Journal of Robotics and Control (JRC)
Volume 6, Issue 4, 2025
ISSN: 2715-5072, DOI: 10.18196/jrc.v6i4.26537

Adaptive Task-Space Control of Five-Bar Parallel Robot Dynamic Model with Fully Unknown Using Radial Basis Function Neural Networks for High-Precision Applications

Thanh Hai Tran [1], Thanh Quyen Ngo [2*], Hoang Thi Tu Uyen [3], Van Tho Nguyen [4], Tien Đoan Duong [5]
[1] Office of Planning and Investment, Industrial University of Ho Chi Minh City, Vietnam
[2, 3, 4] Faculty of Electrical Engineering Technology, Industrial University of Ho Chi Minh City, Vietnam
[5] Office of Services, Industrial University of Ho Chi Minh City, Vietnam
Email: [1] tranthanhhai@iuh.edu.vn, [2] ngothanhquyen@iuh.edu.vn, [3] hoangthituuyen@iuh.edu.vn, [4] nguyenvantho@iuh.edu.vn, [5] duongtiendoan@iuh.edu.vn
*Corresponding Author

***Abstract*—Designing a stable and accurate controller for nonlinear systems remains a significant challenge, mainly when the system contains uncertain factors or is affected by external disturbances. This study proposes an adaptive control method based on a Radial Basis Function Neural Network (RBFNN) to effectively estimate the uncertain components in nonlinear systems. The gradient descent algorithm updates the RBFNN parameters, and the control system's stability is rigorously proven based on the Lyapunov theory. The designed controller ensures accuracy under changing conditions and can adapt to nonlinear disturbances and system fluctuations flexibly. Through 45 consecutive test cycles, the system significantly improves precision and outperforms other control methods in comparative tests. This study opens up the potential for broad application in highly uncertain nonlinear MIMO systems, thanks to the effective combination of adaptive learning ability, stability, and simple implementation structure of the proposed controller.**

***Keywords*—Nonlinear Uncertain Systems; MIMO Systems; Lyapunov Stability; Radial Basis Function Neural Network (RBFNN); Adaptive Parameter Tuning.**

I. INTRODUCTION

In robotics, trajectory tracking is one of the most important and long-standing challenges, especially when the system is subject to unforeseen disturbances or the dynamic model is not fully known. Various control methods have been proposed to address this problem, primarily based on Lyapunov stability theory, including adaptive force and vision tracking control [1], adaptive force/position control [2], [3], coordinated control [4], [5], conduction control [6], and adaptive impedance control [7]. However, designing controllers for unknown physical systems to ensure the required accuracy is challenging and complex in many control applications [8], [9]. The main reasons come from the difficulties in constructing accurate dynamic models, the errors in the modeling process, the influence of complex environmental factors, and unpredictable disturbances [10]-[18]. These factors degrade the stability and significantly affect the accuracy of nonlinear systems [11]. In particular, in collaborative applications between robots to perform tasks such as polishing, assembly, or monitoring [12], the design of a consensus tracking controller is extremely necessary to ensure operational efficiency and accuracy, even in the face of uncertainties and disturbances [13]-[14]. Therefore, in cases where system parameters and external environmental influences are not well defined or continuously varying, linear controllers often become difficult to apply in practice [15]-[18].

Control methods such as PID [19] and fuzzy control systems [20]-[22] are often favored because they do not require complex mathematical models, but instead rely on parameter tuning from system feedback to optimize accuracy. However, their major drawback lies in the difficulty of optimizing control parameters. Therefore, various control methods have emerged, including adaptive control [23]-[25], robust control [25], sliding mode control [26]-[27], disturbance rejection control [28], and disturbance observation-based methods [29], [30]. Among them, sliding mode control [31]-[34] derived from the theory of Variable Structure Control Systems (VSCS) stands out due to its outstanding advantages: resistance to parameter uncertainty, low sensitivity to external disturbances, and high response speed [35]-[38]. However, SMC still has limitations when requiring precise understanding of the mathematical model and physical parameters of the system, leading to difficulties in practical application and reduced efficiency in some cases [38]-[39].

Although current methods offer some advantages in dealing with problems such as parameter uncertainty, incomplete dynamic models, and disturbances, limitations still affect the system's accuracy. These limitations include high-gain feedback [40], bandwidth constraints [41], and chattering phenomena [42]-[43], all of which affect the controllability, reachability, and adaptability of the system [44]-[46]. When the system faces significant disturbances, achieving complete stability is no longer practical from an engineering perspective [47]. Since the target state of the

system may be mathematically uncertain and the actual state of the system may fluctuate around this target, the concept of asymptotic stability is considered practically appropriate and acceptable in engineering applications [47]-[48]. Studies on disturbance estimation have been carried out for uncertain nonlinear systems in [49]-[50] for specific uncertain nonlinear systems. The development of robust adaptive controllers that do not require precise information about uncertainty bounds has been addressed in [50]. In this study, the authors have combined adaptive control technology and sliding mode control theory. In addition, other control methods, such as neural networks and fuzzy control, have attracted considerable attention due to their ability to approximate functions and flexibility in tuning the system [51]-[53].

A disturbance-aware control strategy has also been proposed for systems with multiple controllers and saturated inputs using Radial Basis Function Neural Networks (RBFNN) [54]. The global learning capability of neural networks makes them helpful in capturing uncertainties in robot control. In particular, controllers that combine neural networks with fast online learning and convergence capabilities, such as RBFNN, are becoming a prominent trend in robot control [55]-[59].

This paper proposes an intelligent controller for MIMO systems, which is capable of effectively handling uncertain constraints and disturbances from the environment. The highlight of the method is the use of Radial Basis Function Neural Network (RBFNN) to accurately estimate the dynamic model of the control system. Thanks to the powerful approximation and online learning capabilities of RBFNN, the controller can enhance its adaptiveness to unpredictable fluctuations and disturbances. The effectiveness and feasibility of the controller have been verified through practical experiments, confirming the potential application in control systems.

The main contributions of this work can be summarized as follows:

1. The parameters of the RBFNN controller are optimized using the gradient descent method. Specifically, stability is guaranteed through a Lyapunov-based approach.
2. A prominent strength of the proposed method is that it does not require linear conditions for the robot parameters, which expands its applicability to complex nonlinear systems. Therefore, this method can be applied to many types of uncertain MIMO systems due to the adaptability of the controller.
3. The accuracy and reliability of the proposed controller were verified through testing on an actual model. Experimental results demonstrate the stable operation of the controller in managing the robot system.

The paper is organized as follows: Section 2 discusses the issues in uncertain systems. Next, Section 3 introduces the theory of RBFNN and proves its stability using Lyapunov. Section 4 introduces the robust controller and demonstrates the stability of the added component using Lyapunov stability analysis. The experimental process of the proposed controller is presented in Section 5. Finally, Section 6 summarizes and draws conclusions based on the presented information.

II. SYSTEM DESCRIPTION

The state vector equation of the five-bar parallel robot system is found in [66] and defined as follows:

$$\ddot{x} = f(\underline{x}) + g(\underline{x})u \tag{1}$$

where $x(t) \in \Re^m$ represent the system's output, m indicates the number of input or output dimensions. The control input vector $u = [u_1, u_2, \cdots, u_m]^T \in \Re^m$, $\underline{x} = \left[x^T, \dot{x}^T, \cdots, x^{(n-1)T}\right]^T \in \Re^{m \times n}$ encompasses the system's states, including the output and its derivatives up to the $(n-1)th$ order. $f(\underline{x})$ is the nominal function representing the system's nonlinear dynamics and $g(\underline{x})$ is the nominal constant gain matrix that influences the control input u. Eq. (1) can be expressed as state as follows:

$$\begin{aligned} \ddot{x} &= [f_n(\underline{x}) + \Delta f(\underline{x})] + [g_n(\underline{x}) + \Delta g(\underline{x})]u \\ &= f_n(\underline{x}) + g_n(\underline{x})u + d \end{aligned} \tag{2}$$

The system dynamics are represented by standard functions $f_n(\underline{x}) \in \Re^m$ and $g_n(\underline{x}) \in \Re^{m \times m}$, which are assumed to be bounded and can be determined. The terms $\Delta f(\underline{x})$ and $\Delta g(\underline{x})$ are the respective uncertain terms in $f(\underline{x})$ and $g(\underline{x})$. $d = \Delta f(\underline{x}) + \Delta g(\underline{x})u$ is the sum of the uncertain components inside and outside the system.

The primary goal of the overall control strategy is to ensure that the system's trajectory x aligns with a desired reference trajectory $x_d \in \Re^m$. To achieve this, the tracking error $e \in \Re^m$ is defined as: $e = x_d - x$. Therefore, considering the state vector x, the tracking error vector $\underline{e}$ for the system is defined as follows:

$$\underline{e} = \left[e^T, \dot{e}^T \cdots, e^{(n-1)T}\right]^T \tag{3}$$

The first step is to select a sliding surface to implement. Then, design a control law so that the system state trajectories converge and remain on the sliding surface, which can be defined as follows:

$$s = e^{(n-1)} + \kappa_1 e^{(n-2)} + \cdots + \kappa_{n-1} e \tag{4}$$

where $s = [s_1, s_2, \cdots, s_m]^T$, $\kappa_i = diag\left(\kappa_{i1}, \kappa_{i2}, \cdots, \kappa_{ij}\right)$ for $i = 1,2, \cdots, n$; with each element in κ_{ij} being a positive constant satisfying the Hurwitz characteristic polynomial.

