Hindawi
Complexity
Volume 2020, Article ID 3172431, 12 pages
https://doi.org/10.1155/2020/3172431

Research Article

Weighted Multiple-Model Neural Network Adaptive Control for Robotic Manipulators with Jumping Parameters

Jiazhi Li,[1] **Weicun Zhang**,[1] **and Quanmin Zhu**[2]

[1]*Key Laboratory of Knowledge Automation for Industrial Processes of Ministry of Education, School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing 100083, China*
[2]*Department of Engineering Design and Mathematics, University of the West of England, Bristol BS161QY, UK*

Correspondence should be addressed to Weicun Zhang; weicunzhang@263.net

Received 4 September 2019; Revised 19 December 2019; Accepted 31 December 2019; Published 22 January 2020

Academic Editor: Xianggui Guo

This study addresses the tracking control issue for n-link robotic manipulators with largely jumping parameters. Based on radial basis function neural networks (RBFNNs), we propose weighted multiple-model neural network adaptive control (WMNNAC) approach. To cover the variation ranges of the parameters, different models of robotic are constructed. Then, the corresponding local neural network controller is constructed, in which the neural network has been used to approximate the uncertainty part of the control law, and an adaptive observer is implemented to estimate the true external disturbance. The WMNNAC strategy with improved weighting algorithm is adopted to ensure the tracking performance of the robotic manipulator system when parameters jump largely. Through the Lyapunov stability theory and the method of virtual equivalent system (VES), the stability of the closed-loop system is proved. Finally, the simulation results of a two-link manipulator verify the feasibility and efficiency of the proposed WMNNAC strategy.

1. Introduction

Robotic manipulators are highly coupled, time-varying, and multivariable nonlinear dynamic systems. Due to the complexity of the system, the control problem has been extensively studied [1–4] in recent years.

It should be noted that dynamic uncertainties in robotic system models are unavoidable due to the unknown load, mass, etc., and such modeling uncertainties may lead to a degradation on the control accuracy or even cause instability of the robotic system. The modeling uncertainties can be divided as structured uncertainties and unstructured uncertainties. For structured uncertainties, adaptive control has been widely used as an effective method for controlling complex systems in the field of robotic control [5–12] in recent years. In order to deal with the unstructured uncertainties of robotic manipulator systems, various learning-based control methods [13–17], including neural network (NN) and fuzzy systems, have been proposed to overcome them. As an effective method for approximating arbitrary nonlinear continuous functions with arbitrary precision, NN technology is very mature in modeling complex processes and compensating for unstructured uncertainties [14, 16, 18–22].

However, most of the abovementioned references only consider the case that the robotic dynamic parameters are fixed or changed slowly. In real applications, robots often pick up or lay down different loads abruptly, which is known as large parameter uncertainties of robotic dynamics. Thus, the control performance of the abovementioned methods did not consider the jumping parameters which may deteriorate. Moreover, the system identification rate of conventional adaptive control is comparatively slow, which makes the transient performance of robotic systems poor in this situation [23]. As the robotic system with uncertain parameters is very complex, some researchers combine multiple-model strategy with neural networks to improve the control performance in recent years [24, 25].

Multiple-model control strategy is a solution for controlling complex systems with jumping parameters [26, 27]. The goal of multiple-model control is to determine the most appropriate controller at any time based on identification of errors and appropriate criteria. In the past few years, switching multiple-model adaptive control [28–30] has been widely used to deal with the robotic system with jumping parameters. However, the switching between the controllers may make the control performance nonsmooth. In order to make the transition process smoother, we introduce weighted multiple-model adaptive control (WMMAC) approach [31–35]. Through the WMMAC method, the control range can cover the variation range of system parameter change, which can solve the control problem of the complex nonlinear system well when parameters change or jump unexpectedly. Moreover, it enhances the robustness of the system, effectively reduce the model identification time, and decrease the system transient error. As we all know, it is rather difficult to prove the closed-loop stability of the WMMAC system. Fortunately, in [36–38], some new results are presented on the system stability based on VES theory in recent years.

Aiming at stabilizing the two-link robotic manipulator with largely jumping parameters, uncertainties, and external disturbance, a WMMAC scheme combining multiple NN-based controllers is proposed. The main contributions of this paper are summarized as follows:

(i) A WMNNAC scheme is proposed for robotic manipulators to deal with dynamic uncertainties and largely jumping parameters. It can improve the transient performance of the system.

(ii) The disturbance observer is constructed based on the RBFNNs. By online estimation, the error of observation is reduced.

(iii) In this paper, a modified weighting algorithm is given under the structure of robotic manipulators. Through VES theory, the stability and convergence of the robotic system are analyzed.

The rest of this paper is organized as follows. Section 2 gives the preliminaries. Section 3 is divided into three parts which successively describe the design process of the local controller and global controller and stability analysis of the controller against the jumping parameters in practice. Section 4 gives the simulation results to verify the feasibility of the developed control method. Finally, Section 5 concludes the paper.

Notations. Throughout this paper, the superscript T stands for matrix transposition. $\mathbb{R}$ denotes the space of real numbers; $\mathbb{R}^n$ denotes the n-dimensional Euclidean space with the vector norm $\|\cdot\|$; and $\mathbb{R}^{n\times m}$ is the set of all $n \times m$ real matrices. $\lambda_{\min}(\cdot)$ and $\lambda_{\max}(\cdot)$ are the minimum and maximum eigenvalues of matrix $\cdot$, respectively. $|*|$ denotes taking the absolute values of all the elements in the vector $*$.

2. Preliminaries

2.1. Model of Robotic Manipulators. Considering the disturbance, the dynamics model of the n-link rigid robotic can be drawn by the Lagrange equation as

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = u - d, \qquad (1)$$

where $q, \dot{q}, \ddot{q} \in \mathbb{R}^n$ represent the vector of the joint angular position, corresponding velocity, and acceleration, respectively; $u \in \mathbb{R}^n$ is the applied joint torque; $M(q) \in \mathbb{R}^{n\times n}$ is the inertia matrix; $C(q,\dot{q}) \in \mathbb{R}^{n\times n}$ denotes the Coriolis and centrifugal force; $G(q) \in \mathbb{R}^n$ is the gravity items; and $d \in \mathbb{R}^n$ is the external disturbance. In this study, the following properties of system (1) with revolute joints are available [39, 40].

Property 1. $M(q)$ is symmetric and positive definite and is bounded by the following inequalities: $m_a \le \|M(q)\| \le m_b$, where m_a and m_b are positive constants.

Property 2. $(1/2)\dot{M}(q) - C(q,\dot{q})$ is skew-symmetric, i.e., $x^T[\dot{M}(q) - 2C(q,\dot{q})]x = 0$, $\forall x \in \mathbb{R}^n$.

The following assumption is used in this paper.

Assumption 1. We assume that the disturbance $d(t)$ is continuous and uniformly bounded, i.e., there exists a constant $\overline{d} > 0$ such that $|d(t)| \le \overline{d}$, $\forall t \in [0,\infty)$ and a constant $\overline{d}' > 0$ such that $|\dot{d}(t)| \le \overline{d}'$, $\forall t \in [0,\infty)$.

2.2. RBFNN. RBFNN is a special NN architecture with some advantages. In the field of robotic control, RBFNN has been widely used owing to the powerful approximation ability to nonlinear functions. RBFNN has a simple structure with input, output, and hidden layers. The input layer is simply a fan-out layer, and the input vector is donated as $Z = [Z_1, Z_2, \ldots, Z_q]^T \in \Omega_Z \subset \mathbb{R}^q$. In the hidden layer, Gaussian kernel functions are selected as activation functions, that is,

$$(h)_j = \exp\left[\frac{-(Z-\mu_j)^T(Z-\mu_j)}{b_j^2}\right], \quad j = 1, 2, \ldots, m, \qquad (2)$$

where $\mu_j = [\mu_{j1}, \mu_{j2}, \ldots, \mu_{jq}]$ represents the center value of the jth node, NN node number $m > 1$, and b_j is the value of width [41].

The output layer computes the output value through the vector of connection weights. In this paper, RBFNN is utilized to approximate the continuous function, where the weight vector $W_i \in \mathbb{R}^m$. There exist an ideal constant weight vector W_i^* and the bounded approximation error $\varepsilon_i(Z)$, i.e., $|\varepsilon_i(Z)| < \overline{\varepsilon}_i$ with $\overline{\varepsilon}_i > 0$ for all input $Z \in \Omega_Z$. This is computed as

$$f_i(Z) = W_i^{*T} H_i(Z) + \varepsilon_i(Z), \quad \forall Z \in \Omega_Z \subset \mathbb{R}^q, i = 1, 2, \ldots, n, \qquad (3)$$

where $H_i = [h_1, h_2, \ldots, h_m]^T$.

