Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 9996719, 12 pages
https://doi.org/10.1155/2021/9996719

Research Article

Fractional-Order Iterative Sliding Mode Control Based on the Neural Network for Manipulator

Xin Zhang,[1,2] **Wenbo Xu**,[1] **and Wenru Lu**[1]

[1]*School of Automation & Electrical Engineering, Lanzhou Jiaotong University, Lanzhou 730070, China*
[2]*Gansu Provincial Engineering Research Center for Artificial Intelligence and Graphics & Image Processing, Lanzhou Jiaotong University, Lanzhou 730070, China*

Correspondence should be addressed to Wenbo Xu; 18326990418@163.com

Received 13 March 2021; Accepted 21 July 2021; Published 2 August 2021

Academic Editor: Adnan Maqsood

This study aimed to improve the position tracking accuracy of the single joint of the manipulator when the manipulator model information is uncertain. The study is based on the theory of fractional calculus, radial basis function (RBF) neural network control, and iterative sliding mode control, and the RBF neural network fractional-order iterative sliding mode control strategy is proposed. First, the stability analysis of the proposed control strategy is carried out through the Lyapunov function. Second, taking the two-joint manipulator as an example, simulation comparison and analysis are carried out with iterative sliding mode control strategy, fractional-order iterative sliding mode reaching law control strategy, and fractional-order iterative sliding mode surface control strategy. Finally, through simulation experiments, the results show that the RBF neural network fractional-order iterative sliding mode control strategy can effectively improve the joints' tracking and control accuracy, reduce the position tracking error, and effectively suppress the chattering caused by the sliding mode control. It is proved that the proposed control strategy can ensure high-precision position tracking when the information of the manipulator model is uncertain.

1. Introduction

Since the middle of the twentieth century, industrial manipulators have developed rapidly. So far, it has become indispensable and essential equipment in the fields of machinery manufacturing [1], aerospace [2], and transportation [3]. With more and more comprehensive application range of the manipulator, the control precision, speed, and stability of the manipulator system are demanding. The manipulator system is strongly coupled, uncertain, and time-varying, and its goal is to control the end of the manipulator to its desired position accurately and stably. However, due to the external uncertainties in the working environment, changes in load, and interference from joint friction, it will increase the control difficulty of the manipulator [4]. Therefore, research on high-precision trajectory tracking control that does not rely on precise mathematical models is significant for manipulator control systems.

In recent years, many scholars have performed a lot of research on the control of the manipulator. At present, there are many control strategies for the manipulator, such as sliding mode control [5, 6], adaptive control [7], fuzzy control [8, 9], PID control [10], neural network control [11, 12], and decoupling control [13]. Sliding mode control is widely used in manipulator control systems because of its strong robustness. However, sliding mode control will frequently switch the control state of the system, resulting in a chattering phenomenon [14–16]. Therefore, it is usually combined with other control methods to suppress the chattering phenomenon and improve control accuracy. Through research, it is found that the combination of sliding mode control and iterative learning control can enhance the robustness of the system, reduce the chattering of the system, and improve the control accuracy of the manipulator [17, 18]. Lu et al. [19] proposed a new iterative learning control strategy combining sliding mode control and applied

it to hypersonic vehicle attitude control. It solves the problem of unmatched uncertain control caused by strong coupling and interference during the reentry of the aircraft. It improves the control accuracy and robustness of the system. Iman et al. [20] proposed a new fractional-order iterative learning control strategy based on the sliding mode for nonlinear manipulator systems with uncertainties. The sliding mode part improves the influence of nonlinear uncertainties. The iterative learning part eliminates the same frequent disturbances, thus improving the system's response speed and control precision. Because fractional calculus has the characteristics of memory and heritability, it is widely used in automatic control, electrical engineering, signal processing, and other fields [21]. Because of excess torque disturbance in the electric loading system, a compound control strategy of position and torque closed-loop as fractional-order PID feedback control and fractional-order iterative learning control as compensation control was proposed in [22]. The information memory characteristics of fractional-order calculus were used to improve the dynamic performance and robustness of the control system. Zhang et al. [23] combined fractional-order calculus, iterative learning control, and sliding mode control to put forward the fractional-order iterative sliding mode control strategy, which was applied to the robotic arm system to improve the control performance.

In recent years, with the rapid development of technology and economy, neural networks have been extensively developed and applied in perceptual learning, pattern recognition, system control, signal processing, and modeling technology [24]. Because the neural network has a high degree of parallel structure, strong learning ability, and continuous nonlinear function approximation ability, the condition of neural network control is less [25], and the neural network is widely used in management. Vu et al. [26] proposed a robust adaptive sliding mode neural network control strategy that combined sliding mode technology, RBF neural network approximation, and adaptive technology. The control precision and robustness of industrial manipulators in an uncertain dynamic environment are improved. Zhang et al. [27] aimed at the problem of significant synchronization errors in the vacuum pump system controlled by multimotor speed synchronization. In the relative coupling control strategy, neural network PID was introduced as the system's speed compensator, which improved the synchronization and antiinterference of multimotors.

The above studies all involve tracking control accuracy, but the abovementioned literature has not solved the problems of external interference and the uncertainty of the manipulator model establishment error, the low accuracy of trajectory tracking control, and the effects of chattering caused by sliding mode control. Inspired by the abovementioned literature, the study takes the double-joint manipulator as the research object and studies the fractional-order iterative sliding mode controller based on the RBF neural network, which effectively solves the problems in the literature mentioned above. The main contributions of this study are summarized as follows:

(1) RBF neural network is used to estimate the model uncertainty information of the double-joint manipulator, which further improves the tracking control accuracy and robustness of the system

(2) The fractional sliding mode surface and the fractional reaching law are designed to reduce the chattering phenomenon of the system

(3) Considering the uncertainties such as external disturbance and modeling error, the online update algorithm of neural network weight is designed using Lyapunov stability theory

The structure of the study is as follows. The second part introduces the modeling information of the double-joint manipulator, the design of the control law and the adaptive law, and the stability analysis of the controller. The third part provides the simulation experiment and the result analysis. Finally, the fourth part summarizes the study.

2. Manipulator Joint Tracking Control Strategy

According to Lagrange's theorem, the model of the n-joint manipulator is as follows:

$$\mathbf{M}(\mathbf{q})\ddot{q} + \mathbf{C}(\mathbf{q}, \dot{q})\dot{q} + \mathbf{G}(\mathbf{q}) + \mathbf{F}(\mathbf{t}) = \mathbf{u}, \tag{1}$$

where $q \in R^n$ is the joint position of the manipulator, $\dot{q} \in \mathbf{R}^n$ and $\ddot{q} \in R^n$ are the velocity vector and acceleration vector of the manipulator, $M(q) \in R^{n\times n}$ is the inertia matrix of the manipulator, $C(q, \dot{q}) \in R^{n\times n}$ is the centrifugal force and the Coriolis matrix, and $G(q) \in R^n$ is the gravity term. $u \in R^n$ is the control torque. System modeling errors, parameter changes, and other factors are considered external disturbances, which can be denoted by $\mathbf{F}(t)$.

