

Article

Research on Neural Network Terminal Sliding Mode Control of Robotic Arms Based on Novel Reaching Law and Improved Salp Swarm Algorithm

Jianguo Duan¹, Hongzhi Zhang^{2,*}, Qinglei Zhang¹ and Jiyun Qin¹¹ Free Trade Zone Supply Chain Research Institute, Shanghai Maritime University, Shanghai 201306, China² Logistics Engineering College, Shanghai Maritime University, Shanghai 201306, China

* Correspondence: 202130210086@stu.shmtu.edu.cn

Abstract: Modeling errors and external disturbances have significant impacts on the control accuracy of robotic arm trajectory tracking. To address this issue, this paper proposes a novel method, the neural network terminal sliding mode control (ALSSA-RBFTSM), which combines fast nonsingular terminal sliding mode (FNTSM) control, radial basis function (RBF) neural network, and an improved salp swarm algorithm (ALSSA). This method effectively enhances the trajectory tracking accuracy of robotic arms under the influence of uncertain factors. Firstly, the fast nonsingular terminal sliding surface is utilized to enhance the convergence speed of the system and achieve finite-time convergence. Building upon this, a novel multi-power reaching law is proposed to reduce system chattering. Secondly, the RBF neural network is utilized to estimate and compensate for modeling errors and external disturbances. Then, an improved salp swarm algorithm is proposed to optimize the parameters of the controller. Finally, the stability of the control system is demonstrated using the Lyapunov theorem. Simulation and experimental results demonstrate that the proposed ALSSA-RBFTSM algorithm exhibits superior robustness and trajectory tracking performance compared to the global fast terminal sliding mode (GFTSM) algorithm and the RBF neural network fast nonsingular terminal sliding mode (RBF-FNTSM) algorithm.



Citation: Duan, J.; Zhang, H.; Zhang, Q.; Qin, J. Research on Neural Network Terminal Sliding Mode Control of Robotic Arms Based on Novel Reaching Law and Improved Salp Swarm Algorithm. *Actuators* **2023**, *12*, 464. <https://doi.org/10.3390/act12120464>

Academic Editors: Ahmad Taher Azar, Amjad J. Humaidi and Ammar K. Al Mhdawi

Received: 7 November 2023

Revised: 8 December 2023

Accepted: 11 December 2023

Published: 13 December 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: robotic arm; fast nonsingular terminal sliding mode; RBF neural network; improved salp swarm algorithm; novel multi-power reaching law; Lyapunov theorem

1. Introduction

With the development of automation and intelligent manufacturing, the application scope of robots and automation equipment is becoming increasingly extensive. As integral components of robots, robotic arms are widely utilized in various fields, such as production, manufacturing, healthcare, aviation, logistics, and agriculture [1]. Trajectory tracking of robotic arms is a crucial technique in robot control, enabling them to accurately follow a given path in space, thereby achieving high-quality manufacturing and automation operations. Currently, trajectory tracking control of robotic arms faces challenges such as strong nonlinearity, high uncertainty, and complexity. Therefore, researching high-performance trajectory tracking control strategies for robotic arms is of significant importance.

PID control has the advantages of simplicity, ease of tuning, and ease of implementation, making it a popular strategy in the field of robotic arm control [2–4]. However, PID control also suffers from the disadvantages of limited adaptability to nonlinear systems, poor robustness to parameter variations and uncertainties, and difficulty in dealing with delays and time lags. Therefore, scholars both domestically and internationally have proposed many other methods for trajectory tracking control of robotic arms, such as robust control [5], adaptive control [6], fuzzy control [7], iterative learning control [8], neural network control [9], sliding mode control [10], and backstepping control [11]. Sliding mode control is widely applied in the field of robot control due to its high tracking accuracy,

