$$

Proof. A positive definite Lyapunov function is considered follow as *equation (23)* below:

$$V_2 = \frac{1}{2}{s_i}^T M(\theta)s_i + \frac{1}{2}tr\big(\widetilde{W}_i^T\Delta^{-1}\widetilde{W}_i\big) \tag{23}$$

where $\widetilde{W}_i = W_i - \widehat{W}_i$ is the error weight matrix, $\dot{\widetilde{W}}_i = -\dot{\widehat{W}}_i$. The derivative of V_2 gives

$$\dot{V}_2 = {s_i}^T M(\theta)\dot{s}_i + \frac{1}{2}{s_i}^T\dot{M}(\theta)s_i + tr\left(\widetilde{W}_i^T\Delta^{-1}\dot{\widehat{W}}_i\right) = {s_i}^T M(\theta)\dot{s}_i + \frac{1}{2}{s_i}^T\dot{M}(\theta)s_i - tr\left(\widetilde{W}_i^T\Delta^{-1}\dot{\widehat{W}}_i\right) \tag{24}$$

Substituting *equations (7), (9), (12)* and *(21)* into *(24)* gives

$$\dot{V}_2 = {s_i}^T(-ks_i - \eta\text{sign}(s_i)) - {s_i}^T\sigma_i - 2{s_i}^T\widetilde{W}_i^T H_i + \frac{1}{2}{s_i}^T\big(\dot{M}(\theta) - 2C(\theta,\dot{\theta})\big)s_i + \|s_i\|\,tr\big(\widetilde{W}_i^T(W_i - \widetilde{W}_i)\big) \tag{25}$$

According to the Cauchy-Schwarz inequality [23], the following inequality equation can be satisfied:

$$tr\big(\widetilde{W}_i^T(W_i - \widetilde{W}_i)\big) \leq \|\widetilde{W}_i\|\|W_i\| - \|\widetilde{W}_i\|^2 \tag{26}$$

Substituting *eq. (26)* into *(25)* gives:

$$\dot{V}_2 \leq s^T(-ks_i - \eta sign(s_i)) + + \tfrac{1}{2}{s_i}^T\big(\dot{M}(\theta) - 2C(\theta,\dot{\theta})\big)s_i - {s_i}^T\sigma_i - 2{s_i}^T\widetilde{W}_i^T H_i - \rho\|s_i\|\left(\|\widetilde{W}\|^2 - \|\widetilde{W}_i\|\|W_i\|\right) \tag{27}$$

Applying *Assumption 3* and Properties 1 to 4 of the robotic manipulator to *equation (27)* gives:

$$\dot{V}_2 \leq -k\|s_i\|^2 - \eta|s_i| - \rho\|s_i\|\left(\tfrac{1}{2}\|W_i\| - \|\widetilde{W}_i\|\right)^2 \leq 0 \tag{28}$$

Therefore, the proposed controller in this paper has been proven to be stable according to Lyapunov theory.

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of Electrical and Electronics Research (IJEER)
Research Article | Volume 14, Issue 2 | Pages 271-278 | e-ISSN: 2347-470X

4. SIMULATION TEST

4.1. Setup Simulation

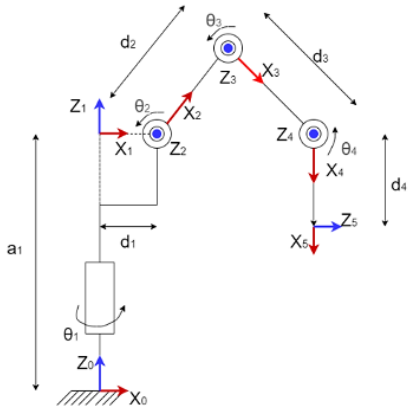


Figure 3. The 4 degrees-of-freedom robotic manipulators

The kinematic structure of the 4-degree-of-freedom robotic manipulators that this paper wants to simulate is shown in *figure 3*. Give some simulations based on MATLAB Simulink 2018b software and use the ode14x calculation method. Simulation time is 30 seconds with a sample time of 0.01 seconds. The physical parameters of the system are referred to in [6]. The tracking trajectory $x_d = [x_d, y_d, z_d]^T$ for the robot is shown below