Assuming that the components $f_n(\underline{x})$, $g_n(\underline{x})$ and the sum of the uncertain components d have been determined, the ideal controller can be designed as follows:

$$\begin{aligned} u = g_n^{-1}(\underline{x})\big[\ddot{x}_d - f_n(\underline{x}) - d + \kappa_i e^{(n-1)} \\ + \sigma sgn(s)\big] \end{aligned} \tag{5}$$

Differentiating s for time leads to:

$$\dot{s} = e^{(n)} + \kappa_i e^{(n-1)} \tag{6}$$

Consider the candidate Lyapunov function in the following form:

$$V_1(s) = \frac{1}{2} s^2 \tag{7}$$

Differentiating Eq. (7) and using Eq. (6):

$$\begin{aligned}\dot{V}_1(s) &= s\dot{s} = s\left[\ddot{x}_d - \ddot{x} + \kappa_i e^{(n-1)}\right] \\ &= s\left[\ddot{x}_d - f_n(\underline{x}) - g_n(\underline{x})u - d + \kappa_i e^{(n-1)}\right] \\ &= s\Big[\ddot{x}_d - f_n(\underline{x}) - \left[x_d^{(n)} - f_n(\underline{x}) - d + \kappa_i e^{(n-1)} \right. \\ &\qquad \left. + \sigma sgn(s)\right] - d + \kappa_i e^{(n-1)}\Big] \\ &= s[-\sigma sgn(s)] \le -\sigma|s| \le 0\end{aligned} \tag{8}$$

The sliding-mode controller ensures stability according to the Lyapunov theorem [60]. However, a high control gain σ typically leads to significant chattering effects. Besides, the uncertainty components d are generally unknown in practical scenarios. Therefore, the ideal control input u in Eq. (5) is unavailable. Therefore, an RBFNN control system, which consists of a RBFNN combined with a robust controller, is proposed to address trajectory tracking:

$$u_{controller} = u_{RBFNN} + u_{RC} \tag{9}$$

In this control structure, the main controller, denoted as u_{RBFNN}, is designed to approximate the components $f_n(\underline{x})$ and $g_n(\underline{x})$ in the ideal controller in Eq. (5). u_{RC} is a robust controller used to compensate the approximation error between u_{RBFNN} and u.

III. CONTROLLERS DESIGN

A. Definition of RBFNN Controller

The Radial Basis Function Neural Network (RBFNN), introduced by J. Moody and C. Darken in the late 1980s, is a neural network model with a three-layer structure: input layer, hidden layer, and output layer [61]-[63]. The RBFNN is characterized by its ability to accurately approach and approximate any continuous function, thanks to its operating mechanism, which resembles that of human neurons. RBFNN differs from backpropagation (BP) neural networks by its strong local approximation ability and significantly faster training time [64]-[65].

In terms of structure, each layer of the RBFNN contains many nodes, equivalent to neurons, and these nodes are fully connected between adjacent layers. The input data is transmitted linearly from the input layer to the hidden layer, where it undergoes a nonlinear transformation before being further transmitted and transformed linearly to the output layer. RBFNN often uses the Gaussian function as the radial basis function, allowing it to process input data efficiently and accurately.

$$h_{ij} = exp\left[\frac{-\left(x_i - c_j\right)^2}{2b_j^2}\right], i = f, g \tag{10}$$

where c_j is the center vector of the j-th node of the network and b_j is the base width parameters of the node j= 1,2, ⋯ , m, and it is a number greater than zero. The weight vectors of network nodes in hidden layer:

$$W = \left[w_{11} \cdots w_{1k} \cdots w_{j1} \cdots w_{jk}\right]$$

The output of network is described by

$$\begin{aligned} f^* &= W^{*T} h_f + \varepsilon_f \\ g^* &= V^{*T} h_g + \varepsilon_g \end{aligned} \tag{11}$$

where W^* and V^* are the ideal weights of the network structure, ε_f and ε_g are the approximation errors. The output of the RBFNN is defined:

$$\begin{aligned} \hat{f}(\underline{x}) &= \widehat{W}^T h_f + \varepsilon_f \\ \hat{g}(\underline{x}) &= \hat{V}^T h_g + \varepsilon_g \end{aligned} \tag{12}$$

where h_f and h_g are the Gaussian functions of the RBFNN.

The ideal controller u in Eq. (5) is rewritten as follows:

$$\begin{aligned} u_{RBFNN} = \hat{g}^{-1}(\underline{x})\big[&\ddot{x}_d - \hat{f}(\underline{x}) - d + \kappa_i e^{(n-1)} \\ &+ \sigma sgn(s)\big] \end{aligned} \tag{13}$$

B. Adaptive Law of Parameters in Weight Layer and Gauss Function

Eq. (6) is rewritten by substituting Eq. (9) into Eq. (1) and using Eq. (5):

$$\begin{aligned} \dot{s} &= e^{(n)} + \kappa_i e^{(n-1)} = \ddot{x}_d - \ddot{x} + \kappa_i e^{(n-1)} \\ &= g_n(\underline{x})(u - u_{RBFNN} - u_{RC}) - \sigma sgn(s) \end{aligned} \tag{14}$$

The gradient descent algorithm is the origin of the parameter learning algorithm, and its goal is to minimize $s\dot{s}$ so that s converges fast. In this method, the adaptive parameters are designed with a law as follows:

$$\begin{aligned} \dot{\widehat{W}} &= -\gamma_1 \frac{\partial s\dot{s}}{\partial \widehat{W}} = -\gamma_1 \frac{\partial s\dot{s}}{\partial u_{RBFNN}} \frac{\partial u_{RBFNN}}{\partial \widehat{W}} \\ &= \gamma_1 s h_f \end{aligned} \tag{15}$$

$$\begin{aligned} \dot{\hat{V}} &= -\gamma_2 \frac{\partial s\dot{s}}{\partial \hat{V}} = -\gamma_2 \frac{\partial s\dot{s}}{\partial u_{RBFNN}} \frac{\partial u_{RBFNN}}{\partial \hat{V}} \\ &= \gamma_2 s h_g u \end{aligned} \tag{16}$$

$$\begin{aligned} \dot{\hat{c}} &= -\gamma_3 \frac{\partial s\dot{s}}{\partial c} = -\gamma_3 \frac{\partial s\dot{s}}{\partial \Xi(\underline{x})} \frac{\partial \Xi(\underline{x})}{\partial h_f} \frac{\partial h_f}{\partial \mathrm{c}} \\ &= \gamma_3 s h_i X \frac{x_i - c}{b^2} \end{aligned} \tag{17}$$

$$\begin{aligned} \dot{\hat{b}} &= -\gamma_4 \frac{\partial s\dot{s}}{\partial b} = -\gamma_4 \frac{\partial s\dot{s}}{\partial \Xi(\underline{x})} \frac{\partial \Xi(\underline{x})}{\partial h_f} \frac{\partial h_f}{\partial b} \\ &= \gamma_4 s h_i X \frac{(x_i - c)^2}{b^3} \end{aligned} \tag{18}$$

where $\Xi(\underline{x}) = \hat{f}(\underline{x}), \hat{g}(\underline{x})$ corresponds to $X = W, V$ respectively and $\gamma_1 > 0, \gamma_2 > 0, \gamma_3 > 0, \gamma_4 > 0$.

Proof: The Lyapunov function is chosen as follows:

$$\begin{aligned} L = \frac{1}{2}s^2 + \frac{1}{2\gamma_1}\widetilde{W}^T\widetilde{W} + \frac{1}{2\gamma_2}\tilde{V}^T\tilde{V} + \frac{1}{2\gamma_3}\tilde{c}^T\tilde{c} \\ + \frac{1}{2\gamma_4}\tilde{b}^T\tilde{b} \end{aligned} \tag{19}$$

where $\tilde{c} = c^* - \hat{c}$; $\tilde{b} = b^* - \hat{b}$. Derivative Eq. (19):

$$\begin{aligned} \dot{L} &= s\dot{s} + \frac{1}{\gamma_1}\widetilde{W}^T\dot{\widetilde{W}} + \frac{1}{\gamma_2}\tilde{V}^T\dot{\tilde{V}} + \frac{1}{\gamma_3}\tilde{c}^T\dot{\tilde{c}} + \frac{1}{\gamma_4}\tilde{b}^T\dot{\tilde{b}} \\ &= s\left[e^{(n)} + \kappa_i e^{(n-1)}\right] + \frac{1}{\gamma_1}\widetilde{W}^T\dot{\widetilde{W}} + \frac{1}{\gamma_2}\tilde{V}^T\dot{\tilde{V}} \end{aligned} \tag{20}$$

$$+\frac{1}{\gamma_3}\tilde{c}^T\dot{\tilde{c}}+\frac{1}{\gamma_4}\tilde{b}^T\dot{\tilde{b}}$$
$$=s\left[\ddot{x}_d-\ddot{x}+\kappa_i e^{(n-1)}\right]+\frac{1}{\gamma_1}\widetilde{W}^T\dot{\widetilde{W}}+\frac{1}{\gamma_2}\tilde{V}^T\dot{\tilde{V}}+\frac{1}{\gamma_3}\tilde{c}^T\dot{\tilde{c}}+\frac{1}{\gamma_4}\tilde{b}^T\dot{\tilde{b}}$$
$$=s\left[\ddot{x}_d-f_n(\underline{x})-g_n(\underline{x})u-d+\kappa_i e^{(n-1)}+\hat{g}(\underline{x})u-\hat{g}(\underline{x})u\right]+\frac{1}{\gamma_1}\widetilde{W}^T\dot{\widetilde{W}}+\frac{1}{\gamma_2}\tilde{V}^T\dot{\tilde{V}}+\frac{1}{\gamma_3}\tilde{c}^T\dot{\tilde{c}}+\frac{1}{\gamma_4}\tilde{b}^T\dot{\tilde{b}}$$
$$=s\left[\ddot{x}_d-f_n(\underline{x})-d+\kappa_i e^{(n-1)}+\hat{g}(\underline{x})u-g_n(\underline{x})u-\ddot{x}_d+\hat{f}(\underline{x})+d-\kappa_i e^{(n-1)}-\sigma sgn(s)\right]+\frac{1}{\gamma_1}\widetilde{W}^T\dot{\widetilde{W}}+\frac{1}{\gamma_2}\tilde{V}^T\dot{\tilde{V}}+\frac{1}{\gamma_3}\tilde{c}^T\dot{\tilde{c}}+\frac{1}{\gamma_4}\tilde{b}^T\dot{\tilde{b}}$$
$$=s\left[\hat{f}(\underline{x})-f_n(\underline{x})-\sigma sgn(s)+[\hat{g}(\underline{x})-g_n(\underline{x})]u\right]+\frac{1}{\gamma_1}\widetilde{W}^T\dot{\widetilde{W}}+\frac{1}{\gamma_2}\tilde{V}^T\dot{\tilde{V}}+\frac{1}{\gamma_3}\tilde{c}^T\dot{\tilde{c}}+\frac{1}{\gamma_4}\tilde{b}^T\dot{\tilde{b}}$$

Lets: $\tilde{f}(\underline{x})=\hat{f}(\underline{x})-f_n(\underline{x})$, $\tilde{g}(\underline{x})=\hat{g}(\underline{x})-g_n(\underline{x})$

$$\tilde{f}(\underline{x})=\widehat{W}^T h_f-W^{*T}h_f-\varepsilon_f=\widetilde{W}^T h_f-\varepsilon_f \quad (21)$$

$$\tilde{g}(\underline{x})=\hat{V}^T h_g-V^{*T}h_g-\varepsilon_g=\tilde{V}^T h_g-\varepsilon_g \quad (22)$$

where $\widetilde{W}=W^*-\widehat{W}$ and $\tilde{V}=V^*-\hat{V}$.