strong robustness, and low requirement for system modeling. A terminal sliding mode control method was proposed by Zhao et al. [12], which can stabilize the system state to an equilibrium point within finite time without requiring precise robot dynamics models. This approach offers faster computation speed and a simpler controller structure, but the speed of convergence still needs to be improved. To address this problem, Doan et al. [13] proposed a fast terminal sliding mode control method that combines fast terminal sliding surfaces with the super-twisting control law. This combination results in smoother control torque, enabling faster and more accurate compensation for external disturbances and nonlinear elements. It ensures system stability and robustness. The method further improves the convergence speed but suffers from the singularity problem. For the singularity problem, Jin et al. [14] introduced a nonsingular terminal sliding mode control method that combines nonsingular terminal sliding surfaces with time delay estimation. It achieves fast convergence using nonsingular terminal sliding mode and implements model-free control using time delay estimation. This method solves the singularity problem, is easy to implement, and has high robustness and accuracy, but the convergence speed is relatively slow. Considering the two main issues of convergence speed and singularity, a fast nonsingular terminal sliding mode control method was proposed by Liang et al. [15]. It utilizes a second-order fast nonsingular terminal sliding mode to achieve rapid convergence, avoid singularities, and reduce chattering. Furthermore, an artificial neural network is introduced to handle model uncertainties and disturbances without the need for any prior knowledge. The method can achieve finite-time convergence with a fast convergence rate while avoiding the singularity problem.

However, sliding mode control suffers from chattering, which can degrade control performance and even lead to system instability. To address this issue, the reaching law approach is often employed to adjust and control the system states. A new exponential reaching law was proposed by Wang et al. [16], which incorporates the system state variable to relate the convergence speed with the variation of the system state. This approach enhances the dynamic performance and robustness of the system, effectively suppressing the phenomenon of oscillations. This method speeds up the convergence of the system and allows the system to approach the sliding mold surface faster, yet it may also lead to high-frequency oscillations and overshoot phenomena, resulting in degradation of control performance. To address this problem, Xia et al. [17] combined a double-power reaching law with an improved terminal sliding mode. The double-power reaching law ensures that the system can reach the sliding surface within a finite time from any initial state, while the improved terminal sliding surface ensures that position and velocity errors approximate zero. This method can balance the convergence speed and stability of the system within a certain range, ensuring faster convergence speed and effectively attenuating the jitter vibration phenomenon of the robotic arm. It achieves high tracking accuracy and a strong anti-interference ability, but it also leads to a relatively complex parameter adjustment. Additionally, Ba et al. [18] designed a composite reaching law that combines the cotangent function and the exponential reaching law. This approach shortens the reaching time to the sliding surface and reduces the velocity near the sliding surface. It makes the convergence process smoother and avoids the problem of high-frequency jittering caused by excessive speed when approaching the sliding mold surface. In order to solve the jitter problem caused by the sign function, Zhang et al. [19] designed an improved multiple-power reaching law by replacing the sgn functions with the sigmoid functions, utilizing the smoothing property of the sigmoid functions to reduce the jitter and vibration caused by the sgn functions. This method enhances the control quality and convergence speed of sliding mode control, exhibiting strong robustness and versatility.

Neural network control has the advantages of high precision, low latency, and strong adaptability, making it highly advantageous in nonlinear control, and extensive research has been conducted in this area [20]. Liu et al. [21] studied a robot neural network control system based on a genetic algorithm. The genetic algorithm is used to optimize the neural network, simplifying the network structure and improving tracking effectiveness. However,

the method has high computational complexity and requires a long optimization time. He and Dong [22] proposed a fuzzy neural network learning algorithm to identify uncertain system models. This method does not require prior knowledge or a sufficient amount of observation data about uncertainties. Impedance learning is introduced to address the interaction between the robot and the environment. This control method ensures the tracking performance of the system under state constraints and uncertainties, but the design and parameter tuning of the fuzzy neural network are relatively complex. Tlijani et al. [23] presented a non-singular fast terminal sliding mode control strategy based on a wavelet neural network observer. A wavelet observer is designed using the online approximation capability of neural networks to estimate modeling errors, external disturbances, and uncertainties in joint robots. This control method can overcome the jitter vibration phenomenon and ensure the accuracy and stability of the position control of articulated robots, but the design and parameter adjustment of the observer are relatively complex and sensitive to the modeling and observation errors of the system. Several of the above methods are relatively complex; thus, Sun et al. [24] designed a neural network control method based on radial basis functions, which utilizes neural networks to approximate the unknown model of the robot and deal with the uncertainty of the system with good tracking performance and small tracking error. Compared with other neural network control methods, this method has a fast training speed and relatively simple parameter adjustment. On this basis, Fan et al. [25] developed a sliding mode controller based on RBF neural networks. This controller utilizes the sliding mode control algorithm to counteract external disturbances and employs the radial basis function neural network control algorithm to address system uncertainties. The combination of the two control methods ensures a relatively simple controller structure while improving tracking accuracy and robustness.