2.3. Useful Mathematic Tools

Definition 1. $\operatorname{sgn}(a) \in \mathbb{R}^n$ is defined as

$$\operatorname{sgn}(a) = [\operatorname{sign}(a_1), \operatorname{sign}(a_2), \ldots, \operatorname{sign}(a_n)]^T, \qquad (4)$$

where a is an n-dimensional vector and $\text{sign}(\cdot)$ is the standard signum function.

Definition 2. $b^t \in \mathbb{R}^n$ is defined as

$$b^t = \left[|b_1|^t \text{sign}(b_1), |b_2|^t \text{sign}(b_2), \ldots, |b_n|^t \text{sign}(b_n)\right]^T, \quad (5)$$

where b is an n-dimensional vector.

Definition 3. The operator "$\odot$" is defined as

$$c \odot d = [c_1 d_1, c_2 d_2, \ldots, c_2 d_2]^T, \quad (6)$$

where c and d are two n-dimensional vectors.

3. Control Design

In this study, the control objective is that the robotic manipulators can still track the desired trajectory well with good transient performance when the system parameters jump. In order to achieve this objective, we develop the WMNNAC. The design process is given as follows: firstly, according to the variation ranges of parameters, we create multiple submodels to built a model set. Secondly, corresponding local NN-based controllers are designed by Lyapunov theory, and stable local closed-loop systems can be obtained. Finally, an improved weighting algorithm is proposed for the characteristics of the robotic system. Based on the new algorithm, the weights corresponding to each local controller are obtained at each moment. It is used for fusing local controllers to generate the global control signals.

Obviously, the controller design is divided into two parts: the local controller and the global controller. The concise block diagram of the proposed WMNNAC system is shown in Figure 1, where $u(t)$ is the global control signal, $y(t)$ is the output of robotic manipulators, $y_\xi(t)$ is the output of the local model, NNC_ξ is the ξth local NN-based controller, and $u_\xi(t)$ and $p_\xi(t)$ are the corresponding control signal and weight, $\xi = 1, 2, \ldots, N$.

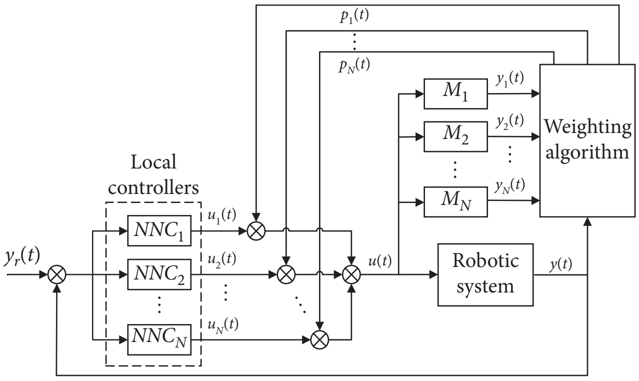


FIGURE 1: Concise block diagram of the WMNNAC system.

3.1. Local Controller Design Based on RBFNNS.

We first consider the design of the local controller. When the parameters of the robotic are known, the control objectives can be well achieved. However, the model of the robotic includes uncertainties and disturbance, and the accurate values are unknown in the actual system. Considering the complexity of manipulator dynamics and the effective learning mechanism of NN, the RBFNN adaptive controller is developed to eliminate the influence of uncertainties. Furthermore, we give a generalized error to speed up the recovery after system instability. Meanwhile, we design an adaptive observer to compensate for external disturbances.

Let $x_1 = q$ and $x_2 = \dot{q}$, and the robotic dynamics model (1) is described as

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= M^{-1}[u - d - Cx_2 - G]. \end{aligned} \quad (7)$$

Define a tracking error z_1 and a second error by introducing a virtual control $\alpha_1 = -K_1 z_1 + \dot{x}_d$, $K_1 \in \mathbb{R}^n$ is a gain matrix:

$$\begin{aligned} z_1 &= x_1 - x_d, \\ z_2 &= x_2 - \alpha_1. \end{aligned} \quad (8)$$

Differentiating z_1 and taking the time derivative of z_2 lead to

$$\begin{aligned} \dot{z}_1 &= z_2 + \alpha_1 - \dot{x}_d = -K_1 z_1 + z_2, \\ \dot{z}_2 &= M^{-1}(u - d - Cx_2 - G) - \dot{\alpha}_1. \end{aligned} \quad (9)$$

Then, we propose the first Lyapunov function candidate as

$$V_1 = \frac{1}{2} z_1^T z_1. \quad (10)$$

Differentiating V_1 yields

$$\dot{V}_1 = -z_1^T K_1 z_1 + z_1^T z_2. \quad (11)$$

Choosing a generalized error yields

$$s = z_1 + \frac{1}{\beta} z_2^{\varphi}, \quad (12)$$

where $\beta = \text{diag}(\beta_1, \beta_2, \ldots, \beta_n)$ with $\beta_i > 0$ and φ satisfies the following condition: $1 < \varphi < 2$. Then, $\dot{s}$ can be written as

$$\dot{s} = \dot{z}_1 + \varphi \frac{1}{\beta} \dot{z}_2 \odot |z_2|^{\varphi - 1}. \quad (13)$$

Consider a new Lyapunov function candidate as follows:

$$V_2 = \frac{1}{2} z_1^T z_1 + \frac{1}{2} s^T M s. \quad (14)$$

Differentiating V_2 with respect to time, we obtain

$$\begin{aligned}\dot{V}_2 &= \dot{V}_1 + s^T M\dot{s} + \frac{1}{2}s^T\dot{M}s\\ &= -z_1^T K_1 z_1 + z_1^T z_2 + s^T Cs + s^T M\left(\dot{z}_1 + \varphi\frac{1}{\beta}\dot{z}_2 \odot \left|z_2\right|^{\varphi-1}\right)\\ &= -z_1^T K_1 z_1 + z_1^T z_2 + s^T\left(Cs + M\dot{z}_1\right)\\ &\quad + s^T\varphi\frac{1}{\beta}\left(u - d - Cx_2 - G - M\dot{\alpha}_1\right)\odot\left|z_2\right|^{\varphi-1}.\end{aligned} \tag{15}$$

According to Assumption 1, we can obtain the upper boundary of the disturbance. However, the upper boundary of external disturbance is usually unknown and cannot be measured. We construct a disturbance observer $\widehat{d}$ to solve these problems. Define the estimation error $\widetilde{d} = d - \widehat{d}$. Then, a new Lyapunov function is given as follows:

$$V_3 = V_2 + \frac{1}{2}\widetilde{d}^T\Gamma_1^{-1}\widetilde{d}. \tag{16}$$

Its time derivative can be written as

$$\begin{aligned}\dot{V}_3 &= \dot{V}_2 + \widetilde{d}^T\Gamma_1^{-1}\left(\dot{d} - \dot{\widehat{d}}\right)\\ &\le -z_1^T K_1 z_1 + z_1^T z_2 + s^T\left(Cs + M\dot{z}_1\right)\\ &\quad + s^T\varphi\frac{1}{\beta}\left(u - \widehat{d} - Cx_2 - G - M\dot{\alpha}_1\right)\odot\left|z_2\right|^{\varphi-1}\\ &\quad - \sum_{i=1}^{n}\widetilde{d}_i\Gamma_{1i}^{-1}\left(\Gamma_{1i}s_i^T\varphi\frac{1}{\beta_i}\odot\left|z_{2i}\right|^{\varphi-1} + \dot{\widehat{d}}_i\right) + \widetilde{d}^T\Gamma_1^{-1}\overline{d}'.\end{aligned} \tag{17}$$

When M, C, and G are known, the controller can be designed as

$$\begin{aligned}u_0 &= Cx_2 + G + M\dot{\alpha}_1 + \widehat{d} - \varphi^{-1}\beta\left(Cs + M\dot{z}_1\right)\odot\left|z_2\right|^{1-\varphi}\\ &\quad - \varphi^{-1}\beta\left(\frac{s}{\|s\|^2}z_1^T z_2 + \eta|s|\operatorname{sign}(s)\right)\odot\left|z_2\right|^{1-\varphi}.\end{aligned} \tag{18}$$