2.1. Neural Network Iterative Sliding Surface. Sliding mode control consists of two parts: reaching motion and sliding mode motion. Selecting a reasonable reaching law can accelerate the reaching speed and reduce the chattering of the system to a certain extent so that a better dynamic quality can be obtained.

According to the dynamic model of the manipulator, the expected position error of each joint of the manipulator is

$$\mathbf{e}(t) = \mathbf{q}_d(t) - \mathbf{q}(t), \tag{2}$$

where $\mathbf{q}_d(t)$ is the expected position of the joint, and $\mathbf{q}(t)$ is the actual position of the joint.

The sliding mode surface function is designed as

$$\mathbf{s} = \mathbf{ce} + \dot{e}. \tag{3}$$

Derivation of (3) can be obtained:

$$\dot{s} = \mathbf{c}\dot{e} + \ddot{e}. \tag{4}$$

Incorporate equations (1) and (2) into equation (4) to obtain

$$\dot{s} = \mathbf{c}\dot{e} + \ddot{q}_d - \mathbf{M}^{-1}(\mathbf{u} - \mathbf{C}\dot{q} - \mathbf{G} - \mathbf{F}). \tag{5}$$

The exponential reaching law is taken as

$$\dot{s} = -\boldsymbol{\lambda}\text{sign}(\mathbf{s}) - k\mathbf{s}. \tag{6}$$

From formula (4), the following equation can be obtained:

$$\dot{q} = -\dot{s} + \mathbf{c}\mathbf{e} + \dot{q}_d. \tag{7}$$

From formula (7), the following equation can be obtained:

$$\begin{aligned}\mathbf{M}\dot{s} &= M\left(\ddot{q}_d + \mathbf{c}\dot{e}\right) + \mathbf{C}\mathbf{q} + \mathbf{G} + \mathbf{F} - \mathbf{u} \\ &= \mathbf{M}\left(\ddot{q}_d + \mathbf{c}\dot{e}\right) + \mathbf{C}\left(\dot{q}_d + \mathbf{c}\mathbf{e}\right) + \mathbf{G} - \mathbf{F} \\ &\quad - \mathbf{C}\mathbf{s} - \mathbf{u} = -\mathbf{C}\mathbf{s} - \mathbf{u} + \mathbf{f},\end{aligned} \tag{8}$$

where $\mathbf{f}(x) = \mathbf{M}\dot{q}_r + \mathbf{C}\mathbf{q}_r + \mathbf{F}$; $\mathbf{q}_r = \dot{q}_d + \mathbf{c}\mathbf{e}$.

It can be derived from the expression $\mathbf{f}(x)$, which contains all the model information in the formula (1). The goal of the control is to approximate it using the radial basis (RBF) neural network control.

The model structure of the RBF neural network is simple and adaptable. For a feedforward neural network, its operation is relatively simple. The RBF neural network is a three-layer network structure, including an input layer, a hidden layer, and an output layer. The input signal in the input layer is $\mathbf{x} = [\, x_1 \;\; x_2 \;\; \ldots \;\; x_n \,]$. The hidden layer contains calculation units, which are called hidden nodes. The neuron activation function is composed of radial basis functions [28].

The output formula of the hidden layer is

$$\mathbf{h}_j(t) = \exp\left(-\frac{\left\|\mathbf{x}(t) - \mathbf{c}_j(t)\right\|^2}{2\mathbf{b}_j^2}\right), \quad j = 1, 2, \ldots, m, \tag{9}$$

where $\mathbf{h}_j(t)$ is the Gaussian function of the j^{th} neuron node, $\mathbf{x}(t)$ is the input of the RBF neural network, m is the number of hidden layer nodes, $\mathbf{c}_j(t)$ is the center vector of the j^{th} hidden node, and b_j is the base width of the Gaussian function.

The output form of the RBF neural network is

$$\mathbf{f}(x) = \boldsymbol{\omega}^T\mathbf{h}_j + \varepsilon, \tag{10}$$

where ω is the weight of the neural network, and ε represents the approximation error of the neural network, which is a small positive real number.

The approximate output of the RBF neural network is

$$\widehat{f}(x) = \widehat{\omega}^T\mathbf{h}_j, \tag{11}$$

where $\widehat{f}(x)$ is the approximation of the neural network to the output $\mathbf{f}(x)$, and $\widehat{\omega}$ is the ideal weight.

Subtract formulas equations (10) and (11) to get

$$\mathbf{f}(x) - \widehat{f}(x) = \boldsymbol{\omega}^T\mathbf{h}_j + \varepsilon - \boldsymbol{\omega}^T\mathbf{h}_j = \widetilde{\omega}^T\mathbf{h}_j + \varepsilon, \tag{12}$$

where $\widetilde{\omega} = \omega - \widehat{\omega}$.

In summary, the control law of the manipulator combined with formulas (5) and (6) is designed as

$$\mathbf{u}_1 = \mathbf{M}\left[\ddot{q}_d + \mathbf{c}\dot{e} + \boldsymbol{\lambda}\text{sign}(\mathbf{s}) + k\mathbf{s}\right] + \mathbf{G} + \mathbf{C}\dot{q} + \mathbf{F} + \widehat{\omega}^T\mathbf{h}_j - \mathbf{v}, \tag{13}$$

where $\mathbf{v} = -\gamma\text{sign}(\mathbf{s})$ is the neural network compensation item, and γ is the adaptive coefficient.

RBF neural network weight adaptive law is

$$\dot{\widehat{\omega}} = \Gamma\mathbf{h}\mathbf{s}^T, \tag{14}$$

where $\Gamma = \Gamma^T > 0$.

Select the Lyapunov function as

$$V = \frac{1}{2}\mathbf{s}^2 + \frac{1}{2}\mathbf{s}^T\mathbf{M}\mathbf{s} + \frac{1}{2}\text{tr}\left(\widetilde{\omega}^T\Gamma^{-1}\widetilde{\omega}\right). \tag{15}$$

Derivation of equation (15) is obtained:

$$\dot{V} = \mathbf{s}\dot{s} + \mathbf{s}^T\mathbf{M}\dot{s} + \frac{1}{2}\mathbf{s}^T\dot{M}\mathbf{s} + \text{tr}\left(\widetilde{\omega}^T\Gamma^{-1}\dot{\widetilde{\omega}}\right). \tag{16}$$

Substitute formulas (6) and (8) into formula (16):

$$\begin{aligned}\dot{V} &= \mathbf{s}\left(-\boldsymbol{\lambda}\text{sign}(\mathbf{s}) - k\mathbf{s}\right) + \mathbf{s}^T\left(-\mathbf{C}\mathbf{s} - \mathbf{u} + \mathbf{f}\right) \\ &\quad + \frac{1}{2}\mathbf{s}^T\dot{M}\mathbf{s} + \text{tr}\left(\widetilde{\omega}^T\Gamma^{-1}\dot{\widetilde{\omega}}\right).\end{aligned} \tag{17}$$