Eq. (20) is rewritten using Eq. (21) and Eq. (22) as follows:

$$\begin{aligned}\dot{L}&=s\left[\widetilde{W}^T h_f-\varepsilon_f-\sigma sgn(s)+\left(\tilde{V}^T h_g-\varepsilon_g\right)u\right]+\frac{1}{\gamma_1}\widetilde{W}^T\dot{\widetilde{W}}+\frac{1}{\gamma_2}\tilde{V}^T\dot{\tilde{V}}+\frac{1}{\gamma_3}\tilde{c}^T\dot{\tilde{c}}+\frac{1}{\gamma_4}\tilde{b}^T\dot{\tilde{b}}\\&=\widetilde{W}^T\left(sh_f+\frac{1}{\gamma_1}\dot{\widehat{W}}\right)+\tilde{V}^T\left(sh_g u+\frac{1}{\gamma_2}\dot{\hat{V}}\right)+s\left[-\varepsilon_f-\sigma sgn(s)-\varepsilon_g u\right]+\tfrac{1}{\gamma_3}\tilde{c}^T\left[\dot{\tilde{c}}+\hat{c}-\hat{c}\right]+\tfrac{1}{\gamma_4}\tilde{b}^T\left[\dot{\tilde{b}}+\hat{b}-\hat{b}\right]\\&=\widetilde{W}^T\left(sh_f+\tfrac{1}{\gamma_1}\dot{\widehat{W}}\right)+\tilde{V}^T\left(sh_g u+\tfrac{1}{\gamma_2}\dot{\hat{V}}\right)+s\left[-\varepsilon_f-\sigma sgn(s)-\varepsilon_g u\right]+\tfrac{1}{\gamma_3}\tilde{c}^T\left[\dot{\tilde{c}}+\dot{\hat{c}}\right]+\tfrac{1}{\gamma_4}\tilde{b}^T\left[\dot{\tilde{b}}+\dot{\hat{b}}\right]\end{aligned} \quad (23)$$

Let $\dot{\hat{c}}=-\dot{\tilde{c}}$ and $\dot{\hat{b}}=-\dot{\tilde{b}}$, Eq. (23) is rewritten:

$$\dot{L}=\widetilde{W}^T\left(sh_f+\frac{1}{\gamma_1}\dot{\widehat{W}}\right)+\tilde{V}^T\left(sh_g u+\frac{1}{\gamma_2}\dot{\hat{V}}\right)+s\left[-\varepsilon_f-\sigma sgn(s)-\varepsilon_g u\right]+\frac{1}{\gamma_3}\tilde{c}^T\left[\dot{\tilde{c}}-\dot{\tilde{c}}\right]+\frac{1}{\gamma_4}\tilde{b}^T\left[\dot{\tilde{b}}-\dot{\tilde{b}}\right] \quad (24)$$

Substitute the adaptive parameters from Eq. (15) – Eq. (18) into Eq. (24):

$$\dot{L}=\widetilde{W}^T\left(s\mathbb{Z}_f-\frac{1}{\gamma_1}\gamma_1 s\mathbb{Z}_f\right)+\tilde{V}^T\left(sh_g u-\frac{1}{\gamma_2}\gamma_2 sh_g u\right)+s\left[-\varepsilon_f-\sigma sgn(s)-\varepsilon_g u\right]+\frac{1}{\gamma_3}\tilde{c}^T\left[-\gamma_3 sh_i X\frac{x_i-c}{b^2}+\gamma_3 sh_i X\frac{x_i-c}{b^2}\right]+\frac{1}{\gamma_4}\tilde{b}^T\left[-\gamma_4 sh_i X\frac{(x_i-c)^2}{b^3}+\gamma_4 sh_i X\frac{(x_i-c)^2}{b^3}\right]=s\left[-\varepsilon_f-\sigma sgn(s)-\varepsilon_g u\right]\le-\left(\varepsilon_f+\varepsilon_g u\right)-\eta|s| \quad (25)$$

If the approximation errors ε_f and ε_g are bounded with $\sigma\ge\left|\varepsilon_f+\varepsilon_g\tau\right|$ then $\dot{L}\le 0$.

IV. Analyze Control Structures

In actual practical applications, due to the limited capability of the RBFNN, there will exist an approximation error between the ideal controller ($\boldsymbol{u}_{IDEAL}$) and the estimated controller from RBFNN ($\boldsymbol{u}_{RBFNN}$), which is defined:

$$\varepsilon=u-u_{RBFNN} \quad (26)$$

Here, ε represents the approximation error. By substituting Eq. (26) into Eq. (14):

$$\dot{s}=g_n(\underline{x})(\varepsilon-u_{RC})-\sigma sgn(s) \quad (27)$$

The approximation error is often difficult to measure in practical applications. The study employed a robust controller u_{RC} to enhance adaptability and respond to uncertainties:

$$u_{RC}=\hat{\varepsilon}+(2R^2)^{-1}\left[[I+(W-V)^2]R^2+I\right]s \quad (28)$$

where $\hat{\varepsilon}(t)$ denotes the estimated value of approximation error and $R=diag[\zeta_1,\zeta_2]$ is the learning rate of the robust controller for the system to converge. Substituting (28) into (27) yields:

$$\begin{aligned}\dot{s}&=g_n(\underline{x})(\varepsilon-\hat{\varepsilon}-(2R^2)^{-1}[(I+(W-V)^2)R^2+I]s)\\&=g_n(\underline{x})(\tilde{\varepsilon}-(2R^2)^{-1}[(I+(W-V)^2)R^2+I]s)\end{aligned} \quad (29)$$

where $\tilde{\varepsilon}=\varepsilon-\hat{\varepsilon}$. The Lyapunov function chosen to ensure the stability of the RBFNN system is defined as follows:

$$V_2(s,\tilde{\varepsilon})=\frac{1}{2}s^2+\frac{1}{2\mu_\varepsilon}\tilde{\varepsilon}^2 \quad (30)$$

where μ_ε is a positive learning rate. Let's perform the time derivative analysis of Eq. (30). Then, use Eq. (29) to complete:

$$\begin{aligned}\dot{V}_2(s,\tilde{\varepsilon})&=s\dot{s}+\frac{1}{\mu_\varepsilon}\tilde{\varepsilon}^T\dot{\tilde{\varepsilon}}\\&=sg_n(\underline{x})(\tilde{\varepsilon}-(2R^2)^{-1}[(I+(W-V)^2)R^2+I]s)+\frac{1}{\mu_\varepsilon}\tilde{\varepsilon}^T\dot{\tilde{\varepsilon}}-\eta|s|\\&=sg_n(\underline{x})\tilde{\varepsilon}+\frac{1}{\mu_\varepsilon}\tilde{\varepsilon}^T\dot{\tilde{\varepsilon}}-\eta|s|-s^2g_n(\underline{x})(2R^2)^{-1}[(I+(W-V)^2)R^2+I]\\&=-s^2g_n(\underline{x})(2R^2)^{-1}[(I+(W-V)^2)R^2+I]\tilde{\varepsilon}\left[sg_n(\underline{x})+\frac{1}{\mu_\varepsilon}\dot{\tilde{\varepsilon}}\right]-\eta|s|\end{aligned} \quad (31)$$

For achieving $\dot{V}_2(s,\tilde{\varepsilon}) \leq 0$, the error estimation law is designed as:

$$\dot{\hat{\varepsilon}} = -\dot{\tilde{\varepsilon}} = \mu_\varepsilon s g_n(\underline{x}) \tag{32}$$

Then Eq. (31) can be rewritten as:

$$\dot{V}_2(s,\tilde{\varepsilon}) = -(2R^2)^{-1}[(I+(W-V)^2)R^2 + I]s^2 - \eta|s| \leq 0 \tag{33}$$

Since $\dot{V}_2(s,\tilde{\varepsilon})$ is negative semidefinite, that is $\dot{V}_2(s,\tilde{\varepsilon}) \leq \dot{V}_2\big(s(0),\tilde{\varepsilon}(0)\big)$, it implies that s and $\tilde{\varepsilon}$ are bounded. Let function $\Upsilon \equiv -(2R^2)^{-1}[(I+(W-V)^2)R^2 + I]s^2 - \eta|s|$, and integrate Υ for time, then it is obtained that:

$$\int_0^t \Upsilon(\tau)d\tau \leq V_2(0) - V_2(t) \tag{34}$$

Given that $V_2(t)$ is non-increasing and bounded, and $V_2(0)$ is bounded, the result is expressed as follows:

$$\lim_{t\to\infty} \int_0^t \Upsilon(\tau)d\tau < \infty \tag{35}$$

Moreover, since $\dot{\Upsilon}(t)$ is bounded, by Barbalat's Lemma, $\lim_{t\to\infty} \int_0^t \Upsilon(t) = 0$ [60]. That is, $s \to 0$ as $t \to \infty$. The proposed RBFNN system's stability can be ensured as a result.