Also, the controller can be designed as

$$\dot{\widehat{d}}_i = -\Gamma_{1i}\left(s_i^T\varphi\frac{1}{\beta_i}\odot\left|z_{2i}\right|^{\varphi-1} + \delta_{1i}\widehat{d}_i\right), \tag{19}$$

where Γ_1 is the constant matrix and δ_1 is a positive constant. Plugging (18) and (19) into (17) yields

$$\begin{aligned}\dot{V}_3 &\le -z_1^T K_1 z_1 - \eta\|s\|^2 + \sum_{i=1}^{n}\widetilde{d}_i\Gamma_{1i}^{-1}d'_{1i} + \sum_{i=1}^{n}\widetilde{d}_i\delta_{1i}\widehat{d}_i\\ &\le -z_1^T K_1 z_1 - \eta\|s\|^2 - \frac{1}{2}\sum_{i=1}^{n}\left(\delta_{1i} - \Gamma_{1i}^{-1}\right)\widetilde{d}_i^2 + \frac{1}{2}\sum_{i=1}^{n}\Gamma_{1i}^{-1}\overline{d'}_i^2\\ &\quad + \frac{1}{2}\sum_{i=1}^{n}\delta_{1i}\overline{d}_i^2\\ &\le -\rho_1 V_3 + c_1,\end{aligned} \tag{20}$$

where ρ_1 and c_1 are two positive constants given as

$$\begin{aligned}\rho_1 &= \min\left(2\lambda_{\min}\left(K_1\right), \frac{2\eta}{\lambda_{\max}(M)}, \min_{i=1,2,\ldots,n}\frac{\delta_{1i} - \Gamma_{1i}^{-1}}{\lambda_{\max}\left(\Gamma_1^{-1}\right)}\right),\\ c_1 &= \frac{1}{2}\sum_{i=1}^{n}\Gamma_{1i}^{-1}\overline{d'}_i^2 + \frac{1}{2}\sum_{i=1}^{n}\delta_{1i}\overline{d}_i^2.\end{aligned} \tag{21}$$

To guarantee $\rho_1 > 0$, the control gain Γ_{1i} and δ_{1i} is selected to satisfy

$$\min_{i=1,2,\ldots,n}\left(\delta_{1i} - \Gamma_{1i}^{-1}\right) > 0. \tag{22}$$

Since uncertainties sometimes unavoidably exist in M, C, and G, the aforementioned control law, which is built on the assumption that the exact parameters can be obtained, is not applicable for the robotic system. Thus, we utilize RBFNNs to approximate the uncertainties by the online estimation.

$$\begin{aligned}W^{*T}H(Z) &= \left(-Cx_2 - G - M\dot{\alpha}_1\right)\odot\left|z_2\right|^{\varphi-1}\\ &\quad + \varphi^{-1}\beta\left(Cs + M\dot{z}_1\right) - \varepsilon(Z),\end{aligned} \tag{23}$$

where $Z = [q, \dot{q}, s, \dot{\alpha}_1]$ are the input variables and $\varepsilon(Z) \in \mathbb{R}^n$ is the approximation error.

We propose the RBFNN adaptive control as

$$\begin{aligned}u &= -\varphi^{-1}\beta\left(\frac{s}{\|s\|^2}z_1^T z_2 + \eta|s|\operatorname{sign}(s)\right)\odot\left|z_2\right|^{1-\varphi}\\ &\quad -\widehat{W}^T H(Z)\odot\left|z_2\right|^{1-\varphi} + \widehat{d}.\end{aligned} \tag{24}$$

The weight adaptation law is designed as

$$\dot{\widehat{W}}_i = \Gamma_{2i}\left(s_i^T\frac{1}{\beta_i}\varphi H_i(Z) - \delta_{2i}\widehat{W}_i\right), \tag{25}$$

where Γ_2 is the constant matrix and δ_2 is a positive constant.

Theorem 1. *Considering the robotic system described by (1) with bounded initial conditions, under the control law (24), the NN adaption law (25), and the disturbance observer adaption law (19), if Assumption 1 holds, we can obtain the semiglobal uniform boundedness stability of the closed-loop system. In addition, the closed-loop error signals z_1, s, $\widetilde{d}$, and $\widetilde{W}$ will remain within the compact sets Ω_{z_1}, Ω_s, $\Omega_{\widetilde{d}}$, and $\Omega_{\widetilde{W}}$, respectively, defined by*

$$\begin{aligned}\Omega_{z_1} &= \left\{z_1 \in \mathbb{R}^n \mid \left\|z_1\right\| \le \sqrt{D}\right\},\\ \Omega_s &= \left\{s \in \mathbb{R}^n \mid \|s\| \le \sqrt{\frac{D}{\lambda_{\min}(M)}}\right\},\\ \Omega_{\widetilde{d}} &= \left\{\widetilde{d} \in \mathbb{R}^n \mid \|\widetilde{d}\| \le \sqrt{\frac{D}{\lambda_{\min}\left(\Gamma_{1i}^{-1}\right)}}\right\},\\ \Omega_{\widetilde{w}} &= \left\{\widetilde{W} \in \mathbb{R}^n \mid \|\widetilde{W}\| \le \sqrt{\frac{D}{\lambda_{\min}\left(\Gamma_{2i}^{-1}\right)}}\right\},\end{aligned} \tag{26}$$

where $D = 2(V(0) + (c/\rho))$ and ρ and c are two positive constants.

Proof. Consider the overall Lyapunov function candidate

$$V = \frac{1}{2}z_1^T z_1 + \frac{1}{2}s^T M s + \frac{1}{2}\tilde{d}^T \Gamma_1^{-1}\tilde{d} + \frac{1}{2}\sum_{i=1}^{n}\tilde{W}_i^T \Gamma_{2i}^{-1}\tilde{W}_i. \quad (27)$$

Differentiating (27) and substituting control law (24) into it yield

$$\dot{V} \le - z_1^T K_1 z_1 - \eta\|s\|^2 - \frac{1}{2}\sum_{i=1}^{n}\left(\delta_{1i} - \Gamma_{1i}^{-1}\right)\tilde{d}_i^2 + \frac{1}{2}\sum_{i=1}^{n}\Gamma_{1i}^{-1}\overline{d'_i}^2 + \frac{1}{2}\sum_{i=1}^{n}\delta_{1i}\bar{d}_i^2 + s^T\varphi\frac{1}{\beta}\left(\varepsilon - \tilde{W}^T H(Z)\right) + \sum_{i=1}^{n}\tilde{W}_i^T\Gamma_{2i}^{-1}\dot{\hat{W}}_i. \quad (28)$$

Substituting weight adaptation law (25) into (28) and simplifying it yield

$$\dot{V} \le - z_1^T K_1 z_1 - s^T\left(\eta I - \varphi\frac{1}{2\beta}\right)s - \frac{1}{2}\sum_{i=1}^{n}\left(\delta_{1i} - \Gamma_{1i}^{-1}\right)\tilde{d}_i^2 - \frac{1}{2}\sum_{i=1}^{n}\tilde{W}_i^T\delta_{2i}\tilde{W}_i + \frac{1}{2}\sum_{i=1}^{n}\Gamma_{1i}^{-1}\bar{d}_i^2 + \frac{1}{2}\sum_{i=1}^{n}\delta_{1i}\bar{d}_i^2 + \varphi\frac{1}{2\beta}\bar{\varepsilon}^2 + \frac{1}{2}\sum_{i=1}^{n}W_i^{*T}\delta_{2i}W_i^* \le -\rho V + c, \quad (29)$$

where

$$\rho = \min\left(2\lambda_{\min}(K_1), \frac{2\lambda_{\min}(\eta I - (1/2\beta)\varphi)}{\lambda_{\max}(M)}, \min_{i=1,2,\ldots,n}\frac{\delta_{1i} - \Gamma_{1i}^{-1}}{\lambda_{\max}(\Gamma_1^{-1})}, \min_{i=1,2,\ldots,n}\frac{\delta_{2i}}{\lambda_{\max}(\Gamma_2^{-1})}\right),$$
$$c = \frac{1}{2}\sum_{i=1}^{n}\Gamma_{1i}^{-1}\overline{d'_i}^2 + \frac{1}{2}\sum_{i=1}^{n}\delta_{1i}\bar{d}_i^2 + \frac{1}{2\beta}\varphi\bar{\varepsilon}^2 + \frac{1}{2}\sum_{i=1}^{n}W_i^{*T}\delta_{2i}W_i^*. \quad (30)$$

To guarantee $\rho > 0$, the control gains η, Γ_{1i}, and δ_{1i} are selected to satisfy

$$\lambda_{\min}\left(\eta I - \left(\frac{1}{2\beta}\right)\varphi\right) > 0,$$
$$\min_{i=1,2,\ldots,n}\left(\delta_{1i} - \Gamma_{1i}^{-1}\right) > 0. \quad (31)$$

From the above analysis, we know that the signals z_1 and s, as well as the approximation errors $\tilde{d}$ and $\tilde{W}$ are bounded.