To address the problem of decreased trajectory tracking accuracy in robotic arms caused by modeling errors and external disturbances, this paper proposes a neural network terminal sliding mode control algorithm based on a novel reaching law and an improved salp swarm algorithm. Firstly, a fast nonsingular terminal sliding mode surface is selected to achieve finite-time convergence and avoid singularity issues. We propose a novel multi-power reaching law, replacing the sign functions with saturation functions to reduce system chattering. Secondly, considering the capability of RBF neural networks to approximate any continuous functions, we utilize RBF neural networks to estimate model uncertainties and external disturbances. An adaptive law is designed to automatically adjust the neural network weights. Furthermore, we propose an adaptive leader salp swarm algorithm to optimize the parameters of the controller, thereby improving the effectiveness of trajectory tracking control. Finally, we conduct simulations of ABB six-axis robotic arm trajectory tracking control in the Matlab environment to validate the effectiveness and reliability of the proposed control method. The main contributions of this paper are as follows:

1. A novel reaching law is proposed, which utilizes the tanh and sigmoid functions to replace the sign functions in traditional multi-power reaching law. This enables the system state to slide rapidly and accurately onto the sliding surface, suppressing oscillations and enhancing system stability and disturbance rejection capabilities.
2. An improved salp swarm algorithm is proposed, incorporating adaptive inertia weight factors and adaptive adjustment strategies to enhance convergence speed, overall performance, and solution accuracy.
3. A novel neural network terminal sliding mode controller is proposed and applied to the trajectory tracking control of an ABB robot. The superior control performance of the controller is verified through simulation and experimental validation.

The organization of this paper is as follows: Section 2 presents the dynamic model of the robot and the design of the ALSSA-RBFTSM controller. Section 3 provides the simulation and experimental results. Section 4 presents the conclusions of this paper.

2. Controller Design

2.1. Robotic Arm Dynamics Model

The dynamic model of the robotic arm can be obtained using the Lagrange method [26], as follows:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + F(\dot{q}) + \tau_d = \tau \quad (1)$$

where $q \in R^{n \times 1}$ represents the joint angles, $\dot{q} \in R^{n \times 1}$ represents the joint angular velocities, $\ddot{q} \in R^{n \times 1}$ represents the joint angular accelerations, $M(q) \in R^{n \times n}$ represents the inertia matrix, $C(q, \dot{q}) \in R^{n \times n}$ represents the Coriolis and centrifugal force matrix, $G(q) \in R^{n \times 1}$ represents the gravity matrix, $F(\dot{q}) \in R^{n \times 1}$ represents the friction matrix, $\tau \in R^{n \times 1}$ represents the control input matrix, and $\tau_d \in R^{n \times 1}$ represents the external disturbance matrix.

In practice, obtaining an accurate dynamic model of a robotic arm is challenging, and modeling errors can degrade control performance and reduce trajectory tracking accuracy. Therefore, considering the modeling errors in the dynamic modeling of the robotic arm, the dynamic model can be divided into a deterministic part and an uncertain part [27]. Thus, $M(q)$, $C(q, \dot{q})$, and $G(q)$ can be represented as:

$$\begin{cases} M(q) = M_0(q) + \Delta M(q) \\ C(q, \dot{q}) = C_0(q, \dot{q}) + \Delta C(q, \dot{q}) \\ G(q) = G_0(q) + \Delta G(q) \end{cases} \quad (2)$$

The aggregate uncertainty arising from modeling errors and external disturbances can be represented as:

$$f = \Delta M(q)\ddot{q} + \Delta C(q, \dot{q})\dot{q} + \Delta G(q) + F(\dot{q}) + \tau_d \quad (3)$$

In this case, the dynamic equation can be reexpressed as:

$$M_0(q)\ddot{q} + C_0(q, \dot{q})\dot{q} + G_0(q) + f = \tau \quad (4)$$

2.2. Design of Fast Nonsingular Terminal Sliding Mode Control

Define the system tracking error as:

$$e = q - q_d \quad (5)$$

where q is the actual position vector and q_d is the desired position vector.

The FNTSM surface [28] is designed as:

$$s = \dot{e} + \alpha e + \beta |e|^\lambda \text{sgn}(e) \quad (6)$$

where $\alpha, \beta \in R, 1 < \lambda < 2$.