Fig. 1 depicts the proposed network architecture for nonlinear control and processing of the system. The input layers of the RBFNN, including state variables x_1, x_2, and x_3, are passed through hidden layers to calculate the output layers $\hat{f}(\underline{x})$ and $\hat{g}(\underline{x})$Experiment Results

The proposed method was validated using a five-bar parallel robot (Fig. 2), chosen for its mechanical complexity to rigorously test the theoretical framework. Operational uncertainties emerge from normal interactions, preventing exact determination, while the control process is further complicated by varying viscosity, friction coefficients, etc. The system's dynamics, modeled via the Lagrange method, are expressed in Eq. (36).

$$M(q')\ddot{q} + C(q',\dot{q}')\dot{q} + B_m\dot{q} + g(q') = u \tag{36}$$

where $q' = (q_1, q_2)^T$ represents the robot's general coordinates. $q', \dot{q}, \ddot{q} \in R^{2\times1}$ are the position, velocity, and joint acceleration vectors; $M(q') \in R^{2\times2}$ is the moment of inertia matrix; $C(q',\dot{q}') \in R^{2\times2}$ is the Coriolis and Centrifugal matrix; $B_m = (b_{m1}, b_{m2})$ represents the viscosity of the engine; $g(q') \in R^{2\times1}$ is the gravity vector; u is the control variable.

The moment of inertia matrix is redefined:

$$M(q') = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} \tag{37}$$

$m_{11} = 2J_1 + J_3 + m_3(l_1^2 + \tau_3^2 + 2cos(\delta_3 + q_3)l_1\tau_3) + m_1\tau_1^2$
$m_{12} = m_{21} = 0$
$m_{22} = 2J_2 + J_4 + m_4(\tau_2^2 + \tau_4^2 + l_2cos(\delta_4 + q_4)\tau_4) + m_2\tau_2^2$
Centripetal force matrix-Coriolis:

$$C(q',\dot{q}') = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} \tag{38}$$

$c_{11} = -l_1m_3\dot{q}_3\tau_3\sin(\delta_3 + q_3)$
$c_{12} = c_{21} = 0$
$c_{22} = -l_2m_4\dot{q}_4\tau_4\sin(\delta_4 + q_4)$
Gravity matrix:

$$g(q') = \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} \tag{39}$$

$g_1 = 9.81cos(\delta_1 + q_1)(l_1m_3 + m_1\tau_1) + 9.81m_3\tau_3cos(\delta_3 + q_1 + q_3)$
$g_2 = 9.81cos(\delta_2 + q_2)(l_2m_4 + m_2\tau_2) + 9.81m_4\tau_4cos(\delta_4 + q_2 + q_4)$

The proposed controller is validated by accurately describing the system dynamics, where the equations are detailed in Eqs (37)-(39), and the parameters are listed in Table I.

TABLE I. MODEL PARAMETERS WHEN SIMULATION

Symbol	Parameters
L_1, L_2, L_3, L_4, L_5	0.127m
m_1, m_2, m_3, m_4	0.065kg
B_m	[1,1]
$\delta_1, \delta_2, \delta_3, \delta_4$	1
d	1

In order to improve the accuracy of the five-bar parallel robot system, this study proposes a control structure to ensure the operating flexible adaptability to different conditions and environments. The experiment and data collection were conducted by integrating the NI PCIe-6351 board with the computer and using Simulink software on the MATLAB platform to control the robot arm. Fig. 3 illustrates the structure of the robot system.

This section presents the experimental results of the five-bar parallel robot system under disturbance-free conditions. The experiment aims to evaluate the feasibility of the proposed robot controller in accurately tracking the desired trajectory. In Section C, the results are compared with other control methods to demonstrate the advantages of the proposed method in handling the robot system

A. Stability Evaluation of the Proposed Controller

The authors validate the theory in this section by running the five-bar parallel robot system for 45 consecutive cycles. After each cycle, errors are calculated to evaluate the convergence of the proposed theory. Fig. 4 depicts the control error over cycles for two angles θ_A and θ_B. The above graph uses the red dashed line to depict the control error of θ_A with irregular oscillation and a decreasing trend. Meanwhile, the chart below uses the blue dashed line to represent the control error of θ_B, which tends to decrease rapidly with each cycle. The decreasing control error of the system shows that the controller is gradually stabilizing and the accuracy of the system is improving over time.

Fig. 5 shows the variation of the error over time during the tracking error. For θ_A, the tracking error starts with a high value, then gradually decreases but still fluctuates and shows an improving trend over time. Meanwhile, the tracking error of θ_B fluctuates sharply and has a decreasing trend. The system accuracy is improving over time, and the tracking error is decreasing, indicating that the controller is stabilizing.