Multiplying both sides by $e^{\rho t}$ in (29) and integrating it, we obtain

$$V(t) \le \left(V(0) - \frac{c}{\rho}\right)e^{-\rho t} + \frac{c}{\rho}$$
$$\le V(0) + \frac{c}{\rho}. \quad (32)$$

Then, the following inequalities hold

$$\frac{1}{2}z_1^T z_1 \le V(0) + \frac{c}{\rho}. \quad (33)$$

Hence, z_1 converges to the compact set. Bounds for s, $\tilde{d}$ and $\tilde{W}$ can be similarly shown, and this concludes the proof. □

3.2. Global Controller Design Based on Weighted Multiple-Model Approach. Great progress has been made in solving uncertainties by NN, yet it cannot be handled well when the parameters jump largely. WMNNAC, as a method combining conventional adaptive control and prior knowledge, can improve the trajectory tracking performance of robotic with largely jumping parameters. The design process includes the construction of the model set and controller set, as well as the weighting algorithm.

Several local models with fixed parameter are established to form a model set, which is expressed as $\Lambda = \{M_\xi \mid \xi = 1, 2, \ldots, N\}$, where M_ξ is the ξth local model.

The design process of the local controller has been given based on Lyapunov direct method. It is capable of achieving good trajectory tracking of the robotic with uncertainties and disturbances. Corresponding to each local model, we design a local NN-based controller by this method. Thus, we can establish the controller set $\Psi = \left\{C_\xi \mid \xi = 1, 2, \ldots, N\right\}$.

Based on [42], a new weighting update algorithm is proposed for the structure of the robotic manipulators. It is employed to coordinate local NN-based controllers to generate the global control signals in real time. The key to the new algorithm is to define a performance index based on the model output error.

For clarity, the algorithm is expressed in the discrete-time form, and a zero-order holder can be adopted to obtain continuous signals.

Since this paper studies the n-link robotic manipulators, we should consider that single error has the approximately same influence on model output errors, and we define the model output errors as

$$e_\xi = q_\xi - q = \left[e_{\xi 1}, e_{\xi 2}, \ldots, e_{\xi n}\right],$$
$$\xi = 1, 2, \ldots, N, \quad (34)$$
$$e_\xi^2(k) = \theta_1 e_{\xi 1}^2(k) + \theta_2 e_{\xi 2}^2(k) + \cdots + \theta_n e_{\xi n}^2(k),$$

where $\theta_i > 0$ is the error weight of the ith link of robotic in the model set.

The performance index is designed as

$$l'_\xi(k) = \alpha + e_\xi^2(k), \tag{35}$$

where α is a small value.

Then, we give the following weighting algorithm:

$$l'_{\min}(k) = \min l'_\xi(k), \tag{36}$$

$$g(k) = \frac{l'_{\min}(k)}{l'_\xi(k)}, \tag{37}$$

$$l_\xi(k) = \begin{cases} l_\xi(k-1), & g(k) = 1, \\ l_\xi(k-1)g(k)^{\mathrm{ceil}(1/(1-g(k)))}, & g(k) < 1, \end{cases} \tag{38}$$

$$p_\xi(k) = \frac{l_\xi(k)}{\sum_{\xi=1}^N l_\xi(k)}. \tag{39}$$

From above all, the global control law is defined as

$$u_c(k) = \sum_{\xi=1}^N p_\xi(k)u_\xi(k), \tag{40}$$

where the value of $p_\xi(k)$ lies between 0 and 1, and $\sum_{\xi=1}^N p_\xi(k) = 1, \forall k \geq 0$.

3.3. Convergence Analysis of the WMNNAC System.

3.3. Convergence Analysis of the WMNNAC System. We have the following convergence result for the WMNNAC system.

Theorem 2. *Respecting the WMNNAC system structured in Figure 1, it owns the following properties:*

(1) *The time interval of parameter jump in the plant model is long enough (for details, see description in section Simulation)*

(2) *Various stages of the plant model can be approximated by model set Λ*

(3) *Each local NN-based controller is well designed such that the closed-loop system is stable, and the output of system is tracking the reference signal*

(4) *$M_\xi \in \Lambda$ is the model closest to the current plant in the following sense with probability one for $\forall k \geq \overline{k}$*

$$\begin{cases} \sum_{r=1}^{T_j+k} e_\mu^2(r) < \sum_{r=1}^{T_j+k} e_\xi^2(r), & \forall k \geq \overline{k}, \mu \neq \xi, \\ \lim_{k\longrightarrow\infty} \frac{1}{k} \sum_{r=1}^{T_j+k} e_\mu^2(r) = R_\mu; \\ \lim_{k\longrightarrow\infty} \frac{1}{k} \sum_{r=1}^{T_j+k} e_\xi^2(r) = R_\xi, & R_\mu < R_\xi, \mu \neq \xi, \end{cases} \tag{41}$$

where $\overline{k}$ is an unknown finite time instant, R_μ is a constant value, R_ξ may be constant value or infinity, and T_j, $j = 0, 1, 2, \ldots$ is the time sequence for jumping parameters.

Then, the global controller converges to the most appropriate local controller at each stage of parameter jumps, and the WMNNAC system is stable.

Proof. Firstly, it is not difficult to see that algorithms (36)–(39) together with properties (2) and (4) in Theorem 2 guarantee with probability one that

$$\begin{cases} l'_{\min}(k) = l'_\xi(k), \\ \dfrac{l'_{\min}(k)}{l'_\xi(k)} = 1, \\ \dfrac{l'_{\min}(k)}{l'_\mu(k)} < 1, \\ \forall k \geq \overline{k}, \mu \neq \xi. \end{cases} \tag{42}$$

Moreover, if there is

$$\lim_{k\longrightarrow\infty} g(k) = \lim_{k\longrightarrow\infty} \frac{l'_{\min}(k)}{l'_\xi(k)} = 1. \tag{43}$$

Then, from (38), we have

$$\lim_{k\longrightarrow\infty} [g(k)]^{\mathrm{ceil}(1/(1-g(k)))} = \frac{1}{e} < 1. \tag{44}$$

It can always guarantee that

$$\begin{gathered} l_\xi(k) \longrightarrow l_\xi(0); \\ l_\mu(k) \longrightarrow l_\mu(0), \\ \xi \neq \mu. \end{gathered} \tag{45}$$

Thus, from (39), we obtain

$$\begin{gathered} \lim_{k\longrightarrow\infty} p_\xi(k) = 1; \\ \lim_{k\longrightarrow\infty} p_\mu(k) = 0, \\ \xi \neq \mu. \end{gathered} \tag{46}$$

That means the weighting algorithm is convergent.

Further, we know that the global controller of the WMNNAC system will converge to the most appropriate local controller corresponding to the local model $M_\xi \in \Lambda$. Besides, the NN-based local controllers are established in the same way. Then, the WMNNAC system can be described as VES in the input-output sense, as shown in Figure 2.

$M_\xi \in \Lambda$ is the model closest to the true plant, C_ξ is the corresponding NN-based controller, $\Delta u(k)$ is the equivalent control error, $\Delta u(k) = u(k) - u_\xi(k)$, and $e'(k)$ is the equivalent output error, $e'(k) = y(k) - y_\xi(k)$.

For the NN-based controller, we have rigorously proved the stability of the local closed-loop system based on Lyapunov theory, and the influence of other fixed controllers is included in $\Delta u(k)$. According to property (3), the stability can be obtained for a short time period.

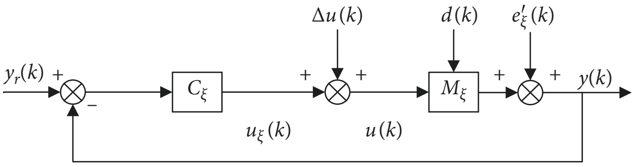


Figure 2: VES of the WMNNAC system.