$$x_d = \begin{bmatrix} 0.38 \\ 0.14\,cos(2\pi f_d t) \\ 0.4 + 0.14\,cos(2\pi f_d t) \end{bmatrix} \quad (m) \qquad (29)$$

where f_d is the frequency, which is chosen to be equal to 0.1 Hz. *Equation (5)* shows the effect of friction on the system, and the parameters chosen are $b = \text{diag}([5,5,5,5])$, $c = 10\text{diag}([5,5,5,5])$ and $\psi = 0.05\text{diag}(1,1,0.5,7)$. This friction factor is assumed to occur at the 10th second. Another known fault assumption is also presented, which is the loss of efficiency in the motors, and they are chosen as follows.

$$\alpha = \begin{cases} 1, & t \le 20 \\ 0.5, & t > 20 \end{cases} \qquad (30)$$

4.2. Controllers For Efficiency Confirmation

Using two controllers: PFTC is SMC, and AFTC is SMCRBFNN to validate the performance of the designed controller. The SMC control law is designed as *equation (13)*, and SMCRBFNN as *equation (21)*, the parameters of the SMC controller of these two controllers are completely the same to demonstrate the difference. And they are shown in the following *table 1* below:

Table 1. Parameters of controllers

Controllers	Parameters
PFTC: SMC	$\lambda = 6.5\text{diag}([6.5,6.5,6,6]);$ $\eta = 6.5\text{diag}([6.5,6.5,6,6]);$ $k = 3.5\text{diag}([3,3,3,2])$
AFTC: SMCRBFNN	$\lambda = 6.5\text{diag}([6.5,6.5,6,6]);$ $\eta = 6.5\text{diag}([6.5,6.5,6,6]);$ $k = 3.5\text{diag}([3,3,3,2]); b = 10;$ $\mu_{1j} = \mu_{2j} = 0.5; \rho = 0.15; \Delta = 0.4; m = 100;$

4.3. Results

First, *figure 3* will present the response signals of the two controllers, SMC and SMCRBFNN, compared with the initial tracking signal on the robotic manipulator system. The black line is the set tracking signal, the blue line is the SMC response signal, and the red line is the SMCRBFNN response signal. When the time is less than 10 seconds, it means that freely fault has occurred.

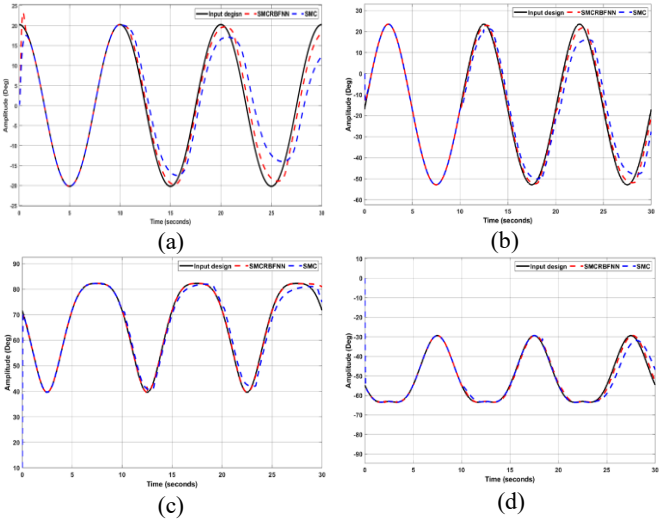


Figure 4. Input design and output responses of the controllers in (a) Joint 1, (b) Joint 2, (c) Joint 3, (d) Joint 4

From the 10th to the 20th second, the assumed friction fault has occurred. From the 20th to the 30th second, the engine performance loss fault has occurred. This means that the lumped uncertainties have been activated. In *figure 4*, the blue line response signal starts to deviate from the black line at the 10th second. That is, the friction error is assumed to be activated. Meanwhile, the red line also experiences the same phenomenon, but the amplitude of the change still tends to follow the black line, showing that it is more positive evidence than the blue line.