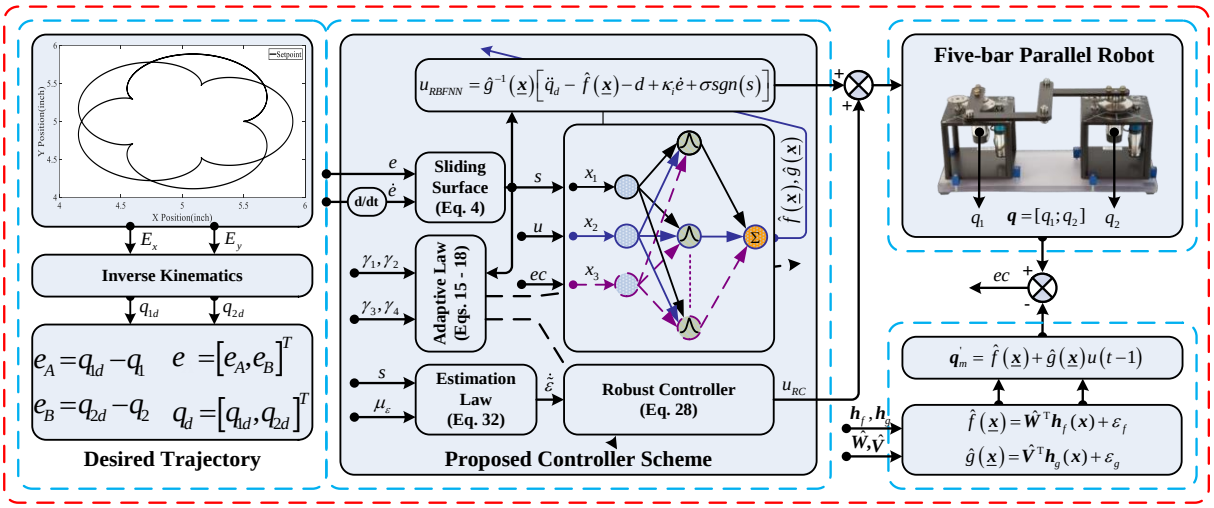


Fig. 1. Diagram depicting the RBFNN control system

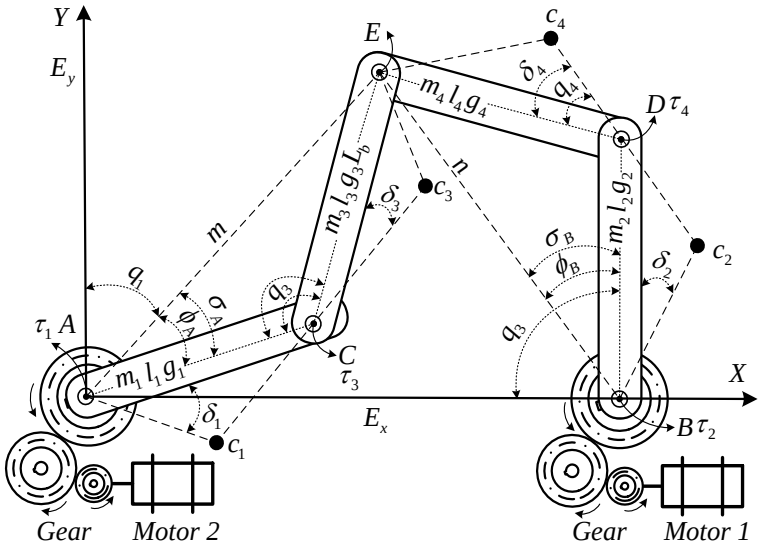


Fig. 2. Structure diagram of five-bar parallel robot

Fig. 3. Experimental system

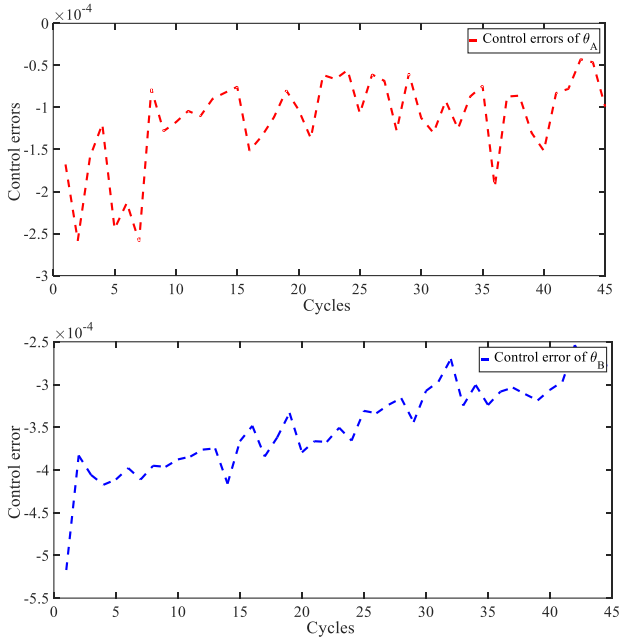


Fig. 4. Control error over cycles

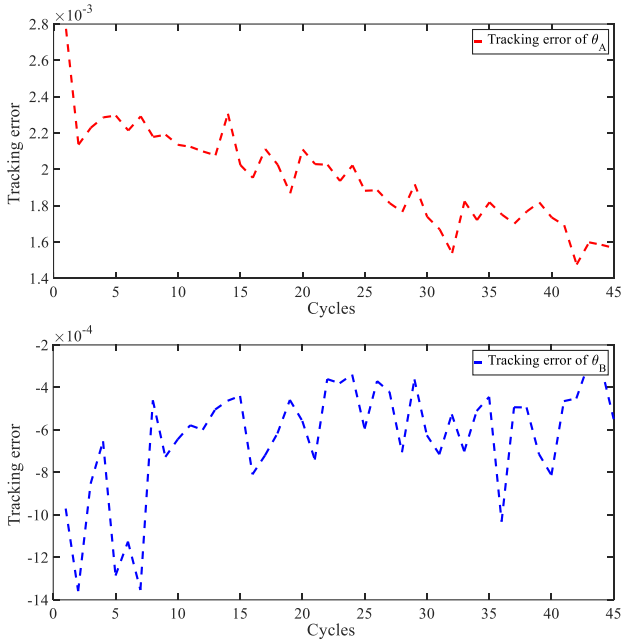


Fig. 5. Tracking error over cycles

Fig. 6 shows the mean square error (MSE) variation over the cycles for two angles, θ_A and θ_B. For angle θ_A, the MSE fluctuates strongly and decreases unsteadily throughout the cycles. In contrast, angle θ_B starts with a high MSE but decreases quickly and stabilizes after a few cycles, indicating that angle θ_B converges faster.

The results demonstrate that the controller is gradually stabilizing, with the system accuracy improving as the parameters decrease over time. However, this error reduction phenomenon must be carefully analyzed to determine the cause. First, it could be a sign of convergence to the steady state, reflecting the effectiveness of the control algorithm. Second, with adaptive controllers, the error reduction is an inevitable consequence of the parameter auto-tuning mechanism.

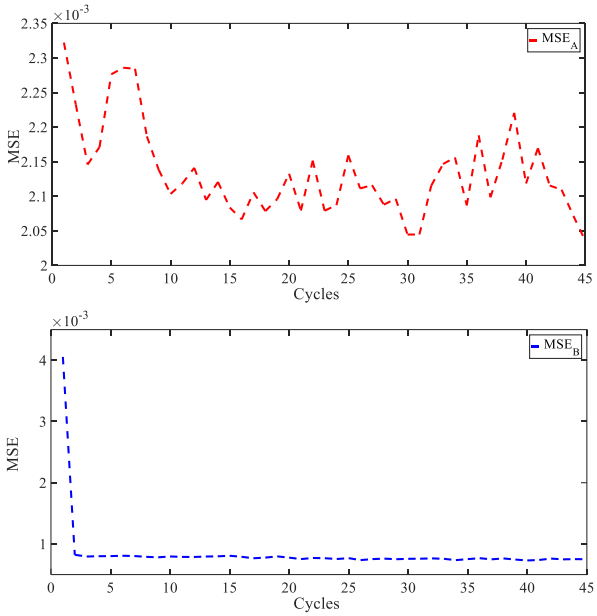


Fig. 6. Mean squared error (MSE) over cycles

B. *Results in One Cycle*

Fig. 7 illustrates the control angles (θ_A and θ_B) over a period of 0 to 15 seconds, with the red line representing the system response when using RBFNN. The two zoomed-in images at each angle emphasize the system's outstanding accuracy. The zoomed-in details show that the RBFNN controller accurately tracks the desired trajectory, even in complex and continuous curves.

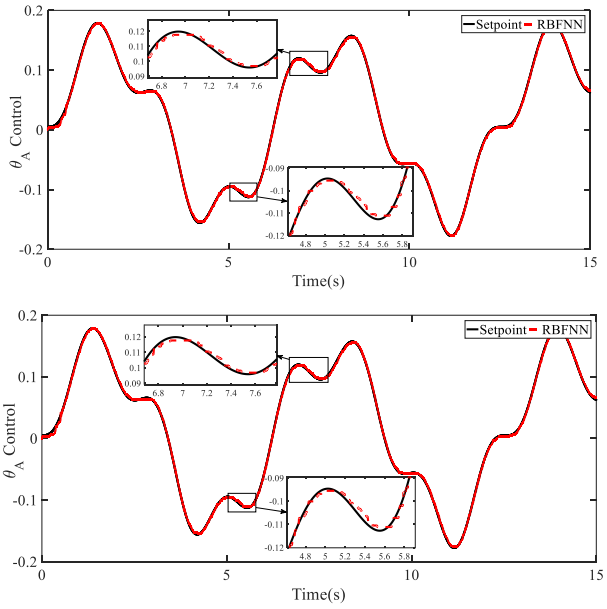


Fig. 7. Control angle θ_A and θ_B

Fig. 8 illustrates the tracking capability of the θ_A and θ_B angles of the control system, with the black lines representing the reference values (Setpoint) and the red lines representing the system response using RBFNN. The zoom-in sections in the figure reveal that the control system exhibits a tracking response with minor deviations between the actual and reference values. These zoom-ins highlight the system's ability to track and handle details in consecutive curves.

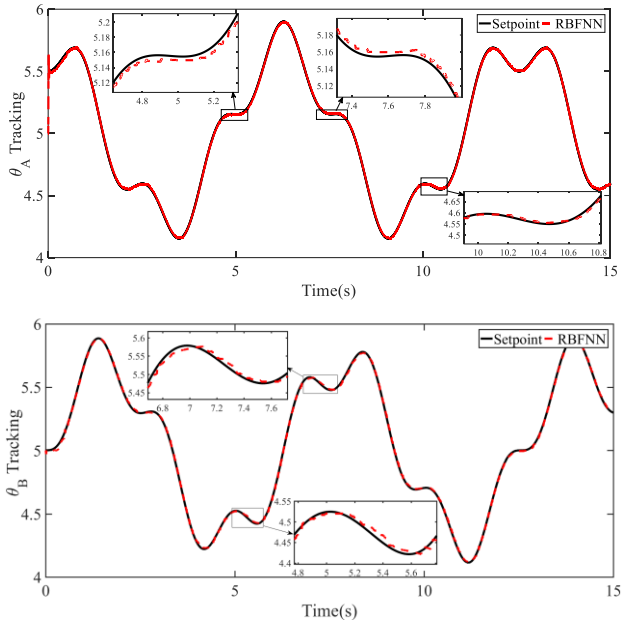


Fig. 8. Tracking angle θ_A and θ_B

Fig. 9 and 10 show four graphs depicting the control error (Fig. 9) and tracking error (Fig. 10) of two parameters θ_A and θ_B in a control system using RBFNN. Fig. 9 shows the control error θ_A, which fluctuates around zero with a large amplitude at the beginning but tends to decrease and stabilize at a certain value range. Fig. 10 shows that the tracking error θ_A also fluctuates but with a smaller amplitude, stabilizing the tracking error over time. Similarly, the two lower graphs of Fig. 9 and Fig. 10 show the control error and tracking error of parameter θ_B. The control error θ_B fluctuates more strongly but is still concentrated around zero, while the tracking error θ_B tends to decrease in amplitude more clearly over time. These results demonstrate the effectiveness of RBFNN in reducing errors and increasing control accuracy.

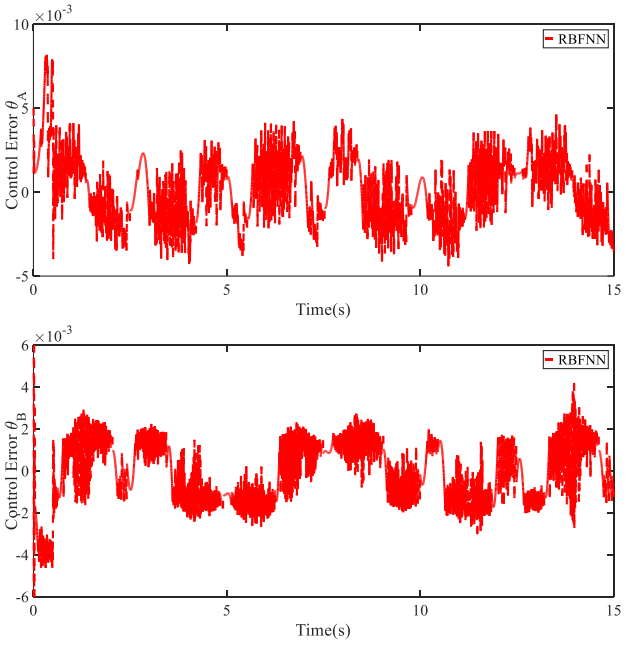


Fig. 9. Control error θ_A and θ_B

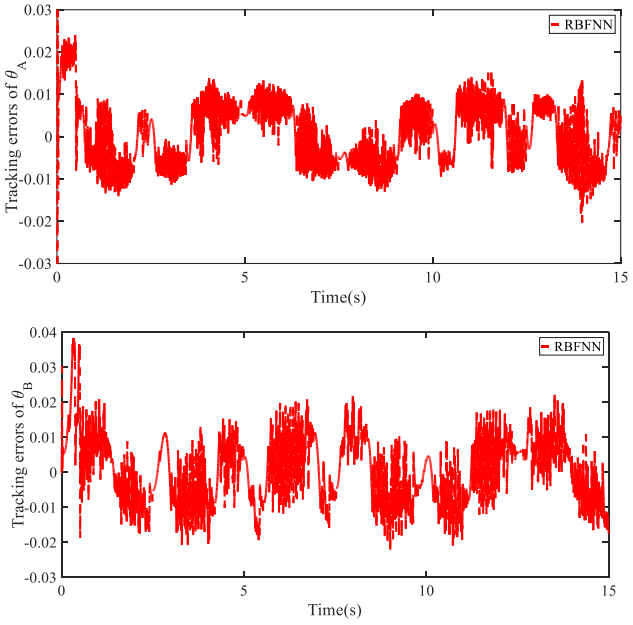


Fig. 10. Tracking error θ_A and θ_B

Fig. 11 shows the tracking ability of the RBFNN control system. The solid line shows the desired trajectory (Setpoint), while the red dashed line shows the actual trajectory (RBFNN). The proposed controller shows good tracking of the desired trajectory, but minor errors remain, especially at sharp corners or curves.