Finally, since the jumps cannot be infinitely fast, the WMNNAC system is composed of "slow" switching between these stable local systems. By switching system theory [43], the WMNNAC system of the robotic manipulator with jumping parameters is stable. □

4. Simulation

In this section, considering a two-link robotic manipulator shown in Figure 3, simulations are carried out to verify the performance of the controller designed in Section 3. It is assumed to move on the Cartesian space and then the position vector q can be rewritten as

$$q = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}. \tag{47}$$

For kinematics model (1), the parameters can be written as

$$M(q) = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & m_2 l_{c2}^2 + I_2 \end{bmatrix}, \tag{48}$$

$$C(q,\dot{q}) = \begin{bmatrix} -m_2 l_1 l_{c2} \dot{q}_2 \sin q_2 & C_{12} \\ m_2 l_1 l_{c2} \dot{q}_1 \sin q_2 & 0 \end{bmatrix}, \tag{49}$$

$$G(q) = \begin{bmatrix} G_1 \\ m_2 l_{c2} g \cos\left(q_1 + q_{2,}\right) \end{bmatrix}, \tag{50}$$

$$M_{11} = m_1 l_{c1}^2 + m_2\left(l_1^2 + l_{c1}^2 + 2 l_1 l_{c2} \cos q_2\right) + I_1 + I_2, \tag{51}$$

$$M_{12} = M_{21} = m_2\left(l_{c2}^2 + l_1 l_{c2} \cos q_2\right) + I_2, \tag{52}$$

$$C_{12} = -m_2 l_1 l_{c2} (\dot{q}_1 + \dot{q}_2) \sin q_2, \tag{53}$$

$$G_1 = (m_1 l_{c2} + m_2 l_1) g \cos q_1 + m_2 l_{c2} g \cos(q_1 + q_2), \tag{54}$$

$$\begin{aligned} l_1 &= 2 l_{c1}, \\ l_2 &= 2 l_{c2}. \end{aligned} \tag{55}$$

System parameter values of robotic are given as

$$\begin{aligned} m_1 &= 2\,\text{kg}, \\ m_2 &= 1\,\text{kg}, \\ l_1 &= 0.35\,\text{m}, \\ l_2 &= 0.31\,\text{m}, \\ I_1 &= 61.25 \times 10^{-3}\,\text{kgm}^2, \\ I_2 &= 20.42 \times 10^{-3}\,\text{kgm}^2. \end{aligned} \tag{56}$$

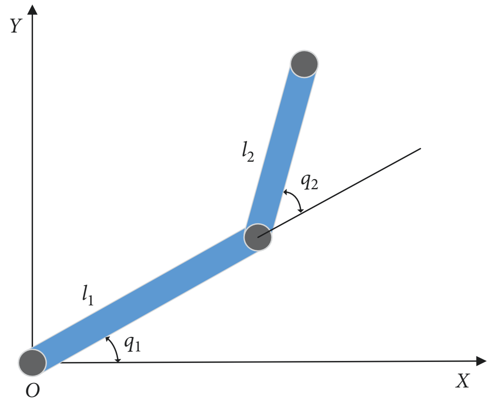


Figure 3: Schematic of the two-link robot manipulator model.

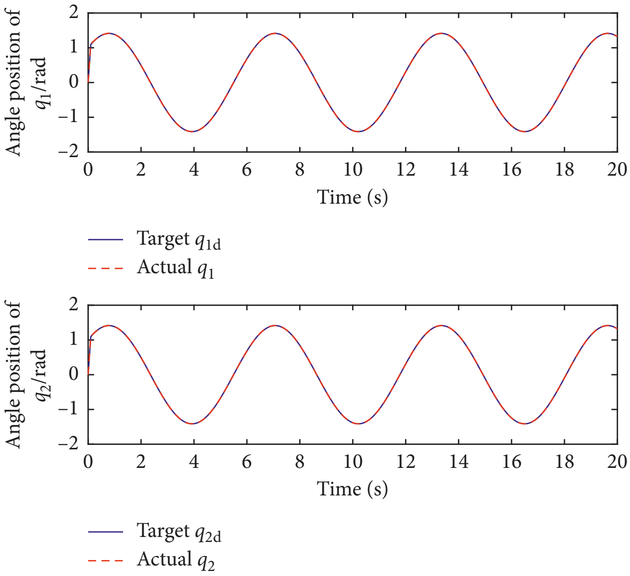


Figure 4: Tracking performance of q_1 and q_2 with the single NN-based controller.

The states of robotic are initialized at $q_1(0) = q_2(0) = 0$ and $\dot{q}_1(0) = \dot{q}_2(0) = 0$, and the original configuration of the NN weights is zero. The target trajectory tracking a circular path is given as $q_d = [q_{1d}, q_{2d}]^T = [\sin(t) + \cos(t),\ \sin(t) + \cos(t)]^T$, where $t \in [0, t_f]$ and $t_f = 20\,\text{s}$. The external disturbance is given as $d = [\sin(t) + 1, 2\ \cos(t) + 0.5]^T$.

There are three parts encompassed in the following simulations. Case 1 presents the good control effects of the proposed local NN-based controller with external disturbance. Case 2 is used for comparison to show the poor performance of the local NN-based controller when largely jumping parameters occur in the robotic system. Case 3 shows the advancement of the proposed WMNNAC method for robotic in the case of parameters jumping.

Case 1. Local NN-based controller.

For the NN-based control system, the constant parameters are chosen to build model as $K_1 = 100 I_{2\times2}$, $\eta = 28$, $1/\beta = 27 I_{2\times2}$, $\varphi = 1.182$, $\Gamma_1 = 0.5 I_{2\times2}$, and $\sigma_2 = 100 I_{2\times2}$,

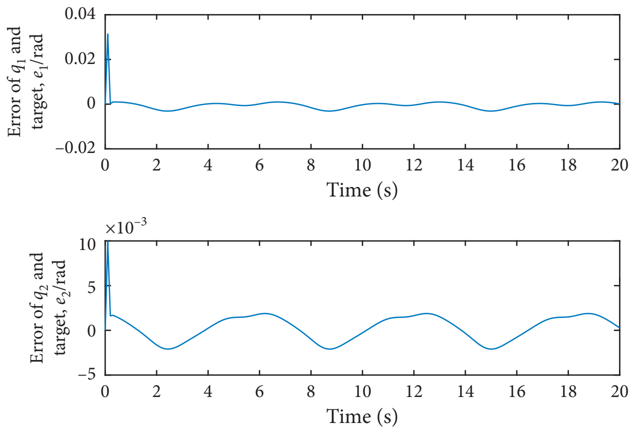


Figure 5: Tracking error under the single NN-based controller.

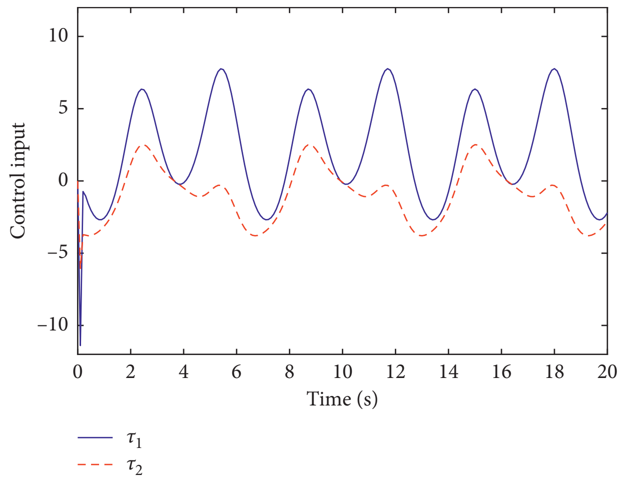


Figure 6: Single NN-based control inputs.

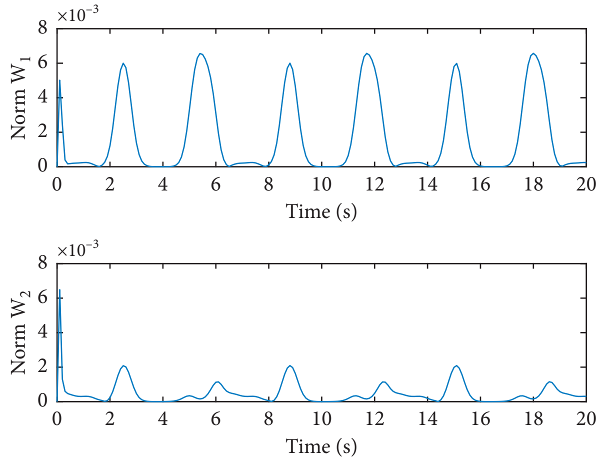


Figure 7: Estimation weights with the single NN-based controller.

which satisfy the conditions in (31). For the RBFNN, it owns 256 nodes, and its parameters are given as $\Gamma_2 = 100I_{256\times256}$ and $\sigma_2 = 0.2I_{2\times2}$. Moreover, the initial weights of RBFNN are 0.