The difference between the proposed controller is clearer from the 20th second, as this is the point where the motor efficiency degradation error starts to kick in. The response of the blue line starts to decrease significantly as the compound noise occurs. Meanwhile, the red line still shows a more positive signal as it can still track the set signal, although it is easy to see that it still has some errors. And they occur at all four joints. To easily see the working ability of the proposed controller, *figure 5* will be presented in relation to the error between the tracking signal and the response signal of the two controllers SMC, and SMCRBFNN, above. In the first 10 seconds, when the assumed fault is not activated, *figure 5* shows that there is no difference in error between the two controllers under consideration. This can be seen by looking back a little bit in *figure 4*. But from the 10th second when the friction fault occurs and the 20th second when the motor efficiency loss occurs, and then the lumped uncertainties occur, the difference between the proposed controller and the SMC controller can be seen.

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of Electrical and Electronics Research (IJEER)
Research Article | Volume 14, Issue 2 | Pages 271-278 | e-ISSN: 2347-470X

Not only does the SMC have a larger error than the RBFNN, but the error also increases when other faults occur in the system. It can be concluded that the proposed controller is a better solution, although there is still an error. To achieve that difference, *Figure 6* will show the ability to estimate errors occurring in the system of RBFNN, which helps the system to resist factors that are harmful to the system. When there is no uncertainty, RBFNN with the selected parameters seems to have a large sensitivity that makes it fluctuate around the zero point, which is a bad point of it.

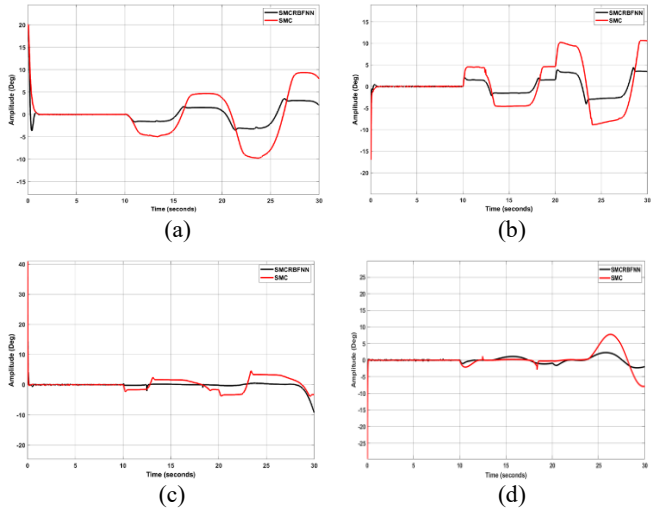
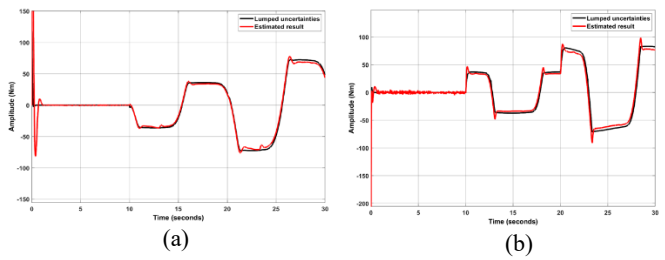


Figure 5. Tracking performance of all joints: (a) Joint 1, (b) Joint 2, (c) Joint 3, and (d) Joint 4

Although the improvement may not appear highly significant under normal operating conditions, the proposed method demonstrates robust and reliable performance when subjected to unexpected perturbations. Specifically, it maintains accurate estimation even in the presence of a sudden friction uncertainty occurring from the 10^{th} second onward and continues to estimate effectively despite an additional reduction in engine efficiency introduced at the 20^{th} second.

The estimation fault serves as a critical and insightful indicator that reflects the degree to which the proposed model is capable of accurately tracking and compensating for real external uncertainties affecting the system's dynamics. This metric is particularly valuable because it not only quantifies the deviation between the estimated and actual uncertainties but also provides a direct measure of the model's adaptability and responsiveness in dynamic operating environments.