The derivative of the sliding mode surface function can be obtained as:

$$\dot{s} = \ddot{e} + \alpha \dot{e} + \beta \lambda |e|^{\lambda-1} \dot{e} \quad (7)$$

Without considering the compound disturbance of the system, and letting $\dot{s} = 0$, the equivalent control law can be obtained as:

$$u_{eq} = M_0(q)\ddot{q}_d - M_0(q) \left[\alpha \dot{e} + \beta \lambda |e|^{\lambda-1} \dot{e} \right] + C_0(q, \dot{q}) + G_0(q) \quad (8)$$

The traditional multi-power reaching law [29] is as follows:

$$\dot{s} = -k_1 |s|^{a_1} \text{sgn}(s) - k_2 |s|^{a_2} \text{sgn}(s) - k_3 |s|^{a_3} \text{sgn}(s) - k_4 s \quad (9)$$

where $k_1 > 0, k_2 > 0, k_3 > 0, k_4 > 0, a_1 > 1, 0 < a_2 < 1$, and the value of a_3 is taken as follows:

$$a_3 = \begin{cases} \max\{a_1, |s|\}, & |s| \geq 1 \\ \min\{a_2, |s|\}, & |s| < 1 \end{cases} \tag{10}$$

when the system state satisfies the condition $|s| < 1$, and the reaching law is mainly influenced by $-k_2|s|^{a_2}sgn(s) - k_3|s|^{a_3}sgn(s)$. When the system state satisfies the condition $|s| \geq 1$, the reaching law is mainly influenced by $-k_1|s|^{a_1}sgn(s) - k_3|s|^{a_3}sgn(s)$. The value of a_3 ensures that the system can adaptively change the exponential parameter in the reaching law, resulting in a faster convergence rate.

Considering the chattering issue caused by the sign functions, this paper proposes improvements to the multi-power reaching law. The sigmoid function and tanh function have the advantages of continuity and fast response. As the system state approaches the sliding surface, the output of the tanh function gradually saturates, resulting in a slower convergence speed. When the system state is far from the sliding surface, the convergence speed of the sigmoid function is relatively fast. Therefore, the sigmoid function is used to replace the first sign function in the multi-power reaching law, the tanh function is used to replace the second sign function, and a nonlinear function is designed to replace the third sign function. This helps avoid the control torque chattering caused by the sign functions in sliding mode control, allowing the system to enter the sliding surface more smoothly and quickly. The designed nonlinear function is as follows:

$$f(s) = \begin{cases} sigmoid(s), & |s| \geq 1 \\ tanh(s), & |s| < 1 \end{cases} \tag{11}$$

The newly designed reaching law in this paper is as follows:

$$\dot{s} = -k_1|s|^{a_1}sigmoid(s) - k_2|s|^{a_2}tanh(s) - k_3|s|^{a_3}f(s) - k_4s \tag{12}$$

when the system state satisfies the condition $|s| < 1$, the reaching law is mainly influenced by $-k_2|s|^{a_2}tanh(s) - k_3|s|^{a_3}tanh(s)$, leading to a reduction in the adjustment magnitude of the system, thereby suppressing the occurrence of oscillations and vibrations. When the system state satisfies the condition $|s| \geq 1$, the reaching law is mainly influenced by $-k_1|s|^{a_1}sigmoid(s) - k_3|s|^{a_3}sigmoid(s)$, enabling the system to adjust its state more quickly and approach the sliding surface rapidly. By leveraging the characteristics of the tanh and sigmoid functions, the system exhibits a certain degree of adaptability to parameter variations and disturbances, thus enhancing its stability and reliability.

According to the reaching law, the switching control law can be obtained as follows:

$$u_{sw} = M_0(q) [k_1|s|^{a_1}sigmoid(s) + k_2|s|^{a_2}tanh(s) + k_3|s|^{a_3}f(s) + k_4s] \tag{13}$$

2.3. Design of RBF Neural Network

Let the input vector of the neural network be $x = (x_1, x_2, \dots, x_n)^T$, the hidden layer basis function be $h = (h_1, h_2, \dots, h_j)^T$, and the output be $y = (y_1, y_2, \dots, y_m)^T$. The functional expression of the RBF neural network can be expressed as:

$$h_j(x) = \exp\left(-\frac{\|x - c_j\|^2}{2b_j^2}\right), j = 1, 2, \dots, m \tag{14}$$

where $c_j = (c_{j1}, c_{j2}, \dots, c_{jn})^T$ denotes the center vector of the function, $b_j = (b_{j1}, b_{j2}, \dots, b_{jn})^T$ denotes the bandwidth of the Gaussian basis function, and m denotes the number of network nodes in the hidden layer.