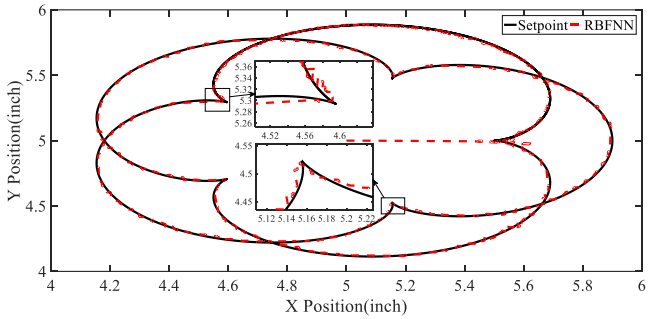


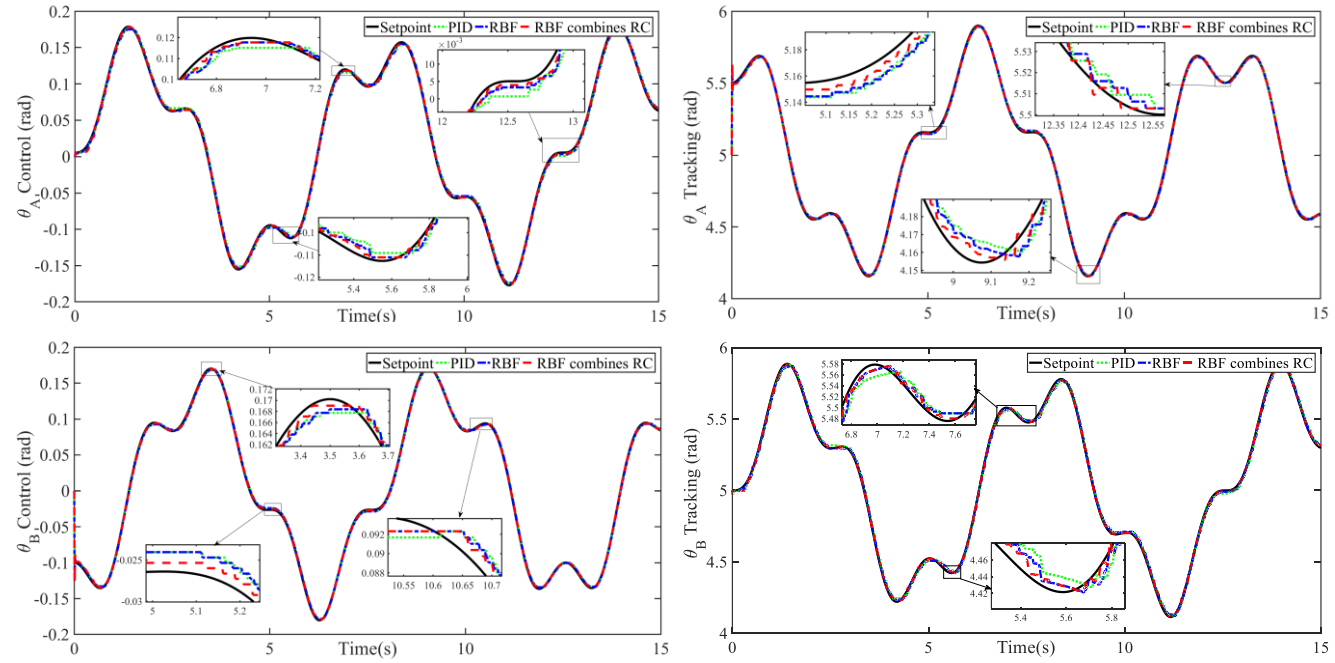
Fig. 11. Actual trajectory

C. *Comparison Results with Algorithms*

Fig. 12 shows the comparison results of different control methods, including PID, RBFNN, and RBFNN combined with RC, in tracking the setpoint for two control angles (θ_A and θ_B) over time. The graph is divided into two parts: control angles (a) and tracking angles (b). Both control methods closely track the setpoint for the control angles (θ_A and θ_B). However, when zooming in on specific regions, it can be seen that the RBFNN combined RC method (red dashed line) has superior precision, especially in minimizing errors and tracking the setpoint more accurately at critical times. The control angles show minimal deviation between the methods, but the RBFNN combined RC still shows smoother control, especially at the bends. Regarding tracking angle, the RBFNN-RC method also shows superior precision in tracking the desired trajectory, especially in fast transitions.

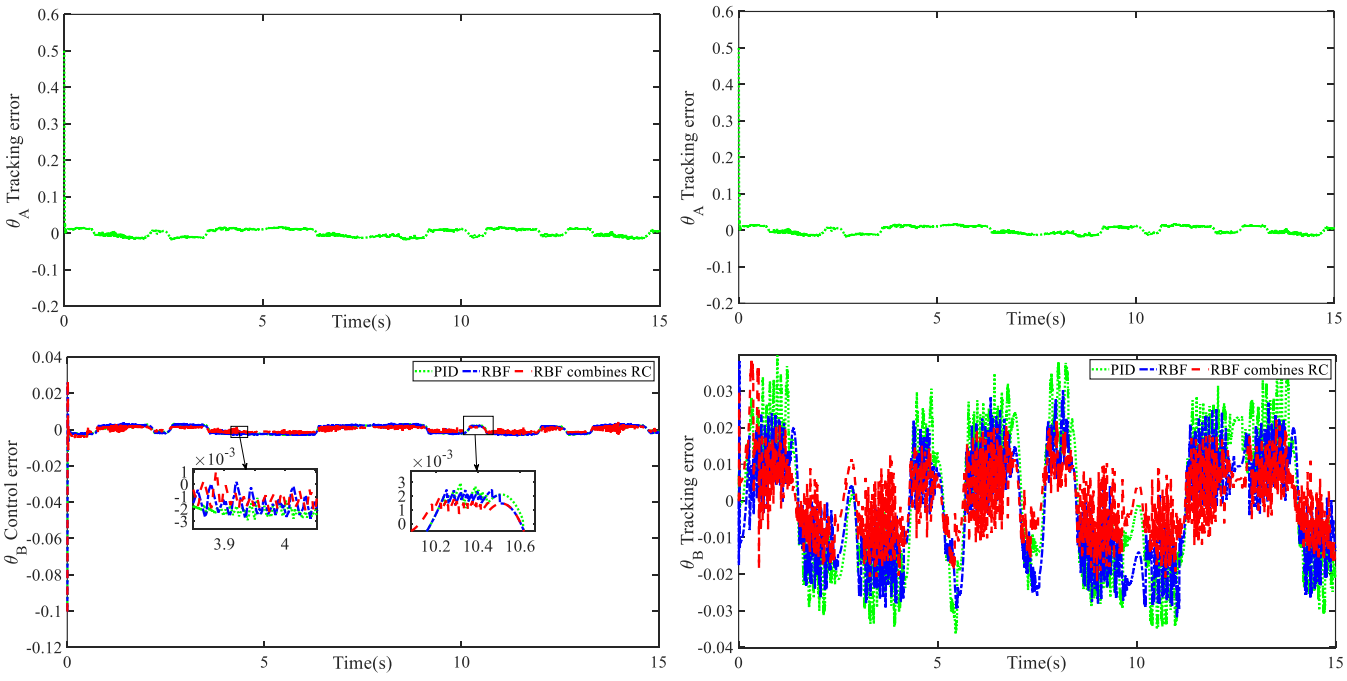
Fig. 13 shows the comparison between three control methods, PID, RBFNN, and RBFNN combined with RC,

through two main indicators: control error (a) and tracking error (b) of the corners (θ_A and θ_B). In the control error part (a), the combination of RBFNN and RC continues to show its superiority when the control error is maintained at a very low level, especially when there are curves. This result confirms that the RBFNN method combined with RC improves tracking precision and enhances control efficiency, which is especially suitable for systems that require fast and accurate response. Similarly, all three methods perform well in the tracking error part (b) when the error fluctuates around a value close to 0. Still, the RBFNN method combined with RC (red line) shows much better precision. The zoomed-in sections on the graph clearly illustrate this method's stability, which helps reduce small tracking errors that pure PID and RBFNN cannot do.



(a) Control angles (b) Tracking angles

Fig. 12. Actual positions of angles θ_A and θ_B at the control angles (a) and tracking angles (b) respectively



(a) Control angles (b) Tracking angles

Fig. 13. Error of angles θ_A and θ_B at the control angles and tracking angles respectivel

Fig. 14 shows a comparison between different control methods, including PID, RBFNN, and RBFNN combined with RC, in tracking the preset trajectory (black line). The results show that the PID controller (green dotted line) has significant deviations from the desired trajectory, especially at sharp curves, indicating less precision tracking. The RBFNN controller (blue dashed line) significantly improves the tracking performance compared to PID but still has some deviations in certain regions. Meanwhile, RBFNN combined with RC (red dashed line) performs superior trajectory tracking with the highest precision. The magnified regions clearly show the differences between the controllers and demonstrate that RBFNN combined with RC helps reduce the system's tracking error. On the other hand, the Lyapunov function shown in the 3D projection in Fig. 14 (b) shows that the stability of the system is guaranteed by Lyapunov. The component values in the Lyapunov function converge to zero, indicating that the system is always stable and approaches zero as time approaches infinity.