Substitute formulas (8) into (17):

$$\begin{aligned}\dot{V} &= -\mathbf{s}\left(\boldsymbol{\lambda}\text{sign}(\mathbf{s}) + k\mathbf{s}\right) - \mathbf{s}^T\left[\mathbf{M}\left(\ddot{q}_d + \mathbf{c}\dot{e} + \boldsymbol{\lambda}\text{sign}(\mathbf{s}) + k\mathbf{s}\right)\right. \\ &\quad \left. + \mathbf{G} + \mathbf{C}\dot{q} + \mathbf{F}\right] + \frac{1}{2}\mathbf{s}^T\left(\dot{M} - 2\mathbf{C}\right)\mathbf{s} + \mathbf{s}^T\widetilde{\omega}^T\mathbf{h} + \text{tr}\left(\widetilde{\omega}^T\Gamma^{-1}\dot{\widetilde{\omega}}\right) \\ &\quad + \mathbf{s}^T\left(\mathbf{v} + \varepsilon\right).\end{aligned} \tag{18}$$

According to the following conditions:

(1) $\mathbf{s}^T\left(\dot{M} - 2\mathbf{C}\right)\mathbf{s} = 0$

(2) $\mathbf{s}^T\widetilde{\omega}^T\mathbf{h} = \text{tr}\left(\widetilde{\omega}^T\mathbf{h}\mathbf{s}^T\right)$

(3) $\dot{\widetilde{\omega}} = -\dot{\widehat{\omega}} = -\Gamma\mathbf{h}\mathbf{s}^T$

Get

$$\begin{aligned}\dot{V} &= -\mathbf{s}\left(\boldsymbol{\lambda}\text{sign}(\mathbf{s}) + k\mathbf{s}\right) + \mathbf{s}^{\mathrm{T}}\left(\varepsilon - \gamma\text{sign}(\mathbf{s})\right) \\ &\quad - \mathbf{s}^T\left[\mathbf{M}\left(\ddot{q}_d + \mathbf{c}\dot{e} + \boldsymbol{\lambda}\text{sign}(\mathbf{s}) + k\mathbf{s}\right) + \mathbf{G} + \mathbf{C}\dot{q} + \mathbf{F}\right].\end{aligned} \tag{19}$$

Consider

$$-\mathbf{s}\left(\boldsymbol{\lambda}\text{sign}(\mathbf{s}) + k\mathbf{s}\right) \leq -\boldsymbol{\lambda}|\mathbf{s}| - k\mathbf{s}^2 \leq 0. \tag{20}$$

Then,

$$\begin{aligned}\dot{V} &\leq -\mathbf{s}^T\left[\mathbf{M}\left(\ddot{q}_d + \mathbf{c}\dot{e} + \boldsymbol{\lambda}\text{sign}(\mathbf{s}) + k\mathbf{s}\right) + \mathbf{G} + \mathbf{C}\dot{q} + \mathbf{F}\right] + \mathbf{s}^T\left(\varepsilon - \gamma\text{sign}(\mathbf{s})\right) \\ &\leq -\mathbf{s}^T\left[\mathbf{M}\left(\ddot{q}_d + \mathbf{c}\dot{e} + \boldsymbol{\lambda}\text{sign}(\mathbf{s}) + k\mathbf{s}\right) + \mathbf{G} + \mathbf{C}\dot{q} + \mathbf{F} + \varepsilon - \gamma\text{sign}(\mathbf{s})\right].\end{aligned} \tag{21}$$

Let

$$L = \mathbf{M}\left(\ddot{q}_d + \mathbf{c}\dot{e} + \boldsymbol{\lambda}\text{sign}(\mathbf{s}) + k\mathbf{s}\right) + \mathbf{G} + \mathbf{C}\dot{q} + \mathbf{F} + \varepsilon - \gamma\text{sign}(\mathbf{s}). \tag{22}$$

If L is strictly designed as $L > 0$, can get $\dot{V} \le 0$. Because of $V \ge 0$, according to the Lyapunov stability criterion, the closed-loop system is stable.

2.2. Neural Network Fractional-Order Iterative Sliding Surface. Fractional calculus has many definitions, among which the three most commonly used are Caputo type, Grunwald–Letnikov type and Riemann–Liouville type. In this study, Caputo-type calculus is selected, and the definition is as follows [24]:

$${}_aD_t^{\beta} f(t) = \frac{1}{\Gamma(m-\alpha)} \int_a^t \frac{f^{(m)}(\tau)}{(t-\tau)^{\beta-m+1}} \mathrm{d}\tau, \quad m-1 < \beta \le m, \tag{23}$$

where m is the smallest integer greater than β, and $\Gamma(\cdot)$ is the Gamma function. To simplify the expression, replace ${}_aD_t^{\beta}$ with D^{β}.

The fractional-order sliding mode surface function is designed as

$$\mathbf{s} = \mathbf{c}\mathbf{e} + D^{\beta-1}\dot{e}, \tag{24}$$

where $0 < \beta < 1$ and $\mathbf{c} = \text{diag}(c_1, c_2, \ldots, c_n)$, $c_i > 0$.

Remark 1. Compared with the traditional sliding mode surface, the fractional sliding mode surface makes the state trajectory of the system have better dynamic characteristics after entering the sliding mode.

Derivation of (24) is

$$\dot{s} = \mathbf{c}\dot{e} + D^{\beta-1}\ddot{e}. \tag{25}$$

Incorporate equations (1) and (2) into equation (25):

$$\dot{s} = \mathbf{c}\dot{e} + D^{\beta-1}\left(\ddot{q}_d - \mathbf{M}^{-1}(\mathbf{u} - \mathbf{F} - \mathbf{G} - \mathbf{C}\dot{q})\right). \tag{26}$$

Choose the exponential reaching law as

$$\dot{s} = -\boldsymbol{\lambda}\text{sign}(\mathbf{s}) - k\mathbf{s}. \tag{27}$$

Combine (26) and (27), and the approximation part of the RBF neural network, the control law of the manipulator can be designed as

$$\begin{aligned}\mathbf{u}_2 = \mathbf{M}\left[\ddot{q}_d + \mathbf{c}D^{1-\beta}\dot{e} + \boldsymbol{\lambda}D^{1-\beta}\text{sign}(\mathbf{s}) + kD^{1-\beta}\mathbf{s}\right] \\ + \mathbf{G} + \mathbf{C}\dot{q} + \mathbf{F} + \widehat{\omega}^T\mathbf{h}_j - \mathbf{v}.\end{aligned} \tag{28}$$

Choose the same Lyapunov function as in Section 2.1 to prove.

Substitute formula (26) into formula (16) and combine formula (27):

$$\dot{V} = \mathbf{s}(-\boldsymbol{\lambda}\text{sign}(\mathbf{s}) - k\mathbf{s}) - \mathbf{s}^T\left[\mathbf{M}\left[\ddot{q}_d + \mathbf{c}D^{1-\beta}\dot{e} + \boldsymbol{\lambda}D^{1-\beta}\text{sign}(\mathbf{s}) + kD^{1-\beta}\mathbf{s}\right] + \mathbf{G} + \mathbf{C}\dot{q} + \mathbf{F}\right] + \mathbf{s}^T(\varepsilon - \gamma\text{sign}(\mathbf{s})). \tag{29}$$

It can be obtained from the stability analysis part of Section 2.1 and the Lyapunov stability theory that the system is asymptotically stable.