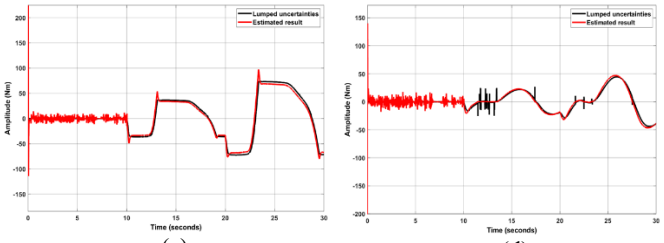


Figure 6. Estimation results in (a) Joint 1, (b) Joint 2, (c) Joint 3, (d) Joint 4

A consistently low estimation error across various scenarios implies that the model is not only learning and generalizing well under nominal conditions but is also capable of adapting to abrupt changes and unexpected faults that may occur in real-time operations. Such consistency under a wide range of conditions—including abrupt uncertainties, friction variations, and system inefficiencies—reinforces the reliability and robustness of the proposed control strategy.

Therefore, the estimation fault is not just a performance metric, but a meaningful representation of the model's real-world feasibility, especially when applied to safety-critical or mission-dependent systems operating in uncertain, time-varying, or highly nonlinear environments, such as unmanned aerial vehicles (UAVs), industrial robotic manipulators, or autonomous driving platforms.

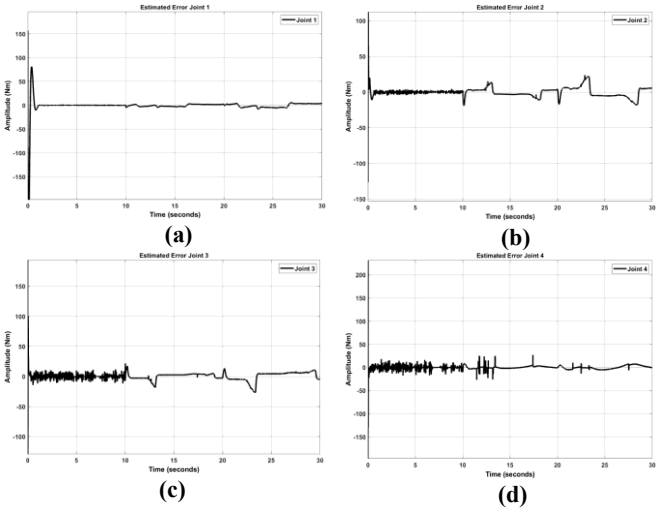


Figure 7. Error estimation results in (a) Joint 1, (b) Joint 2, (c) Joint 3, (d) Joint 4

To further emphasize and gain deeper insights into the effectiveness and reliability of the RBFNN in accurately estimating time-varying and lumped uncertainties under such challenging and uncertain conditions, it becomes necessary to investigate the behavior of the estimation error in a more detailed and comprehensive manner. A closer examination allows for a better understanding of how the RBFNN adapts its internal structure to match complex uncertainty patterns, and how quickly and effectively it can recover from sudden perturbations introduced into the system. By analyzing the

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of Electrical and Electronics Research (IJEER)
Research Article | Volume 14, Issue 2 | Pages 271-278 | e-ISSN: 2347-470X

evolution of the estimation error over time, especially at critical time intervals such as after the 10th second—where friction uncertainties occur—and the 20th second—where a reduction in actuator efficiency is introduced—researchers and engineers can validate the estimator's performance and adaptability in scenarios that closely resemble real-world operational challenges. These insights are presented and visualized in *figure 7* above, which provides a clear and informative representation of the estimation error dynamics, thereby offering valuable evidence of the RBFNN of uncertainty rejection capability and its potential applicability in complex, real-time control systems.

Figure 8 shows that RBFNN has nearly zero errors in all free fault processes up to the unexpected fault process. This will also lead to a change in the control signal.

Thanks to the error compensation capability of the proposed controller, the control signal has been changed in time when a fault occurs. It can be easily seen that from the 10th second, the error compensation control signal occurs, and the amplitude increases when another fault is assumed to be activated in the system from the 20th second.