The data analyzed in Table II shows that the RBFNN-RC control method outperforms both PID and RBFNN in most of the indicators, including Control Error, Tracking Error, IAE, ISE, ITAE, and ITSE. This method provides better accuracy, reduces errors, and improves the system's response. Meanwhile, PID shows limitations, reducing reliability in systems requiring high accuracy. RBFNN has better results than PID but still cannot compare with RBFNN-RC regarding efficiency and stability.

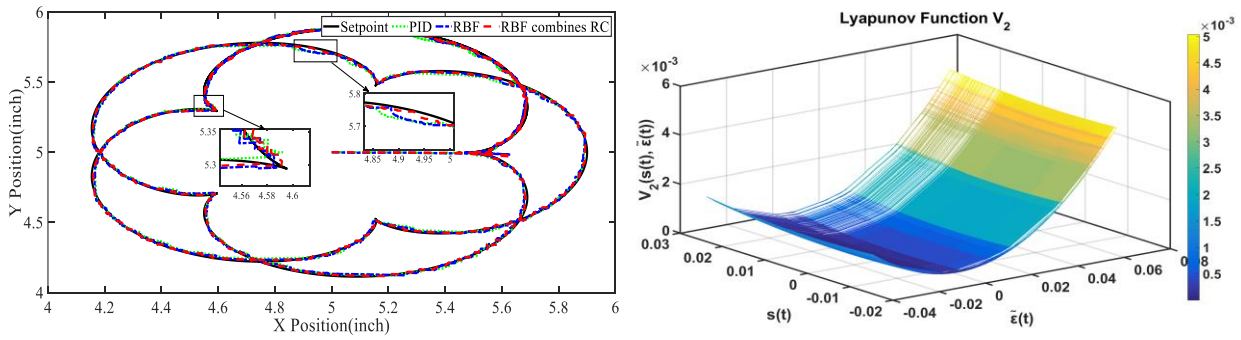
V. CONCLUSION AND DISCUSSION

This study marks a step forward in developing and applying intelligent control methods for uncertain nonlinear MIMO systems. The experimental object chosen is a five-bar parallel robot system characterized by high mechanical complexity, making constructing an accurate mathematical model a significant challenge. Moreover, the interconnected joints interact during operation, making the already difficult position tracking control even more difficult. The study has achieved remarkable results by applying the proposed control method, based on adaptive and learning capabilities, to improve the stability and accuracy during operation, as presented in Section V. To achieve these improvements, all the parameters of the RBFNN neural network are designed based on the gradient descent method, and the Lyapunov theory guarantees their stability. The experimental results show that the controller gradually stabilizes and the system accuracy improves as the adaptive parameters are reduced over time. However, this error reduction needs to be analyzed carefully to ensure the robustness and effectiveness of the controller. First, the error reduction is a positive sign that the system is converging to a steady state. However, this is only meaningful when the control parameters have been fully optimized. Second, the controller uses an adaptive algorithm, so the error reduction results from the parameter self-tuning process. However, the uneven reduction rate suggests that further adjustments may be needed to the controller parameters or the neural network learning weights, to improve the control performance regarding both response speed and accuracy. Finally, the error continues decreasing, but the system has not yet reached optimal stability. In that case, it may reflect limitations in the modeling or challenges in the robustness of the controller when deployed in a actual environment. The study employed a robust controller to enhance adaptability and respond to uncertainties such as disturbances, friction, etc.

In future studies, the authors aim to analyze further the stability and the ability to handle the uncertain component directly affecting the system. Periodic uncertainties are expected to be introduced during operation. This is possible because the experiments in the present study were performed on a model in which the random uncertainty component already exists in the system. The long-term goal is to develop an adaptive and stable control method widely applied to any nonlinear and uncertain MIMO system.

TABLE II. PARAMETERS OF ACTUAL ROBOT SYSTEM OPERATION OF CONTROL METHODS ($\times 10^{-3}$)

	Methods	Control Error	Tracking Error	IAE	ISE	ITAE	ITSE
θ_A	PID	0.494	4.55	24.9	0.101	127	4.34
	RBFNN	0.332	1.17	13.5	0.0565	66.8	2.98
	RBFNN-RC	0.166	0.829	12.4	0.0315	56	1.18
θ_B	PID	-0.163	0.738	15.5	0.0966	73.7	-5.79
	RBFNN	-0.0261	0.583	12.4	0.0878	57.4	-3.45
	RBFNN-RC	-0.0131	0.382	10.5	0.0825	42.7	-2.33



(a) The actual trajectory of the algorithms

(b) 3D drawing of Lyapunov function in actual

Fig. 14. Tracking trajectories (a) and Lyapunov functions of the proposed method (b)

REFERENCES

[1] C. C. Cheah, S. P. Hou, Y. Zhao, and J. -J. E. Slotine, "Adaptive Vision and Force Tracking Control for Robots With Constraint Uncertainty," *IEEE/ASME Transactions on Mechatronics*, vol. 15, no. 3, pp. 389-399, 2010.

[2] Z. Li, S. S. Ge, M. Adams, and W. S. Wijesoma, "Robust adaptive control of uncertain force/motion constrained nonholonomic mobile manipulators," *Automatica*, vol. 44, no. 3, pp. 776-784, 2008.

[3] L. Huang, S. S. Ge, and T. Lee, "Position/force control of uncertain constrained flexible joint robots," *Mechatronics*, vol. 16, no. 2, pp. 111-120, 2006.

[4] Z. Li, J. Li, and Y. Kang, "Adaptive robust coordinated control of multiple mobile manipulators interacting with rigid environments," *Automatica*, vol. 46, no. 12, pp. 2028-2034, 2010.

[5] Z. Li, S.S. Ge, and Z. Wang, "Robust adaptive control of coordinated multiple mobile manipulators," *Mechatronics* vol. 18, no. 5-6, pp. 239-250, 2008.

[6] K. P. Tee, R. Yan, and H. Li, "Adaptive admittance control of a robot manipulator under task space constraint," *2010 IEEE International Conference on Robotics and Automation*, pp. 5181-5186, 2010.

[7] H. N. Rahimi, I. Howard, and L. Cui, "Neural impedance adaption for assistive human–robot interaction," *Neurocomputing*, vol. 290, pp. 50-59, 2018.

[8] O. Cerman and P. Hušek, "Adaptive fuzzy sliding mode control for electro-hydraulic servo mechanism," *Expert Systems with Applications*, vol. 39, no. 11, pp. 10269-10277, 2012.

[9] Q. K. Li, J. Zhao, and G. M. Dimirovski, "Robust tracking control for switched linear systems with time-varying delays," *IET Control Theory & Applications*, vol. 2, no. 6, pp. 449-457, 2008.

[10] M. Roopaei, M. Zolghadri, and S. Meshksar, "Enhanced adaptive fuzzy sliding mode control for uncertain nonlinear systems," *Communications in Nonlinear Science and Numerical Simulation*, vol. 14, no. 9, pp. 3670-3681, 2009.

[11] F. Liu, M. Chen, and T. Li, "Fixed-time anti-disturbance control for nonlinear turbofan engine with impulsive prescribed performance," *Journal of the Franklin Institute*, vol. 361, no. 17, p. 107193, 2024.

[12] Y. Zhang, J. Sun, H. Liang, and H. Li, "Event-Triggered Adaptive Tracking Control for Multiagent Systems With Unknown Disturbances," *IEEE Transactions on Cybernetics*, vol. 50, no. 3, pp. 890-901, 2020.

[13] C. K. Verginis, Y. Kantaros, and D. V. Dimarogonas, "Planning and control of multi-robot-object systems under temporal logic tasks and uncertain dynamics," *Robotics and Autonomous Systems*, vol. 174, p. 104646, 2024.

[14] S. J. Yoo and B. S. Park, "Distributed quantized state feedback strategy for ensuring predesignated formation tracking performance of networked uncertain nonholonomic multi-robot systems with quantized communication," *Expert Systems with Applications*, vol. 201, p. 116987, 2022.

[15] X. Liu and H. Sheng, "Active fault tolerant control of uncertain robotic system based on observer and sliding mode," *IFAC-PapersOnLine*, vol. 55, no. 1, pp. 598-603, 2022.

[16] A. Zhai, J. Wang, H. Zhang, G. Lu, and H. Li, "Adaptive robust synchronized control for cooperative robotic manipulators with uncertain base coordinate system," *ISA Transactions*, vol. 126, pp. 134–143, 2022.

[17] F. Ye, J. Perrett, L. Zhang, Y. Laili, and Y. Wang, "A self-evolving system for robotic disassembly sequence planning under uncertain interference conditions," *Robotics and Computer-Integrated Manufacturing*, vol. 78, p. 102392, 2022.

[18] M. Souzanchi-K and M. R. Akbarzadeh-T, "Brain emotional learning impedance control of uncertain nonlinear systems with time delay: Experiments on a hybrid elastic joint robot in telesurgery," *Computers in Biology and Medicine*, vol. 138, p. 104786, 2021.

[19] Z. Li and S. Li, "Saturated PI Control for Nonlinear System With Provable Convergence: An Optimization Perspective," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 68, no. 2, pp. 742-746, 2021,

[20] T. Q. Ngo, T. N. A. Nguyen, N. T. P. Le, D. C. Pham, and N. D. Ngo, "Adaptive Tracking Control Based on Recurrent Wavelet Fuzzy CMAC for Uncertain Nonlinear Systems," *International Journal of Control and Automation*, vol. 11, no. 1, pp. 75-90, 2018.

[21] A. -V. Nguyen, V. -T. Ngo, W. -J. Wang, V. -P. Vu, T. -Q. Ngo, and A. –. Nguyen, "Fuzzy Logic Based LQG Controller Design for Inverted Pendulum On Cart," *2021 International Conference on System Science and Engineering (ICSSE)*, pp. 387-392, 2021.