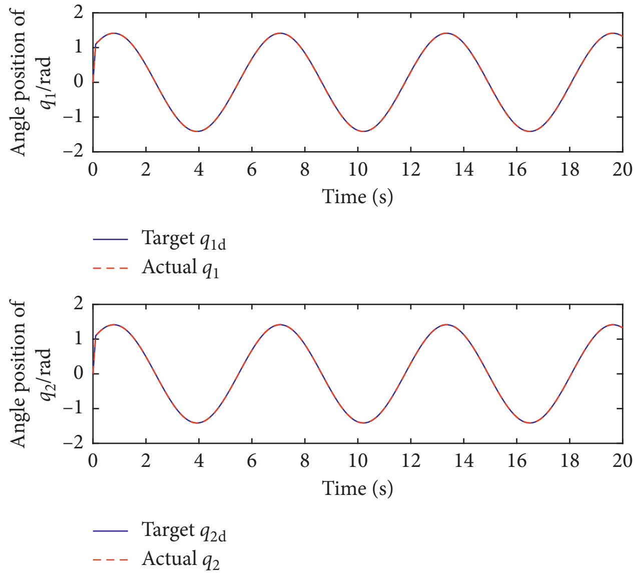


Figure 8: Tracking performance of q_1 and q_2 of the single NN-based controller with jumping parameters.

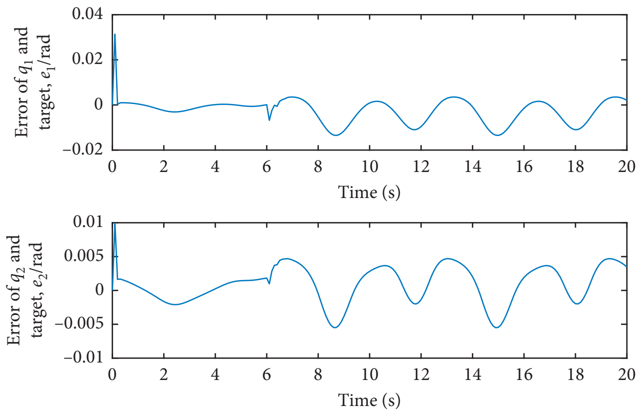


Figure 9: Tracking error under the single NN-based controller with jumping parameters.

The simulation results of the NN-based controller are shown in Figures 4–7. They can be stated that the angle positions can track desired trajectory gratifyingly, and the tracking errors are convergent and fluctuate gently around zero. The values of the controller are presented in Figure 6. From Figure 7, we can notice that the approximation weights of RBFNN remain within bound.

Case 2. Single NN-based controller for the robotic manipulators with jumping parameters.

Meanwhile, we consider that the value of m_2 and l_2 in the robotic system is mutative, and other parameters are the same as above. When the mass of link 2 is changed from 1 kg to 7 kg and the length of link 2 is changed from 0.31 m to 0.2 m at $t = 6s$, we can obtain the results shown in Figures 8 and 9. Compare to Figure 5, they show that the tracking error fluctuates more intense. Obviously, single NN-based

Table 1: The designed parameters of the model set and the controller set.

Model set	Model parameters	Corresponding controller parameters
Model 1	$m_2 = 1\,\text{kg}$, $l_2 = 0.31\,\text{kg}$	$\eta = 28$, $1/\beta = 27I_{2\times2}$
Model 2	$m_2 = 3\,\text{kg}$, $l_2 = 0.25\,\text{kg}$	$\eta = 39$, $1/\beta = 10I_{2\times2}$
Model 3	$m_2 = 7\,\text{kg}$, $l_2 = 0.20\,\text{kg}$	$\eta = 72$, $1/\beta = 15I_{2\times2}$

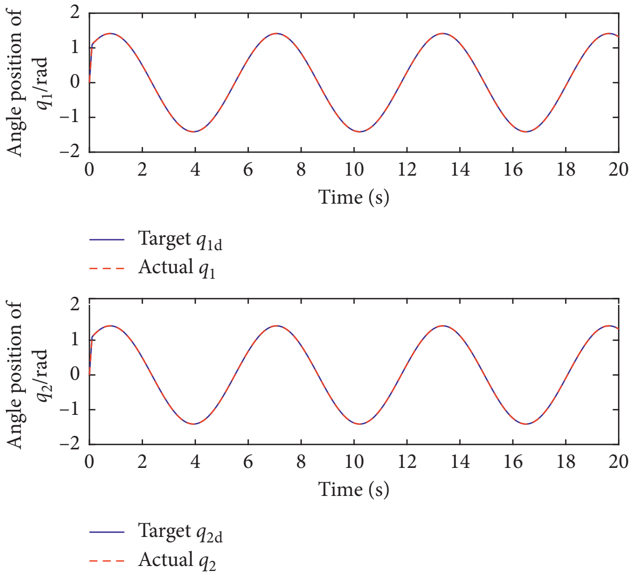


Figure 10: Tracking performance of q_1 and q_2 with WMNNAC.

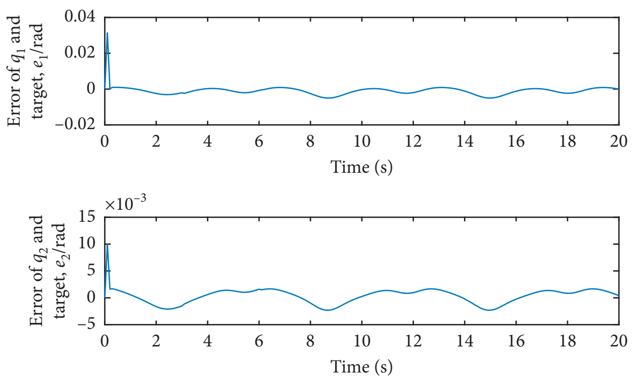


Figure 11: Tracking error under the WMNNAC.

controller cannot meet the tracking effect and even bring about system failures when model parameters jump largely.

Case 3. WMNNAC for the robotic manipulators with jumping parameters.

The implementation method of each local system is similar to Case 1. We just set suitable parameters to each local system for better performances. According to (48)–(55), the different robotic manipulator system is built by different model parameters. In this simulation, m_2 and l_2 are different in each model. Under this circumstance, the parameters of the model set and controller set are shown in Table 1. The other parameters of the three local system are the same as Case 1. It is assumed that the manipulator parameter jumps twice, respectively, from model 1 to model 2 at $t = 3\,\text{s}$ and from model 2 to model 3 at $t = 6\,\text{s}$.

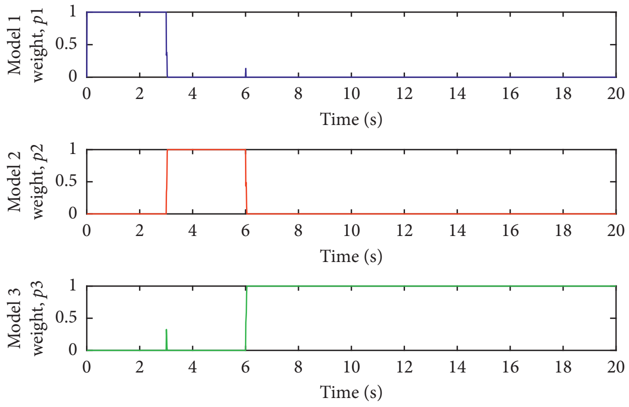


Figure 12: The weights for the local controllers of the WMNNAC system.

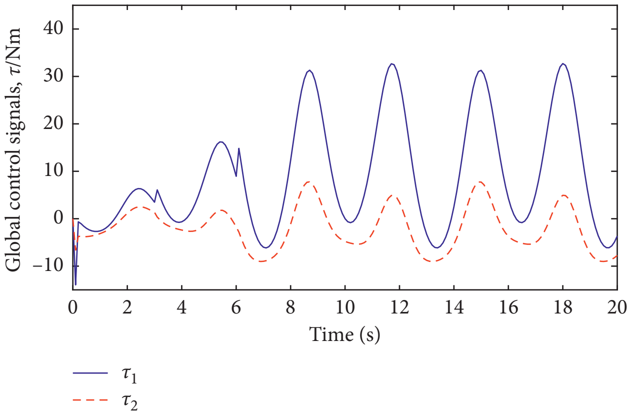


Figure 13: Global control signals for the manipulator with jumping parameters.

The simulation results are shown in Figures 10–13. Compared to Case 2, it can be seen that the angle positions of the robotic can still follow the desired trajectory well, and the errors fluctuate and remain within a small range even when the parameter jumps largely. Moreover, the weights of three local controllers are demonstrated in Figure 12. It can be seen that the weights converge well. Besides, Figure 13 shows the global control signals, and the results indicate that the system can choose suitable control inputs rapidly to adapt the abrupt changes of parameters by the weighting algorithm.