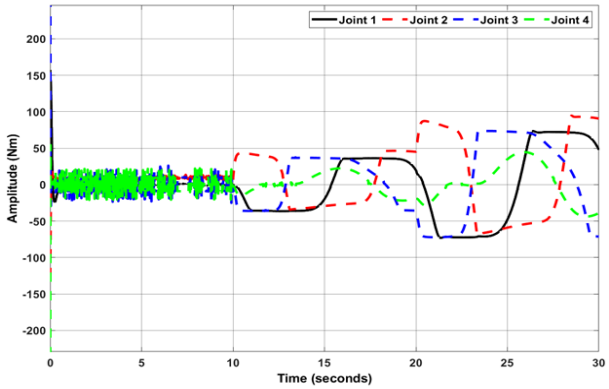


Figure 8. Error estimation results in Joint 1, Joint 2, Joint 3, Joint 4

5. DISCUSSION

This section provides a deeper analysis of these outcomes and highlights the novel contributions of the study.

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\theta_i - \theta_{d_i}\right)^2} \quad (31)$$

The error was calculated based on the RMSE technique using *equation (31)* above. Where θ_i denotes the actual joint angle at the i-th sample, θ_{d_i} the desired joint angle, and N is the total number of samples. *Table 2* below shows the error results between the setpoint signal and the output feedback signal at the four joints of the robot manipulator.

Table 2. Comparison of RMSE performance between SMC and the proposed SMCRBFNN

Joint	SMC	SMCRBFNN
1	5.1409	2.2537
2	5.2411	1.9933
3	5.5501	2.8485
4	3.4810	1.7519

As illustrated in *table 2*, the proposed SMCRBFNN significantly outperforms the traditional SMC in all four joints. The substantial reduction in RMSE confirms the effectiveness of the RBFNN in compensating for lumped uncertainties such as friction and actuator faults, ensuring high-precision trajectory tracking for the 4-DOF manipulator. However, several limitations must be acknowledged. First, the RBFNN cannot completely compensate for all system disturbances, such as complex friction, which necessitates further improvements in estimation accuracy. Second, due to the high sensitivity of the RBFNN, minor tracking errors are observed even in the absence of lumped uncertainties, suggesting a trade-off between adaptability and baseline stability.

6. CONCLUSIONS

In summary, this paper has presented the dynamics of a robotic arm under computed aggregated uncertainties. The lumped uncertainties referred to here are the fault friction, which includes viscous friction, static friction, and the loss of performance in the actuators. Next, the proposed controller based on SMC and RBFNN is presented. While SMC makes the system track the set tracking signal, RBFNN is responsible for estimating the fault output in the system without knowing the dynamic equation of the system, demonstrating its efficiency and simplicity. In addition, the stability of the controller has also been proven according to the Lyapunov theory. Next, the superiority of the proposed controller on the 4-degree-of-freedom robot arm has been presented based on some simulations on MATLAB Simulink 2018b software. The effectiveness of the proposed control strategy is validated through simulation, where the trajectory tracking performance is quantitatively confirmed by low root mean square error (RMSE) values. This study serves as a foundation for the future development and practical implementation of the proposed method on real-world systems.

ACKNOWLEDGMENTS: This work was supported by Ho Chi Minh City University of Technology and Engineering, Vietnam.

REFERENCES

[1] A. Gasparetto and L. Scalera, *A brief history of industrial robotics in the 20th century*. Publisher: Advances in Historical Studies, vol. 8, pp. 24–35, 2019.

[2] Walid Kh. Alqaisi and et al, *Four DOF Robot Manipulator Control Using Feedback Linearization Based on Sliding Mode Control*. Publisher: International Journal of Robotics and Control Systems, vol. 5, no. 2, 2025, pp. 781-793 ISSN 2775-2658.

[3] L. Li, *Fault Types and Diagnostic Methods of Manipulator Robots: A Review*. Publisher: Springer, 2016.

[4] D. T. Tran and T. Q. Nguyen, *Prescribed performance adaptive fault-tolerant control for a manipulator*. Publisher: Proc. 13th Asian Control Conf. (ASCC), 2022.

[5] Adedire D. Adesiji and et al, *Safety Considerations in Deployment of Robotic Systems – A Systematic Review*. Publisher: Journal of Field Robotics, 2026; 43:5–33.