[22] M. A. Llama, R. Kelly, and V. Santibanez, "Stable computed-torque control of robot manipulators via fuzzy self-tuning," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 30, no. 1, pp. 143-150, 2000.

[23] T. Sun, L. Cheng, W. Wang, and Y. Pan, "Semiglobal exponential control of Euler–Lagrange systems using a sliding-mode disturbance observer," *Automatica*, vol. 112, p. 108677, 2020.

[24] M. Chen, G. Tao, and B. Jiang, "Dynamic Surface Control Using Neural Networks for a Class of Uncertain Nonlinear Systems With Input Saturation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 9, pp. 2086-2097, 2015.

[25] T. Yang, N. Sun, and Y. Fang, "Adaptive Fuzzy Control for Uncertain Mechatronic Systems With State Estimation and Input Nonlinearities," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 3, pp. 1770-1780, 2022.

[26] S. Roy, S. Baldi, and L. Fridman, "On adaptive sliding mode control without a priori bounded uncertainty," *Automatica*, vol. 111, p. 108650, 2020.

[27] Q. Deng, Y. Peng, T. Han, and D. Qu, "Event-Triggered Bipartite Consensus in Networked Euler–Lagrange Systems With External Disturbance," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 68, no. 8, pp. 2870-2874, 2021.

[28] J. Han, "From PID to Active Disturbance Rejection Control," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 3, pp. 900-906, 2009.

[29] W. -H. Chen, J. Yang, L. Guo, and S. Li, "Disturbance-Observer-Based Control and Related Methods-An Overview," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 2, pp. 1083-1095, 2016.

[30] W. He, Y. Sun, Z. Yan, C. Yang, Z. Li, and O. Kaynak, "Disturbance Observer-Based Neural Network Control of Cooperative Multiple Manipulators With Input Saturation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 5, pp. 1735-1746, 2020.

[31] M. Basin and P. Ramirez, "Sliding mode controller design for linear systems with unmeasured states," *Journal of the Franklin Institute*, vol. 349, no. 4, pp. 1337-1349, 2012.

[32] H. Wu and P. Shi, "Adaptive variable structure state estimation for uncertain systems with persistently bounded disturbances," *International Journal of Robust and Nonlinear Control*, vol. 20, no. 17, pp. 2003-2015, 2010.

[33] L. Wu, P. Shi, and H. Gao, "State Estimation and Sliding-Mode Control of Markovian Jump Singular Systems," *IEEE Transactions on Automatic Control*, vol. 55, no. 5, pp. 1213-1219, 2010.

[34] Y. Xia, H. Yang, M. Fu, and P. Shi, "Sliding mode control for linear systems with time-varying input and state delays," *Circuits, Systems, and Signal Processing*, vol. 30, no. 3, pp. 629-641, 2011.

[35] İ. Eker, "Second-order sliding mode control with experimental application," *ISA Transactions*, vol. 49, no. 3, pp. 394-405, 2010.

[36] H. F. Ho, Y. K. Wong, and A. B. Rad, "Adaptive fuzzy sliding mode control with chattering elimination for nonlinear SISO systems," *Simulation Modelling Practice and Theory*, vol. 17, pp. 1199-1210, 2009.

[37] J. Hu, Z. Wang, H. Gao, and L. K. Stergioulas, "Robust H∞ sliding mode control for discrete time-delay systems with stochastic nonlinearities," *Journal of the Franklin Institute*, vol. 349, pp. 1459-1479, 2012.

[38] T. Sun, H. Pei, Y. Pan, H. Zhou, and C. Zhang, "Neural network-based sliding mode adaptive control for robot manipulators," *Neurocomputing*, vol. 74, no. 14, pp. 2377–2384, 2011.

[39] M. Shahriari Kahkeshi, F. Sheikholeslam, and M. Zekri, "Design of adaptive fuzzy wavelet neural sliding mode controller for uncertain nonlinear systems," *ISA Transactions*, vol. 52, no. 3, pp. 342–350, 2013.

[40] B. Ren, Q. -C. Zhong, and J. Chen, "Robust Control for a Class of Nonaffine Nonlinear Systems Based on the Uncertainty and

Disturbance Estimator," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 9, pp. 5881-5888, 2015.

[41] B. Ren, Q. -C. Zhong, and J. Dai, "Asymptotic Reference Tracking and Disturbance Rejection of UDE-Based Robust Control," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 4, pp. 3166-3176, 2017.

[42] P. Shendge and B. Patre, "Robust model following load frequency sliding mode controller based on UDE and error improvement with higher order filter," *IAENG International Journal of Applied Mathematics*, vol. 37, no. 1, pp. 27-32, 2007.

[43] Q. Zhong and D. Rees, "Control of uncertain LTI systems based on an uncertainty and disturbance estimator," *Journal of Dynamic Systems, Measurement, and Control, Transactions of the ASME*, vol. 126, no. 4, pp. 905-910, 2004.

[44] B. Cong, X. Liu, and Z. Chen, "Backstepping based adaptive sliding mode control for spacecraft attitude maneuvers," *Aerospace Science and Technology*, vol. 30, no. 1, pp. 1-7, 2013.

[45] M. Rafiq, S. Rehman, F. Rehman, Q. Butt, and I. Awan, "A second order sliding mode control design of a switched reluctance motor using super twisting algorithm," *Simulation Modelling Practice and Theory*, vol. 25, pp. 106-117, 2012.

[46] G. R. Chen, J. Zhou, and S. Celikovsky, "On LaSalle's invariance principle and its application to robust synchronization of general vector Liénard equations," *IEEE Transactions on Automatic Control*, vol. 50, no. 6, pp. 869–874, 2005.

[47] V. Lakshmikantham, S. Leela, and A. A. P. Martynyuk, *Practical Stability of Nonlinear Systems*. World Scientific, 1990.

[48] Y. J. Huang, T. C. Kuo, and S. H. Chang, "Adaptive sliding-mode control for nonlinear systems with uncertain parameters," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 38, pp. 534–539, 2008.

[49] M. Zeinali and L. Notash, "Adaptive sliding mode control with uncertainty estimator for robot manipulators," *Mechanism and Machine Theory*, vol. 45, no. 1, pp. 80-90, 2010.

[50] J. H. Liu, C. L. Wang, and X. Cai, "Adaptive neural network finite-time tracking control for a class of high-order nonlinear multi-agent systems with powers of positive odd rational numbers and prescribed performance," *Neurocomputing*, vol. 419, pp. 157-167, 2021.

[51] Z. Wang, Y. Z. Zhu, H. Xue, and H. J. Liang, "Neural networks-based adaptive event-triggered consensus control for a class of multi-agent systems with communication faults," *Neurocomputing*, vol. 470, pp. 99-108, 2022.

[52] X. Zhao, P. Shi, X. Zheng, and J. Zhang, "Intelligent tracking control for a class of uncertain high-order nonlinear systems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 21, no. 9, pp. 1976-1982, 2019.

[53] W. He, Y. Dong, and C. Sun, "Adaptive neural impedance control of a robotic manipulator with input saturation," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 46, no. 3, pp. 334-344, 2016.

[54] R. Yang, C. Yang, M. Chen, and A. S. Annamalai, "Discrete-time optimal adaptive RBFNN control for robot manipulators with uncertain dynamics," *Neurocomputing*, vol. 234, pp. 107-114, 2017.

[55] Z. Wang, J. Yuan, Y. Pan, and J. Wei, "Neural network-based adaptive fault tolerant consensus control for a class of high order multi-agent systems with input quantization and time-varying parameters," *Neurocomputing*, vol. 266, pp. 315–324, 2017.

[56] H. R. Nohooji, I. Howard, and L. Cui, "Neural adaptive assist-as-needed control for rehabilitation robots," *Proceedings of the Australasian Conference on Robotics and Automation (ACRA)*, 2016.

[57] M. Daachi, T. Madani, B. Daachi, and K. Djouani, "A radial basis function neural network adaptive controller to drive a powered lower limb knee joint orthosis," *Applied Soft Computing*, vol. 34, pp. 324–336, 2015.

[58] H. R. Nohooji, "Constrained neural adaptive PID control for robot manipulators," *Journal of the Franklin Institute*, vol. 357, no. 7, pp. 3907–3923, 2020.

[59] C. Lauretti, F. Cordella, A. L. Ciancio, E. Trigili, J. M. Catalan, F. J. Badesa, S. Crea, S. M. Pagliara, S. Sterzi, and N. Vitiello, "Learning by demonstration for motion planning of upper-limb exoskeletons," *Frontiers in Neurorobotics*, vol. 12, p. 5, 2018.

[60] S. Seshagiri and H. K. Khalil, "Output feedback control of nonlinear systems using RBF neural networks," *IEEE Transactions on Neural Networks*, vol. 11, no. 1, pp. 69–79, 2000.

[61] R. J. Schalkoff, *Artificial Neural Networks*. McGraw-Hill, 1997.

[62] M. R. Berthold and J. Diamond, "Boosting the performance of RBF networks with dynamic decay adjustment," *Advances in Neural Information Processing Systems*, 1994.

[63] S. Zhao and J. Yuh, "Experimental study on advanced underwater robot control," *IEEE Transactions on Robotics*, vol. 21, no. 4, pp. 695–703, 2005.

[64] Y. Lu, J. K. Liu, and F. C. Sun, "Actuator Nonlinearities Compensation Using RBF Neural Networks in Robot Control System," *The Proceedings of the Multiconference on "Computational Engineering in Systems Applications"*, pp. 231-238, 2006.

[65] W. Hao, "A gradient descent method for solving a system of nonlinear equations," *Applied Mathematics Letters*, vol. 112, p. 106739, 2021.

[66] T. Q. Ngo, T. H. Tran, V. T. Nguyen, and T. T. H. Le, "Adaptive single input tracking controller for parallel robot system based on hybrid brain emotion and cerebellar model articulation control network using wavelet function," *Journal of Electrical Engineering and Technology*, 2025.