Finally, we analyze the transient performance of the robotic system by comparing the error of case 2 and case 3 when the parameters jump. By partial enlargement, the

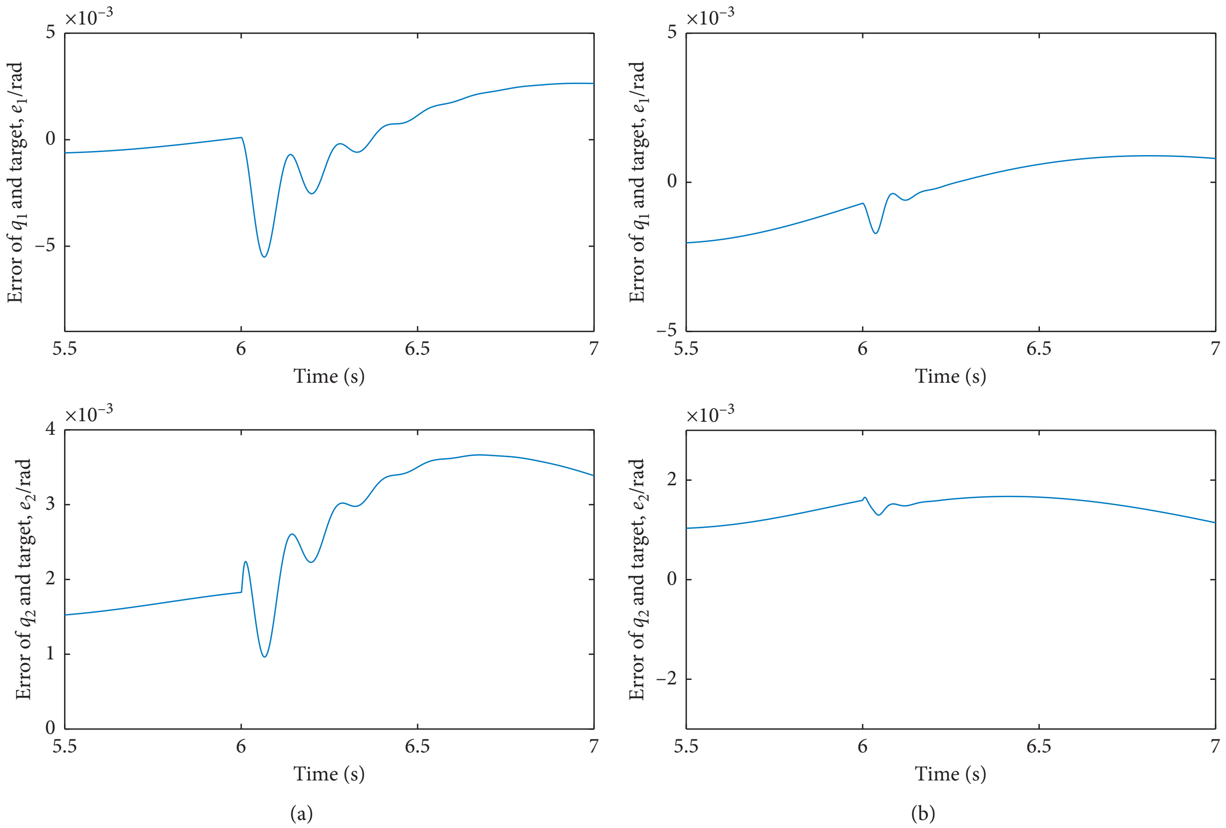


FIGURE 14: Transient performance of (a) single NN-based controller and (b) WMNNAC.

comparisons are shown in Figure 14. As we can see, the greater error fluctuations and longer adjustment times based on the single NN-based controller are shown in Figure 14(a). Encouragingly, the WMNNAC can improve the transient performance shown in Figure 14(b).

By the superiority of fusing multiple controllers, the WMNNAC method is effective to avoid instability and poor transient performance caused by largely jumping parameters.

5. Conclusion

In this paper, we developed a WMNNAC method for robotic manipulators with jumping parameters. The local NN-based controller has been clearly deduced and has good performance in compensating for the influence of uncertainties and external disturbance. By combining multiple NN-based controllers and the WMMAC method based on the improved weighting algorithm, the robotic manipulators can track the desired trajectory well when parameters jump largely. Moreover, global stability analysis and proof have been given based on VES theory. Finally, the simulation results have demonstrated the efficiency of the developed control approach. In the future, research will focus on methods for establishing the model set and the practical application.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Sciences Foundation of China (nos. 61520106010 and 61741302).

References

[1] Z. Yang, Y. Fukushima, and P. Qin, "Decentralized adaptive robust control of robot manipulators using disturbance observers," *IEEE Transactions on Control Systems Technology*, vol. 20, no. 5, pp. 1357–1365, 2012.

[2] W. He, W. Ge, Y. Li, Y.-J. Liu, C. Yang, and C. Sun, "Model identification and control design for a humanoid robot," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 1, pp. 45–57, 2017.

[3] O. Barambones and V. Etxebarria, "Robust neural control for robotic manipulators," *Automatica*, vol. 38, no. 2, pp. 235–242, 2002.

[4] J. Na, M. N. Mahyuddin, G. Herrmann, X. Ren, and P. Barber, "Robust adaptive finite-time parameter estimation and control for robotic systems," *International Journal of Robust and Nonlinear Control*, vol. 25, no. 16, pp. 3045–3071, 2014.

[5] J. Baek, M. Jin, and S. Han, "A new adaptive sliding-mode control scheme for application to robot manipulators," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 6, pp. 3628–3637, 2016.

[6] H. Wang, "Adaptive control of robot manipulators with uncertain kinematics and dynamics," *IEEE Transactions on Automatic Control*, vol. 62, no. 2, pp. 948–954, 2017.

[7] S. Mondal and C. Mahanta, "Adaptive second order terminal sliding mode controller for robotic manipulators," *Journal of the Franklin Institute*, vol. 351, no. 4, pp. 2356–2377, 2014.

[8] H. Li, S. Zhao, W. He, and R. Lu, "Adaptive finite-time tracking control of full state constrained nonlinear systems with dead-zone," *Automatica*, vol. 100, pp. 99–107, 2019.

[9] W. He and S. S. Ge, "Cooperative control of a nonuniform gantry crane with constrained tension," *Automatica*, vol. 66, pp. 146–154, 2016.

[10] Q. Zhou, L. Wang, C. Wu, H. Li, and H. Du, "Adaptive fuzzy control for nonstrict-feedback systems with input saturation and output constraint," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 1, pp. 1–12, 2017.

[11] G. Tao, S. Chen, and S. M. Joshi, "An adaptive control scheme for systems with unknown actuator failures," *Automatica*, vol. 38, no. 6, pp. 1027–1034, 2002.

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

[13] D. Wang, D. Liu, H. Li, B. Luo, and H. Ma, "An approximate optimal control approach for robust stabilization of a class of discrete-time nonlinear systems with uncertainties," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 5, pp. 713–717, 2016.

[14] H. Yang and J. Liu, "An adaptive RBF neural network control method for a class of nonlinear systems," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 2, pp. 457–462, 2018.

[15] X. Su, Y. Wen, Y.-D. Song, and T. Hayat, "Dissipativity-based fuzzy control of nonlinear systems via an event-triggered mechanism," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 6, pp. 1208–1217, 2019.

[16] M. Chen, S. S. Ge, and B. Ren, "Adaptive tracking control of uncertain MIMO nonlinear systems with input constraints," *Automatica*, vol. 47, no. 3, pp. 452–465, 2011.

[17] X. Su, P. Shi, L. Wu, and M. V. Basin, "Reliable filtering with strict dissipativity for T-S fuzzy time-delay systems," *IEEE Transactions on Cybernetics*, vol. 44, no. 12, pp. 2470–2483, 2014.

[18] S. Maiti, V. Verma, C. Chakraborty, and Y. Hori, "An adaptive speed sensorless induction motor drive with artificial neural network for stability enhancement," *IEEE Transactions on Industrial Informatics*, vol. 8, no. 4, pp. 757–766, 2012.

[19] Y.-J. Liu, S. Tong, C. L. P. Chen, and D.-J. Li, "Neural controller design-based adaptive control for nonlinear MIMO systems with unknown hysteresis inputs," *IEEE Transactions on Cybernetics*, vol. 46, no. 1, pp. 9–19, 2016.

[20] H. Gao, Y. Song, and C. Wen, "Backstepping design of adaptive neural fault-tolerant control for MIMO nonlinear systems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 11, pp. 2605–2613, 2017.

[21] K. Zhao, Y. Song, T. Ma, and L. He, "Prescribed performance control of uncertain Euler¨CLagrange systems subject to full-state constraints," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 8, pp. 3478–3489, 2018.

[22] K. Zhao and Y. Song, "Neuroadaptive robotic control under time-varying asymmetric motion constraints: a feasibility-condition-free approach," *IEEE Transactions on Cybernetics*, vol. 50, no. 1, pp. 15–24, 2020.

[23] Y. Zhang, Q. Li, W. Zhang, Y. Liu, and Z. Xue, "Weighted multiple neural network boundary control for a flexible manipulator with uncertain parameters," *IEEE Access*, vol. 7, pp. 57633–57641, 2019.