The expression for the approximation of a nonlinear function by an RBF neural network is:

$$f(x) = W^{*T}h(x) + \varepsilon \tag{15}$$

where W^* represents the ideal network weights, ε is the neural network approximation error, and it satisfies $\varepsilon \leq \varepsilon_N$.

The RBF neural network is utilized to approximate $f(x)$, and the output of the neural network is:

$$\hat{f}(x) = \hat{W}^T h(x) \quad (16)$$

where \hat{W} represents the actual weights of the neural network.

Define the weight estimation error as:

$$\tilde{W} = W^* - \hat{W} \quad (17)$$

Take the derivative of the Equation (17):

$$\dot{\tilde{W}} = -\dot{\hat{W}} \quad (18)$$

Define the neural network output estimation error [30] as:

$$\begin{aligned} \tilde{f} &= f - \hat{f} \\ &= W^{*T} h(x) + \varepsilon - \hat{W}^T h(x) \\ &= \tilde{W}^T h(x) + \varepsilon \end{aligned} \quad (19)$$

In order to enhance the performance of the RBF neural network, a neural network adaptive law is introduced, allowing for adaptive updates of the neural network weights. The neural network adaptive update law designed in this study is as follows:

$$\dot{\hat{W}} = \gamma h(x) s^T \quad (20)$$

where γ is the adjustment factor to be designed.

Define the robust term as:

$$v = \varepsilon_N \text{sgn}(s) \quad (21)$$

The total control law of the system at this point is:

$$\begin{aligned} u &= u_{eq} - u_{sw} + \hat{f} - v \\ &= M_0(q) \ddot{q}_d - M_0(q) [\alpha \dot{e} + \beta \lambda |e|^{(\lambda-1)} \dot{e}] + C_0(q, \dot{q}) + G_0(q) + \hat{W}^T h - \varepsilon_N \text{sgn}(s) - \\ &M_0(q) [k_1 |s|^{a_1} \text{sigmoid}(s) + k_2 |s|^{a_2} \tanh(s) + k_3 |s|^{a_3} f(s) + k_4 s] \end{aligned} \quad (22)$$

2.4. Control System Stability Analysis

2.4.1. Certificate of Necessity

Assuming that the Lyapunov function $V(x)$ is asymptotically stable, to prove that:

1. $V(x) > 0$ holds for all $x \neq 0$: since $V(x)$ is asymptotically stable, according to the definition of the Lyapunov function, there exists a positive constant a and a positive constant b , such that for all x satisfying $\|x\| > a$, there is $V(x) > b$. That is to say, for all non-zero vectors x , as long as their paradigm is greater than a , the Lyapunov function $V(x)$ is greater than b . Therefore, one can conclude that $V(x) > 0$ holds for all $x \neq 0$.
2. $V(0) = 0$: since $V(x) > 0$ holds for all $x \neq 0$, we can deduce that $V(0)$ must be equal to 0. Otherwise, if $V(0)$ is greater than 0, then there exists a small neighborhood where $V(x) > 0$. This contradicts the condition that $V(x) > 0$ holds for all $x \neq 0$.
3. $\dot{V} \leq 0$ holds for all $x \neq 0$: the derivative of the Lyapunov function $V(x)$ can represent the rate of change of the state of the system. Since $V(x)$ is asymptotically stable, by the definition of the Lyapunov function, for all x satisfying $\|x\| > a$, there is $\dot{V} \leq 0$. This implies that the Lyapunov function $V(x)$ is decreasing over the range of these x . The derivative of $V(x)$ holds for all $x \neq 0$: the derivative of $V(x)$ can represent the

rate of change of the state of the system. Also, by the definition of asymptotic stability, $\dot{V} \leq 0$ must tend to 0, i.e., $\dot{V}(0) = 0$.

In summary, if the Lyapunov function is asymptotically stable, we can obtain that $V(x) > 0$, $V(0) = 0$, and $\dot{V} \leq 0$.