2.3. Neural Network Fractional-Order Iterative Sliding Mode Reaching Law. In the traditional sliding mode control, the commonly used reaching laws are constant speed reaching law, exponential reaching law, and power reaching law. The fractional reaching law selected in this study is

$$D^{\beta}\mathbf{s} = -k\text{sign}(\mathbf{s}), \tag{30}$$

where $0 < \beta < 1$. The speed of the control system reaching the sliding surface is changed by changing the order β and the coefficient k.

Remark 2. Compared with the exponential reaching law, the fractional-order reaching law is not only faster in the reaching stage of sliding mode control but also $\dot{s}$ will not be too large at the time of arrival, which reduces the chattering of the system.

Taking the derivative of formula (30) can obtain

$$\dot{s} = -kD^{1-\beta}\text{sign}(\mathbf{s}). \tag{31}$$

Combine (5) and (31), and the approximation part of the RBF neural network, the control law of the manipulator can be obtained as

$$\mathbf{u}_3 = \mathbf{M}\left[\ddot{q}_d + \mathbf{c}\dot{e} + \boldsymbol{\lambda}D^{1-\beta}\text{sign}(\mathbf{s})\right] + \mathbf{G} + \mathbf{C}\dot{q} + \mathbf{F} + \widehat{w}^T\mathbf{h}_j - \mathbf{v}. \tag{32}$$

Choose the same Lyapunov function as in Section 2.1 to prove.

Combine formula (5) into formula (16), and combine formula (31):

$$\begin{aligned}\dot{V} = \mathbf{s}\left(-kD^{1-\beta}\text{sign}(\mathbf{s})\right) - \mathbf{s}^T\left[\mathbf{M}\left(\ddot{q}_d + \mathbf{c}\dot{e} + kD^{1-\beta}\text{sign}(\mathbf{s})\right)\right. \\ \left. + \mathbf{G} + \mathbf{C}\dot{q} + \mathbf{F}\right] + \mathbf{s}^T(\varepsilon - \gamma\text{sign}(\mathbf{s})).\end{aligned} \tag{33}$$

Consider:

$$-\mathbf{s}\left(kD^{1-\beta}\text{sign}(\mathbf{s})\right) \le -kD^{1-\beta}|\mathbf{s}| \le 0. \tag{34}$$

Then, it can be obtained from the stability analysis part of Section 2.1 and the Lyapunov stability theory that the system is asymptotically stable.

3. Simulation Experiment Analysis

3.1. Simulation Parameter Setting. The two-joint industrial manipulator is selected as an experimental simulation object. According to the dynamic model of the second chapter, the specific parameters of each matrix in the model are [29]

$$\mathbf{M}(\mathbf{q}) = \begin{bmatrix} v + q_{01} + 2q_{02}\cos(q_2) & q_{01} + q_{02}\cos(q_2) \\ q_{01} + q_{02}\cos(q_2) & q_{01} \end{bmatrix},$$

$$\mathbf{C}(\mathbf{q}, \dot{q}) = \begin{bmatrix} -q_{02}\dot{q}_2 \sin(q_2) & -q_{02}(\dot{q}_1 + \dot{q}_2)\sin(q_2) \\ q_{02}\dot{q}_1 \sin(q_2) & 0 \end{bmatrix},$$

$$\mathbf{G}(\mathbf{q}) = \begin{bmatrix} 15g\ \cos\ q_1 + 8.75g\ \cos(q_1 + q_2) \\ 8.75g\ \cos(q_1 + q_2) \end{bmatrix},$$

$$\mathbf{F}(t) = 1000\ \exp\left(\frac{-(t-3)^2}{\left(2 \times 0.1^2\right)}\right), \tag{35}$$

where $v = 13.33$, $q_{01} = 8.98$, $q_{02} = 8.75$, and $g = 9.8$.

Figure 1 is a schematic diagram of the two-joint manipulator structure:

The FOMCON [30] toolbox in MATLAB software is used to complete the numerical simulation. The initial state of the two-joint manipulator system is given as $[q_1\ \ q_2\ \ q_3\ \ q_4] = [0\ \ 3\ \ 1\ \ 0]$, where, $c_1 = c_2 = 40$, $\lambda = 0.5$, $k = 50$, and $0 < \beta < 1$. The structure of RBF neural network design is 2-7-1, the input is $\mathbf{x} = [\mathbf{e}\ \ \dot{e}]$, the specific parameters of Gaussian function are $c_j = [-1.5\ \ -1.0\ \ -0.5\ \ 0\ \ 0.5\ \ 1.0\ \ 1.5]$, $b_j = 10$, and the initial weight of the network is 0 [29]. Suppose the number of iterations is 10, and the two-joint position commands are $q_{1d} = \sin(3t)$ and $q_{2d} = \cos(3t)$, respectively.

Different control laws are used for comparison. The following three control strategies were explicitly used for experimental analysis and comparison with the three control strategies in literature [23].

The following control methods are used for simulation comparison and analysis. Method 1 is the iterative sliding mode control (ISMC) [23], method 2 is the RBF neural network iterative sliding mode control (RISMC), method 3 is the fractional-order iterative sliding mode reaching law (FISMAC) [23], method 4 is the RBF neural network fractional-order iterative sliding mode reaching law (RFISMAC), method 5 is the fractional-order iterative sliding mode surface (FISMSC) [23], and method 6 is the RBF neural network fractional-order iterative sliding mode surface (RFISMSC).

3.2. Simulation Results. Use the control method mentioned above. Compared with the literature [23], the following results are obtained. Figures 2 and 3 show the position track tracking curve and local amplification curve of joint 1 and joint 2 under six control methods. By comparing the local amplification curves, it can be obtained that the tracking curves of joint 1 and joint 2 of the three control strategies based on the RBF neural network are closer to the expected trajectory. So the trajectory tracking performance of the control strategy proposed in this study is better. In other words, the position tracking result of the control strategy based on the RBF neural network after 10 iterations is better than the control strategy proposed in the literature [23].

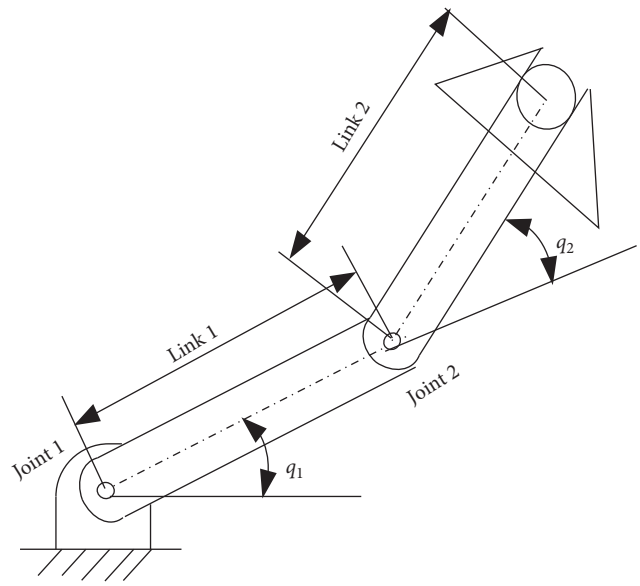

Figure 1: Schematic diagram of the mechanical arm.