[6] D. T. Tran and T. Q. Nguyen, *Fault-tolerant sliding mode controller for a 4 degree of freedom robotic manipulator*. Publisher: Proc. 13th Asian Control Conf. (ASCC), 2022.

International Journal of
Electrical and Electronics Research (IJEER)
Research Article | Volume 14, Issue 2 | Pages 271-278 | e-ISSN: 2347-470X

[7] H. Kalle and K. Iqbal, *Online Estimation of Manipulator Dynamics for Computed Torque Control of Robotic Systems.* Publisher: Sensors 2025, *25*(22), 6831.

[8] X. Cheng, H. Liu, and W. Lu, *Chattering-suppressed sliding mode control for flexible-joint robot manipulators*. Publisher: Actuators, vol. 10, no. 11, p. 288, 2021.

[9] M. Bi, *Control of robot arm motion using trapezoid fuzzy two-degree-of-freedom PID algorithm*. Publisher: Symmetry, vol. 12, no. 4, p. 665, 2020.

[10] S. Horvath and H. Neuner, *Introduction of a framework for the integration of a kinematic robot arm model in an artificial neural network - extended Kalman filter approach*. Publisher: J. Intell. Robot. Syst., vol. 110, p. 137, 2024.

[11] M.et U. Danis, and Z. Y. Bayraktaroglu, *Extended State Observer-Based Motion Control of Robot Manipulators in Presence of Uncertainties and Disturbances*. Publisher: IET Control Theory & Applications, 2025; 19:e70043.

[12] V. N. Son, P. V. Cuong, N. D. Minh and P. H. Nha, *Optimize the parameters of the PID Controller using Genetic Algorithm for Robot Manipulators.* Publisher: The First National Conference on Energy, Electronics, and Automation - EEA 2024.

[13] A. S. Ahmed and S. K. Kadhim, *A comparative study between convolution and optimal backstepping controller for single arm pneumatic artificial muscles*. Publisher: J. Robotics and Control (JRC), 2022.

[14] Ahmad Mohamad El-fallah Ismail, *Effective Control of a Robotic Manipulator's Trajectory Tracking through the utilization of an Optimal Linear Quadratic Gaussian (LQG) controller enhanced by the Improved Particle Swarm Optimization (IPSO) Algorithm*. Publisher: Gharyan University Journal of Engineering Science (GUJES) 1 (1), 16, 2025.

[15] S. Li and Y. Zhang, *Neural Networks for Cooperative Control of Multiple Robot Arms*. Publisher: Springer, 2018.

[16] D. N. Bui and M. D. Phung, *Radial basis function neural networks for formation control of unmanned aerial vehicles*. Publisher: Robotica, 2024.

[17] A. Hasan, M. Tahavori, and H. S. Midtiby, *Model-based fault diagnosis algorithms for robotic systems*. Publisher: IEEE Access, vol. 11, 2023.

[18] E. Y. Chia, Y. L. Chen, T. C. Chien, M. L. Chiang, L. C. Fu, and J. S. Lai, *Velocity field based active-assistive control for upper limb rehabilitation exoskeleton robot*. Publisher: Proc. IEEE Int. Conf. on Robotics and Automation (ICRA), 2020.

[19] J. Zhang, S. Liu, and L. Xie, *Distributed Convex Optimization Over Nonlinear Networks Under Set Constraints.* Publisher: IEEE Transactions on Automatic Control, vol: 70, issue: 7, 2025.

[20] D. Gamez-Herrera and et al, *Composite Adaptive Control of Robot Manipulators with Friction as Additive Disturbance*. Publisher: Actuators 2025, 14(5), 237.

[21] M. Y. Silaa and et al, *An Adaptive Control Strategy with Switching Gain and Forgetting Factor for a Robotic Arm Manipulator*. Publisher: Machines 2025, 13(5), 424.

[22] M. A. Saaideh and et al, *Robust Output Feedback controller for a Serial Robotic Manipulator with Unknown Nonlinearities and External Disturbances*. Publisher: 2023 IEEE International Conference on Robotics and Automation (ICRA 2023), 2023.

[23] S. Sahoo, H. Moradi, and M. Sababheh, *A mixed Cauchy-Schwarz inequality with applications*. Publisher: Operators and Matrices, 2022.