2.4.2. Certificate of Sufficiency

Define the Lyapunov function [31] to be:

$$V = \frac{1}{2}s^T M_0 s + \frac{1}{2}tr(\tilde{W}^T \gamma^{-1} \tilde{W}) \tag{23}$$

Take the derivative of the Lyapunov function:

$$\begin{aligned} \dot{V} &= s^T M_0 \dot{s} + \frac{1}{2}s^T M_0 s + tr(\tilde{W}^T \gamma^{-1} \dot{\tilde{W}}) \\ &= s^T M_0 (\ddot{e} + \alpha \dot{e} + \beta \lambda |e|^{(\lambda-1)} \dot{e}) - tr(\tilde{W}^T \gamma^{-1} \dot{\tilde{W}}) \\ &= s^T M_0 (\ddot{q} - \ddot{q}_d + \alpha \dot{e} + \beta \lambda |e|^{(\lambda-1)} \dot{e}) - tr(\tilde{W}^T \gamma^{-1} \dot{\tilde{W}}) \end{aligned} \tag{24}$$

Substituting Equations (4), (19) and (22) gives:

$$\begin{aligned} \dot{V} &= -s^T M_0 [k_1 |s|^{a_1} sigmoid(s) + k_2 |s|^{a_2} tanh(s) + k_3 |s|^{a_3} f(s) + k_4 s] \\ &\quad + s^T [\varepsilon - \varepsilon_N sgn(s)] - tr \tilde{W}^T (\gamma^{-1} \dot{\tilde{W}} - s^T h(x)) \end{aligned} \tag{25}$$

From the above equation, $\dot{V} \leq 0$ holds all the time; if and only if $s = 0$, then $\dot{V} = 0$. According to the Lyapunov stability theorem, if $V > 0$ and $\dot{V} \leq 0$, it can be obtained that the system is asymptotically convergent under the Lyapunov condition.

2.5. Improved Salp Swarm Algorithm

The salp swarm algorithm (SSA) is a heuristic optimization algorithm inspired by the behavioral characteristics of a marine organism called salp. The algorithm simulates the movement and propagation of a salp swarm, where individuals update their positions and velocities through information exchange and cooperation to search for the optimal solution [32]. By mimicking the biological features and collaborative behavior of salp swarms, the SSA can effectively search for the global optimum in complex optimization problems. The basic process of the SSA [33] is as follows:

Step 1: Initialize the population. Initialize the positions of the salps based on the upper and lower limits of each dimension in the search space. The positions of the salps are initialized as:

$$X_j^i = rand(N, D) \times (ub_j - lb_j) + lb_j \tag{26}$$

where $X_j^i (i = 1, 2, \dots, N, j = 1, 2, \dots, D)$ is the position of the i -th salp in the j -th dimension, N is the population size of the salp, D is the spatial dimension, $rand(N, D)$ is a uniform random number in the range of $[N, D]$, and ub_j and lb_j are the upper and lower bounds on the search space in the j -th dimension.

Step 2: Calculate the fitness value of each salp according to the objective function.

Step 3: Determine the initial location of the food source. Rank the adaptation values of the salps, and the location of the optimal salp is the location of the food source.

Step 4: Identify leaders and followers. The first half of the salp chain are the leaders, and the rest are the followers.

Step 5: Update the position of the salp leader as follows:

$$X_j^1 = \begin{cases} F_j + c_1((ub_j - lb_j)c_2 + lb_j), c_3 \geq 0.5 \\ F_j - c_1((ub_j - lb_j)c_2 + lb_j), c_3 < 0.5 \end{cases} \quad (27)$$

where X_j^1 is the position of the first salp leader in the j -th dimension, F_j is the position of the food source in the j -th dimension, and c_2 and c_3 are uniformly distributed random numbers between 0 and 1. c_1 is adaptively decreasing with the number of iterations. The value of c_1 is taken as follows:

$$c_1 = 2e^{-(4t/T)^2} \quad (28)$$

where t is the current iteration number, and T is the maximum number of iterations.

Step 6: Update the positions of the salp followers as follows:

$$X_j^i = \frac{1}{2}(X_j^i + X_j^{i-1}) \quad (29)$$

where $j \geq 2$, X_j^i represents the position of the i -th salp follower in the j -th dimension search area.

Step 7: Apply boundary processing to each dimension of the updated individual, and update the location of the food source based on the new globally optimal salp location after the update.

Step 8: Determine whether the termination condition is satisfied; if so, output the result. Otherwise, proceed to step 4 and continue iterating.