Figure 4 shows the control torque of the system and its partial amplification curve under the control strategies of ISMC and RISMC. It can be concluded that the RBF neural network iterative sliding mode control strategy has a better chattering suppression effect than the iterative sliding mode control strategy. Figure 5 shows the control torque of the system under FISMAC and RFISMAC. It can be seen that the RBF neural network fractional-order iterative sliding mode reaching law control strategy and the fractional-order iterative sliding mode reaching law control strategy has a better chattering suppression effect. Figure 6 shows the control input under the control strategy of FISMSC and RFISMSC. Comparing the figure indicates that the RBF neural network fractional-order iterative sliding mode control strategy has a better chattering suppression effect than the fractional-order iterative sliding mode control strategy.

Figures 7–10, respectively, show the maximum absolute value convergence process of the position and velocity error of the two joints under the six control strategies. It can be obtained that the maximum absolute value of the position and velocity error of the control strategy based on the RBF neural network proposed in this study are significantly lower than the contrasted control strategy. Tables 1 and 2 , respectively, show the maximum and minimum values of the maximum absolute value of the position error and speed error of the two-joint manipulator under the six control methods. The comparison can draw the following

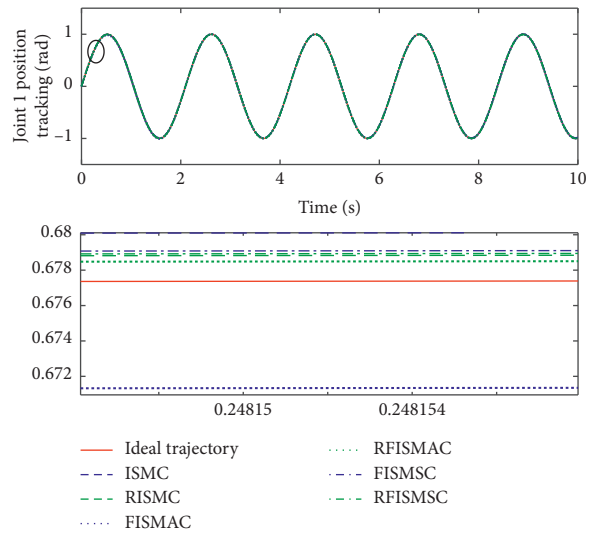

Figure 2: Joint 1 position tracking.

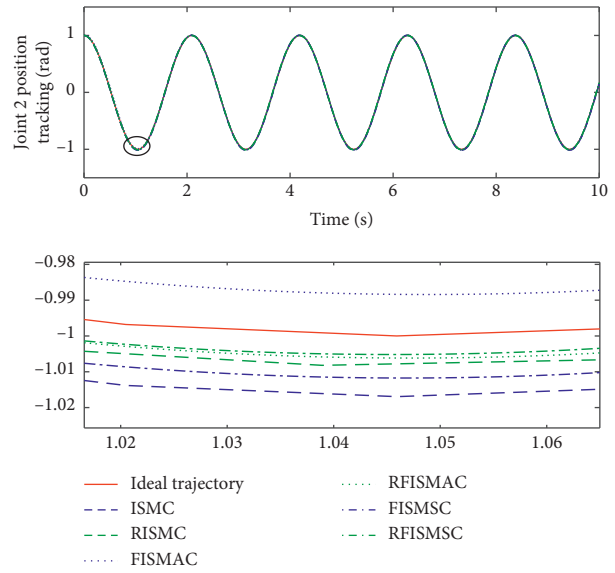

Figure 3: Joint 2 position tracking.

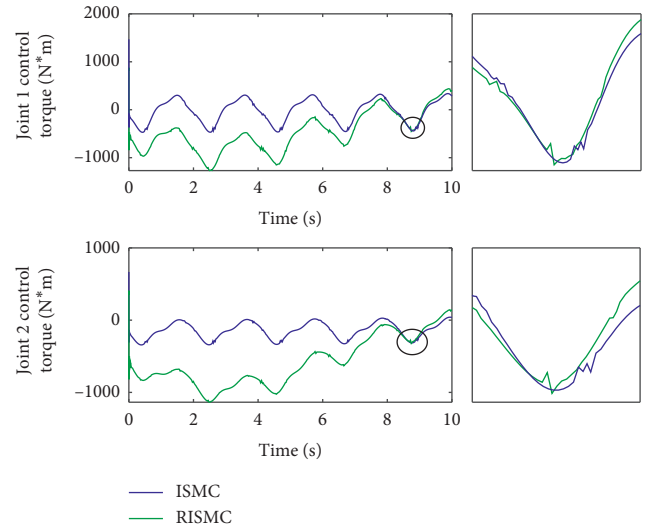

Figure 4: Control input of the system under ISMC and RISMC.

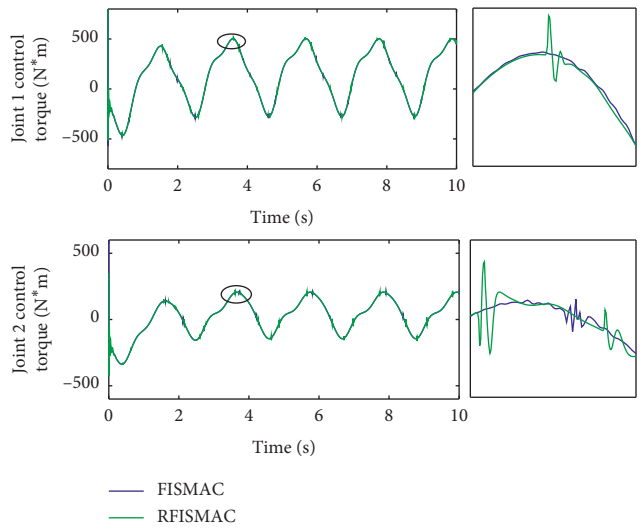

Figure 5: Control input of the system under FISMAC and RFISMAC.

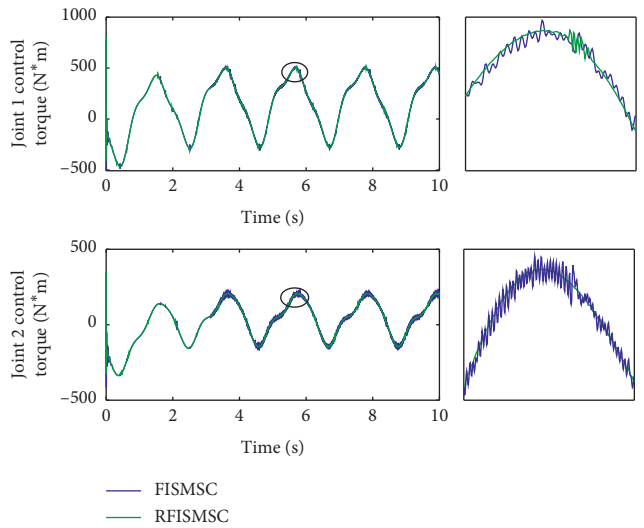

Figure 6: Control input of the system under FISMSC and RFISMSC.