[24] L. Chen and K. S. Narendra, "Nonlinear adaptive control using neural networks and multiple models," *Automatica*, vol. 37, no. 8, pp. 1245–1255, 2001.

[25] Y. Fu, T. Chai, and H. Yue, "Intelligent control using multiple models and neural networks," *International Journal of Adaptive Control and Signal Processing*, vol. 22, no. 9, pp. 495–509, 2008.

[26] K. S. Narendra and X. Cheng Xiang, "Adaptive control of discrete-time systems using multiple models," *IEEE Transactions on Automatic Control*, vol. 45, no. 9, pp. 1669–1686, 2000.

[27] K. S. Narendra and J. Balakrishnan, "Adaptive control using multiple models," *IEEE Transactions on Automatic Control*, vol. 42, no. 2, pp. 171–187, 1997.

[28] N. Ruishan, W. Xia, Z. Jun, and C. Chao, "Switched adaptive control for a class of robot manipulators," in *Proceedings of the 31st Chinese Control Conference*, pp. 4974–4978, Hefei, China, July 2012.

[29] L. Yu, S. Fei, L. Sun, and J. Huang, "An adaptive neural network switching control approach of robotic manipulators for trajectory tracking," *International Journal of Computer Mathematics*, vol. 91, no. 5, pp. 983–995, 2014.

[30] X. Wang and J. Zhao, "Switched adaptive tracking control of robot manipulators with friction and changing loads," *International Journal of Systems Science*, vol. 46, no. 6, pp. 955–965, 2013.

[31] C. Yu, R. J. Roy, H. Kaufman, and B. W. Bequette, "Multiple-model adaptive predictive control of mean arterial pressure and cardiac output," *IEEE Transactions on Biomedical Engineering*, vol. 39, no. 8, pp. 765–778, 1992.

[32] S. Fekri, M. Athans, and A. Pascoal, "Robust multiple model adaptive control (RMMAC): a case study," *International Journal of Adaptive Control and Signal Processing*, vol. 21, no. 1, pp. 1–30, 2007.

[33] M. Kuipers and P. Ioannou, "Multiple model adaptive control with mixing," *IEEE Transactions on Automatic Control*, vol. 55, no. 8, pp. 1822–1836, 2010.

[34] W. Zhang, "Stable weighted multiple model adaptive control with improved convergence rate," *IFAC Proceedings Volumes*, vol. 45, no. 13, pp. 570–575, 2012.

[35] S. Baldi, P. Ioannou, and E. Mosca, "Multiple model adaptive mixing control: the discrete-time case," *IEEE Transactions on Automatic Control*, vol. 57, no. 4, pp. 1040–1045, 2012.

[36] W. Zhang, "Further results on stable weighted multiple model adaptive control: discrete-time stochastic plant," *International Journal of Adaptive Control and Signal Processing*, vol. 29, no. 12, pp. 1497–1514, 2015.

[37] W. Zhang, "Stable weighted multiple model adaptive control: discrete-time stochastic plant," *International Journal of Adaptive Control and Signal Processing*, vol. 27, no. 7, pp. 562–581, 2013.

[38] W. Zhang and J. Li, "Stable weighted multiple model adaptive control of continuous-time plant with large parameter uncertainties," *IEEE Access*, vol. 7, pp. 144125–144133, 2019.

[39] L. Bascetta and P. Rocco, "Revising the robust-control design for rigid robot manipulators," *IEEE Transactions on Robotics*, vol. 26, no. 1, pp. 180–187, 2010.

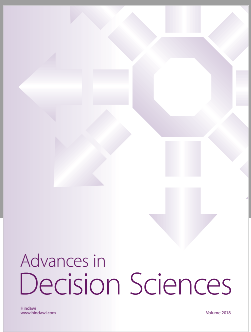
[40] W. He, H. Huang, and S. S. Ge, "Adaptive neural network control of a robotic manipulator with time-varying output constraints," *IEEE Transactions on Cybernetics*, vol. 47, no. 10, pp. 3136–3147, 2017.

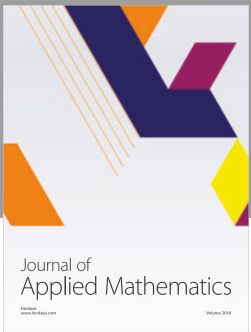
[41] S. S. Ge and C. Wang, "Adaptive neural control of uncertain MIMO nonlinear systems," *IEEE Transactions on Neural Networks*, vol. 15, no. 3, pp. 674–692, 2004.

[42] W. Zhang, "Weighted multiple model adaptive control of discrete-time stochastic system with uncertain parameters," *Acta Automat Sinica*, vol. 41, no. 3, pp. 541–550, 2015.

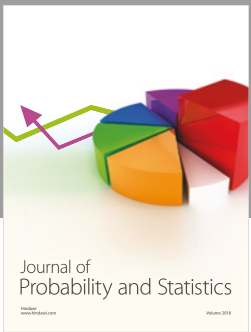
[43] R. Shorten, F. Wirth, O. Mason, K. Wulff, and C. King, "Stability criteria for switched and hybrid systems," *SIAM Review*, vol. 49, no. 4, pp. 545–592, 2007.

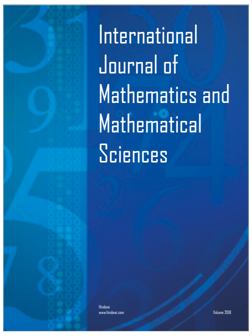
Advances in
Operations Research


Advances in
Decision Sciences

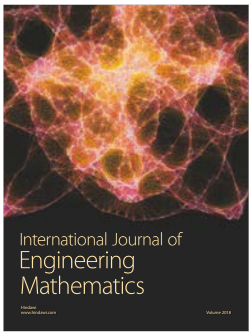

Journal of
Applied Mathematics

The Scientific
World Journal

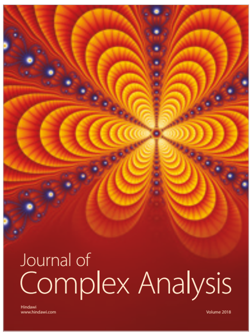

Journal of
Probability and Statistics

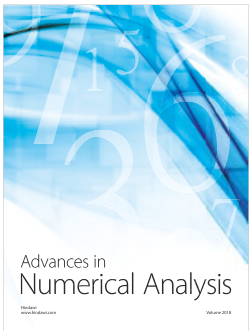

International
Journal of
Mathematics and
Mathematical
Sciences

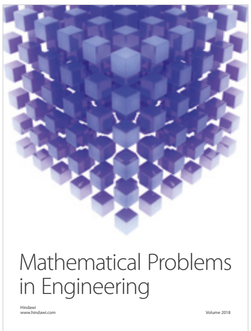
Journal of
Optimization

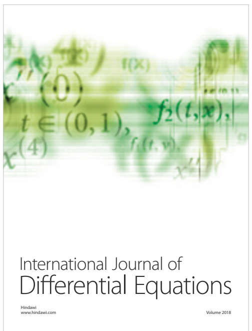

International Journal of
Engineering
Mathematics

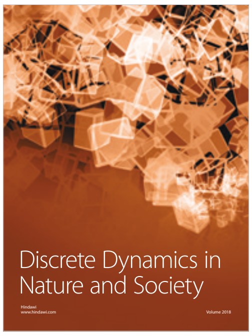
International Journal of
Analysis

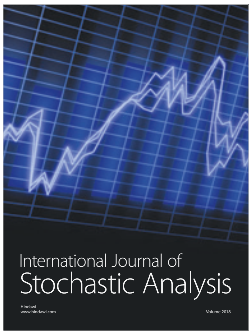

Journal of
Complex Analysis


Advances in
Numerical Analysis


Mathematical Problems
in Engineering

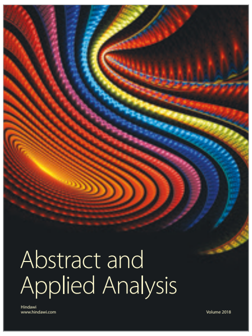

International Journal of
Differential Equations

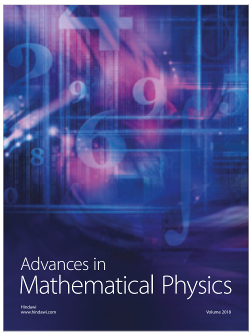

Discrete Dynamics in
Nature and Society


International Journal of
Stochastic Analysis

Journal of
Mathematics

Journal of
Function Spaces


Abstract and
Applied Analysis


Advances in
Mathematical Physics