The SSA is similar to other heuristic algorithms and may encounter the issue of local optima, along with certain limitations in terms of optimization accuracy and solution stability. To address these problems, this study improves the SSA and proposes an adaptive leader salp swarm algorithm (ALSSA).

By introducing an adaptive inertia weight factor, the influence of food sources on the leader position undergoes adaptive changes during the update process. As the number of iterations increases, the impact of food sources gradually decreases, thereby limiting the search range, improving the accuracy and efficiency of the search, and avoiding the unrestricted search range issue in the leader position update stage of SSA. The formula for calculating the inertia weight in this paper is as follows:

$$w = w_{\max} - (w_{\max} - w_{\min})(t/T)^2 \quad (30)$$

where w_{\max} and w_{\min} are the upper and lower limits of the inertia weight.

By incorporating the previous generation's salp leader position into the leader position update formula, the updated leader position is influenced not only by the previous generation's salp leader position but also by the previous generation's global best solution. This approach effectively avoids the problem of the basic algorithm getting trapped in local optima and improves the algorithm optimization accuracy. The improved salp leader position update formula is as follows:

$$X_j^i(t) = X_j^i(t-1) + (F_j(t-1) - X_j^i(t-1)) \cdot rand(0,1) \quad (31)$$

where $X_j^i(t-1)$ represents the position of the i -th salp leader in the j -th dimension of the previous generation, $F_j(t-1)$ represents the position of the food source in the previous generation, and $rand(0,1)$ denotes a uniform random number between 0 and 1.

Combining Equations (29) and (30), the new salp leader position update equation can be expressed as follows:

$$X_j^i(t) = X_j^i(t-1) + (w \cdot F_j(t-1) - X_j^i(t-1)) \cdot rand(0,1) \quad (32)$$

We introduce an adaptive adjustment strategy to dynamically decrease the number of salp leaders with an increasing iteration count while increasing the number of followers adaptively. This approach ensures a balance between global and local search throughout the entire phase, thereby improving the convergence accuracy of the algorithm. It effectively avoids the issues of early convergence to local optima and low optimization accuracy in the later stages. The updated formula for calculating the number of leaders and followers is as follows:

$$r = b_0 \left(\tan \left(-\frac{\pi t}{4T} + \frac{\pi}{4} \right) - k_0 \cdot \text{rand}(0, 1) \right) \quad (33)$$

where b_0 is the coefficient to control the ratio of leaders to followers, k_0 is the perturbation deviation factor, the number of leaders is equal to rN , and the number of followers is equal to $(1 - r)N$.

The objective function of ALSSA is used to minimize angular tracking errors, which is expressed as follows:

$$J = \frac{1}{n} \sum_{i=1}^n |e_i| \quad (34)$$

The specific meaning of this expression is as follows: for a given number of sampling points n , calculate the absolute values of the angular tracking errors $|e_i|$ at each sampling point, and then sum the absolute values of the angular tracking errors at all sampling points to obtain the value of the objective function.

In the designed controller, parameters that have a minimal impact on control performance are considered as priors and do not require adjustment. There are a total of six adjustable controller parameters, namely α , β , k_1 , k_2 , k_3 , k_4 . The precise values of these control parameters will be determined by the ALSSA method.

3. Simulation and Experimental Results

3.1. Simulation Results

To validate the effectiveness and superiority of the proposed neural network terminal sliding mode control algorithm for the robotic arms, based on a novel reaching law and an improved salp swarm algorithm, this study conducted simulation experiments using the ABB IRB120 robotic arm as the research subject in Matlab R2022b software.

The ABB IRB120 robot is a six-axis robotic arm developed by ABB, a leading robotics company. It is designed for various industrial applications, including assembly, material handling, and machine tending. The IRB120 robot offers high precision, flexibility, and compactness, making it suitable for use in small workspaces. It has a payload capacity of up to 3 kg and a reach of 580 mm, allowing it to perform tasks with precision and agility. The IRB120 robot is equipped with advanced features such as integrated vision systems and user-friendly programming interfaces, enabling easy integration and efficient operation in industrial environments. The ABB IRB120 robot is shown in Figure 1.