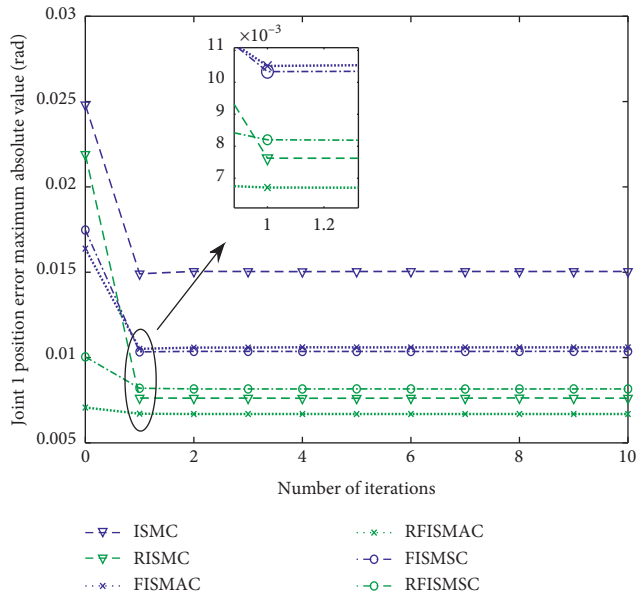

Figure 7: Joint 1 position error maximum absolute value convergence process.

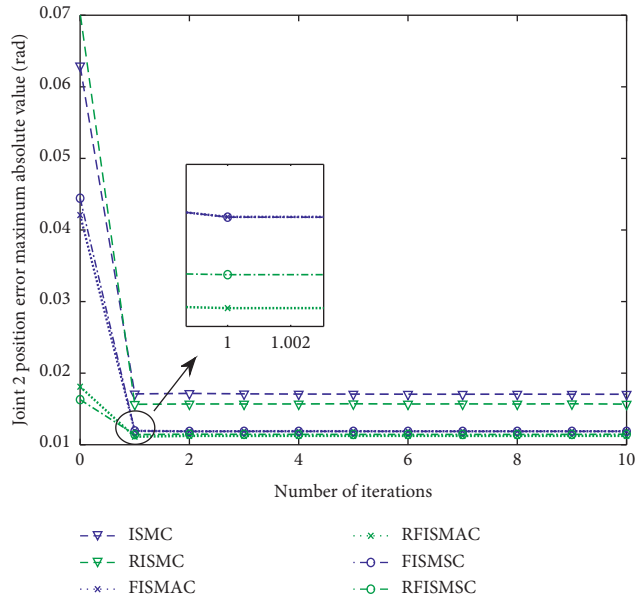

Figure 8: Joint 2 position error maximum absolute value convergence process.

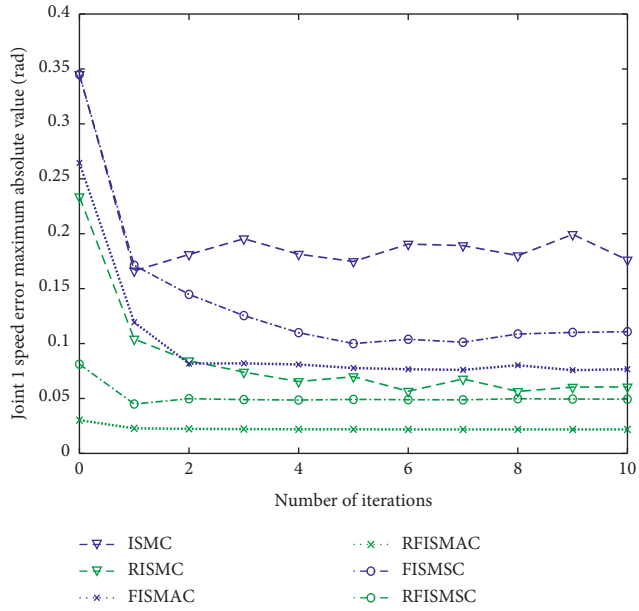

Figure 9: Joint 1 position error maximum absolute value convergence process.

conclusions: the three control strategies proposed in the study and the minimum of the maximum absolute value of position error and speed error are smaller than the control strategies in [23]. It shows that the control strategies proposed in the study have minor position and speed errors, higher position tracking accuracy, and control accuracy.

To further verify the effectiveness of the controller, the unit step signal is used as the given input signal. At the same time, overshoot, rise time, adjustment time, and root mean square error are selected as the basis for judgment. The following results are obtained. Figures 11 and 12 show the unit step response under the six control methods. Table 3 provides the unit step response performance indicators under the six control methods. It can be seen from Figures 9 and 10 that although the overshoot of the control strategy based on the RBF neural network proposed in the study has become larger, it is still within the required range, and the rise time and adjustment time have been reduced. Table 3 provides the specific data; the rise time, adjustment time, and root mean square error are all reduced compared with the

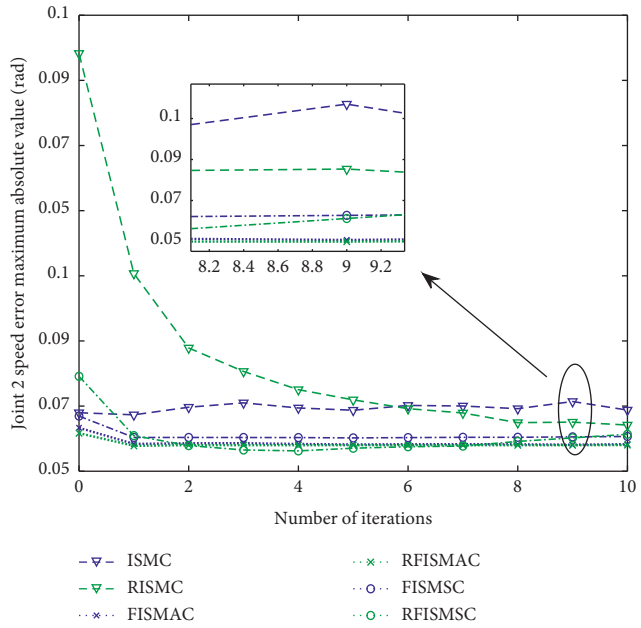

Figure 10: Joint 2 position error maximum absolute value convergence process.

Table 1: Comparison of position error of two joints under six control methods.

Control strategies	Joints	Maximum value (rad)	Minimum value (rad)
ISMC	Joint 1	0.0248	0.0150
	Joint 2	0.0629	0.0171
RISMC	Joint 1	0.0219	0.0076
	Joint 2	0.0703	0.0157
FISMAC	Joint 1	0.0164	0.0106
	Joint 2	0.0421	0.0119
RFISMAC	Joint 1	0.0071	0.0067
	Joint 2	0.0181	0.0111
FISMSC	Joint 1	0.0175	0.0104
	Joint 2	0.0444	0.0119
RFISMSC	Joint 1	0.1000	0.0082
	Joint 2	0.0163	0.0114

Table 2: Comparison of speed error of two joints under six control methods.

Control strategies	Joints	Maximum value (rad)	Minimum value (rad)
ISMC	Joint 1	0.3452	0.1761
	Joint 2	0.1070	0.0939
RISMC	Joint 1	0.2342	0.0494
	Joint 2	0.6482	0.0718
FISMAC	Joint 1	0.2645	0.0757
	Joint 2	0.0667	0.0407
RFISMAC	Joint 1	0.0303	0.0218
	Joint 2	0.0584	0.0398
FISMSC	Joint 1	0.3451	0.1104
	Joint 2	0.0847	0.0501
RFISMSC	Joint 1	0.0812	0.0492
	Joint 2	0.1418	0.0309

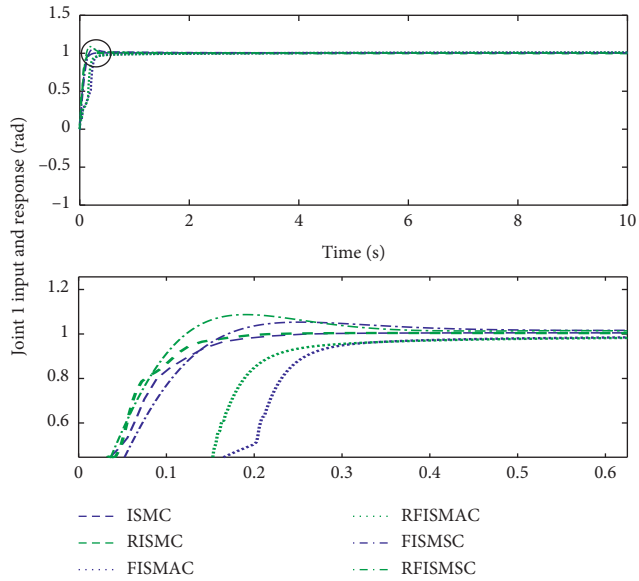

Figure 11: Unit step response of the six control methods of joint 1.

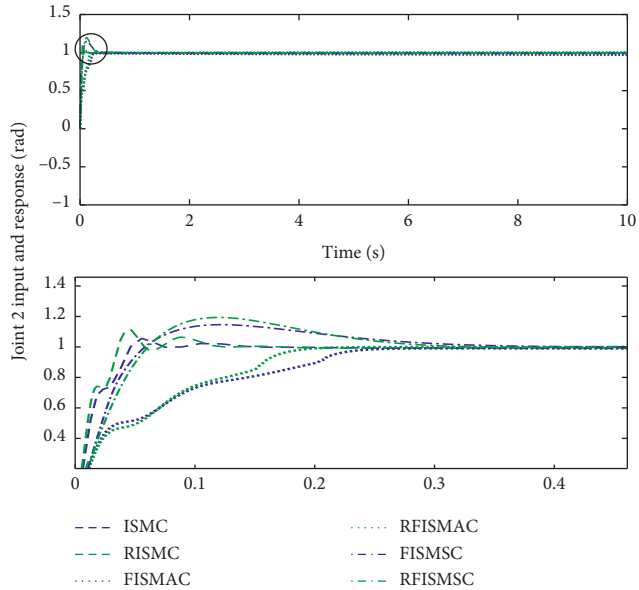

Figure 12: Unit step response of the six control methods of joint 2.

Table 3: Unit step response performance index.

Control strategies	Joints	Maximum overshoot, σ (%)	Rise time, Tr (s)	Adjustment time, Ts (s)	Root mean square error (rad)
ISMC	Joint 1	0.54	0.27	0.27	0.01671
	Joint 2	5.61	0.048	0.03	0.02395
RISMC	Joint 1	0	0.2	0.24	0.01593
	Joint 2	8.23	0.03	0.03	0.02026
FISMAC	Joint 1	0	0.5	0.5	0.00206
	Joint 2	0	0.25	0.25	0.00416
RFISMAC	Joint 1	0	0.3	0.28	0.00189
	Joint 2	0	0.16	0.17	0.00391
FISMSC	Joint 1	5.33	0.165	0.5	0.00371
	Joint 2	14.6	0.06	0.4	0.00557
RFISMSC	Joint 1	8.22	0.1	0.36	0.00300
	Joint 2	19.1	0.05	0.31	0.00433

contrast control strategy. Therefore, it can be obtained that the control method proposed in this study can better meet the dynamic performance index of the unit step response.

In this study, the simulation experiment is compared with the literature [23], so the parameters of the literature [23] are used for the simulation experiment. From the simulation experiment results, it can be obtained that the effect of parameter changes on the convergence speed, tracking accuracy, and control input is as follows:

(1) If the parameter c is increased, the overshoot of the system state will increase, but the convergence speed will be faster

(2) If λ and k are reduced, the chattering of the system will decrease, but the convergence speed will decrease, and the tracking accuracy will decrease

(3) If the parameter Γ is increased, the system state overshoot will increase, the tracking accuracy will increase, and the control input will increase. This study uses reference [29] and trial and error to determine this parameter.

4. Conclusions

Aiming at low position tracking accuracy when the information of industrial manipulator model is uncertain, the RBF neural network fractional iterative sliding mode control strategy is proposed. First, the RBF neural network is used to estimate the uncertainties such as external interference and modeling errors of the manipulator to improve the system's control accuracy and robustness. Second, the fractional sliding mode surface and the fractional reaching law are designed to reduce the chattering phenomenon of the system effectively. In addition, in the design of the adaptive law of the RBF neural network, the appropriate Lyapunov function is selected for stability analysis. The appropriate adaptive law is obtained from it to update the weights of the neural network online. Finally, the simulation comparison experiment is carried out with the fractional-order iterative sliding mode control strategy, and the following conclusions are obtained.

When the given input is a trigonometric function, the minimum value of the maximum absolute value of the dual-joint position error is increased by 49.0%, 8.2%; 36.8%, 6.7%; and 21.2%, 4.2% rad compared with ISMC, FISMAC, and FISMSC to improve the position tracking accuracy of the system. When the given input is a unit step response, the control methods proposed in this study reduce the adjustment time of joint 1 by 11.1%, 44%, and 28% and joint 2 by 0%, 32%, and 22.5%. At the same time, its rise time and root mean square error is reduced. Therefore, the control methods proposed in this study improve the tracking accuracy. Simulation experiments verify the effectiveness of the proposed control method. The weights of the RBF neural network can be updated online by adaptive law, and other parameters of the RBF neural network also have a certain influence on the neural network. Therefore, the next step is to adjust the other parameters of the neural network through the adaptive law in the future controller design.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Natural Science Foundation of Gansu Province (20JR5RA419) and Lanzhou Jiaotong University-Tianjin University Innovation Fund Project (2019053).

References

[1] T. Kangru, J. Riives, K. Mahmood, and T. Otto, "Suitability analysis of using industrial robots in manufacturing," *Proceedings of the Estonian Academy of Sciences*, vol. 68, no. 4, pp. 383–388, 2019.

[2] G. Meng, L. L. Han, and C. F. Zhang, "Research progress and technical challenges of space robot," *Acta Aeronautica et Astronautica Sinica*, vol. 42, no. 1, pp. 1–27, 2021.

[3] Y. Hua and Y. X. Gu, "Kinematics analysis of a robotic arm in a traffic cone automatic recovery and placement device," *Mechanical Engineering and Technology*, vol. 9, no. 1, pp. 1–12, 2020.

[4] C. Xiao, Y. L. Zhou, and H. Y. Sheng, "Modular design of mechanical noumenon for industrial robots," *China Mechanical Engineering*, vol. 27, no. 8, pp. 1018–1025, 2016.

[5] Y. Tian and Y. L. Cai, "A new variable power reaching law for sliding mode control," *Journal of Chinese Inertial Technology*, vol. 27, no. 2, pp. 241–247, 2019.

[6] S. Mobayen, "A novel global sliding ModeControl based on exponential reaching law for a class ofUnderactuated systems with external disturbances," *Journal of Computational and Nonlinear Dynamics*, vol. 11, no. 1, pp. 1–9, 2016.

[7] Z. J. Liu, J. K. Liu, and W. H. Liu, "An adaptive iterative learning algorithm for boundary control of a flexible manipulator," *International Journal of Adaptive Control and Signal Processing*, vol. 31, no. 6, pp. 903–916, 2019.

[8] J. L. Zhou and Q. L. Zhang, "Adaptive fuzzy control of uncertain robotic manipulator," *Mathematical Problems in Engineering*, vol. 2018, no. 2, 10 pages, Article ID 4703492, 2018.

[9] S. X. Hou, Y. D. Chu, and J. T. Fei, "Adaptive type-2 fuzzy neural network inherited terminal sliding mode control for power quality improvement," *IEEE Transactions on Industrial Informatics*, vol. 18, 2021.

[10] V. M. Hernández-Guzmán and J. Orrante-Sakanassi, "Global PID control of robot manipulators equipped with PMSMs," *Asian Journal of Control*, vol. 20, no. 2, pp. 1–14, 2018.

[11] Y. G. Sun, J. Q. Y. Xu, and G. B. Lin, "RBF neural network-based supervisor control for maglev vehicles on an elastic track with network time-delay," *IEEE Transactions on Industrial Informatics*, vol. 17, 2020.

[12] S. Hou, Y. Chu, and J. Fei, "Intelligent global sliding mode control using recurrent feature selection neural network for active power filter," *IEEE Transactions On Industrial Electronics*, vol. 68, no. 8, pp. 7320–7329, 2021.

[13] J. Fang, R. Yin, and X. Lei, "An adaptive decoupling control for three-axis gyro stabilized platform based on neural networks," *Mechatronics*, vol. 27, no. 1, pp. 38–46, 2015.

[14] S. M. Song, L. W. Deng, and X. L. Chen, "Application characteristics of fractional calculus in sliding mode control,"

Journal of Chinese Inertial Technology, vol. 22, no. 4, pp. 439–444, 2014.

[15] G. S. Zhang and X. L. Li, "Design and analysis of a new power reaching law for sliding mode control," *Journal of Tianjin University (Science and Technology)*, vol. 53, no. 11, pp. 1112–1119, 2020.

[16] SMobayen, "Adaptive global terminal sliding mode control scheme with ImprovedDynamic surface for uncertain nonlinear systems," *International Journal of Control, Automation and Systems*, vol. 16, no. 1, pp. 1–9, 2018.

[17] H. Y. Jin and X. M. Zhao, "Speed control of permanent magnet linear synchronous motor based on complementary sliding mode control and iterative learning control," *Control Theory & Applications*, vol. 37, no. 4, pp. 918–924, 2020.

[18] Z. Z. Zhang, D. Ye, and Z. W. Sun, "Sliding mode fault tolerant attitude control for satellite based on iterative learning observer," *Journal of National University of Defense Technology*, vol. 40, no. 1, pp. 918–924, 2018.

[19] X. Lu, H. Zhang, J. Guo, and J. Zhou, "Iterative learning control combination with adaptive sliding mode technique for a hypersonic vehicle," *Xibei Gongye Daxue Xuebao/Journal of Northwestern Polytechnical University*, vol. 37, no. 6, pp. 1120–1128, 2019.

[20] G. Iman, R. N. Abolfazl, and S. Jalil, "Sliding mode based fractional-order iterative learning control for a nonlinear robot manipulator with bounded disturbance," *Transactions of the Institute of Measurement and Control*, vol. 40, no. 1, pp. 49–60, 2018.

[21] M. Matusiak and P. Ostalczyk, "Problems in solving fractional differential equations in a microcontroller implementation of an FOPID controller," *Archives of Electrical Engineering*, vol. 68, no. 3, pp. 565–577, 2019.

[22] S. S. Lyu and H. Lin, "Composite control for electric dynamic loading system based on fractional order iterative learning," *Journal of Beijing University of Aeronautics and Astronautics*, vol. 42, no. 9, pp. 1944–1951, 2016.

[23] X. Zhang, W. R. Lu, and Z. C. Miao, "Iterative sliding mode control of robotic arm based on fractional calculus," *Journal of Measurement Science and Instrumentation*, vol. 12, no. 1, pp. 1–9, 2020.

[24] X. P. Shi and S. R. Liu, "A survey of trajectory tracking control for robot manipulators," *Control Engineering of China*, vol. 18, no. 1, pp. 116–122, 2011.

[25] Y. G. Sun, J. Q. Xu, and H. Wu, "Deep learning based semi-supervised control for vertical security of maglev vehicle WithGuaranteed bounded airgap," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, 2021.

[26] T. Y. Vu, Y. N. Wang, and V. C. Pham, "Robust adaptive sliding mode neural networks control for industrial robot manipulators," *International Journal of Control Automation and Systems*, vol. 17, no. 3, pp. 783–792, 2019.

[27] Y. L. Zhang, Y. J. An, and G. Y. Wang, "Multi motor neural PID relative coupling speed synchronous control," *Archives of Electrical Engineering*, vol. 69, no. 1, pp. 69–88, 2020.

[28] F. Wang, Z. Q. Chao, and L. B. Huang, "Trajectory tracking control of robot manipulator based on RBF neural network and fuzzy sliding mode," *Cluster Computing*, vol. 22, no. 3, pp. 5799–5809, 2012.

[29] J. K. Liu, *Design of Robot Control System and MATLAB Simulation*, Tsinghua University Press, Beijing, China, 2008.

[30] S. Dadras and H. R. Momeni, "Fractional terminal sliding mode control design for a class of dynamical systems with uncertainty," *Communications in Nonlinear Science & Numerical Simulation*, vol. 17, no. 1, pp. 367–377, 2012.