Denavit–Hartenberg (DH) parameters and mass parameters play crucial roles in the modeling of robotic arms, as they are essential for accurately describing the kinematic and dynamic characteristics of the arm. DH parameters describe the geometric and kinematic properties of the robotic arm, providing the geometric relationships between its joints and enabling the precise execution of desired motions. On the other hand, mass parameters describe the distribution of mass within the robotic arm, influencing its inertia properties and dynamic response, which are vital for achieving precise motion control. The DH parameters and quality parameters of the ABB IRB120 robot are shown in Tables 1 and 2.

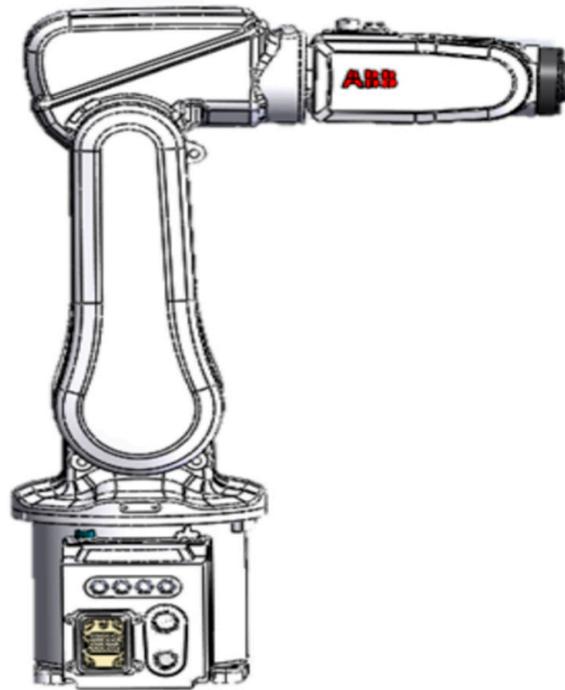


Figure 1. ABB IRB120 robot.

Table 1. The DH parameter list of the ABB IRB120 robot.

Joint	Angle θ ($^\circ$)	Offset d (m)	Length a (m)	Twist α ($^\circ$)
Joint 1	θ_1	0.290	0	-90
Joint 2	θ_2	0	0.270	0
Joint 3	θ_3	0	0.070	-90
Joint 4	θ_4	0.302	0	90
Joint 5	θ_5	0	0	-90
Joint 6	θ_6	0.072	0	0

Table 2. The quality parameter list of the ABB IRB120 robot.

Joint	Mass (kg)	Position of the Center of Mass (m)
Joint 1	9.28	($-0.02819, 0.00002, 0.13210$)
Joint 2	3.91	($-0.00216, 0.00118, 0.39124$)
Joint 3	2.94	($0.00178, -0.01867, 0.61730$)
Joint 4	1.33	($0.00856, -0.22070, 0.62499$)
Joint 5	0.55	($0.01133, -0.29682, 0.62287$)
Joint 6	0.01	($0.01367, -0.36273, 0.61955$)

Since the inertia matrix $M(q)$, the Coriolis and centrifugal matrix $C(q, \dot{q})$, and the gravitational matrix $G(q)$ of the ABB IRB120 six-axis robotic arm are too complicated, the detailed dynamics parameter matrices are not listed here. The model determination section is set to $M_0 = 0.8M$, $C_0 = 0.8C$, $G_0 = 0.8G$. The initial angles of the ABB robotic arm are $q_1(0) = q_2(0) = q_3(0) = q_4(0) = q_5(0) = q_6(0) = 0.05$. The reference trajectories are $q_{d1} = q_{d2} = q_{d3} = q_{d4} = q_{d5} = q_{d6} = 0.1sint$. The parameters of the sliding mode control are $\alpha = 2$, $\beta = 0.4$, $\lambda = 1.5$, $k_1 = k_2 = k_3 = k_4 = 0.5$, $a_1 = 1.2$, $a_2 = 0.2$. The joint friction is $F(\dot{q}) = 0.1\dot{q} + 0.1\text{sgn}(\dot{q})$. The external disturbance is $\tau_d = 0.5sint$. The parameter of the robust term is $\varepsilon_N = 0.1$. The parameters of the RBF neural network are $\gamma = 0.01$, $b = 10$, $c = [-1 \ -0.5 \ -0.25 \ -0.125 \ -0.0625 \ -0.03125 \ 0 \ 0.03125 \ 0.0625 \ 0.125 \ 0.25 \ 0.5 \ 1]$. The parameters of the ALSSA are $N = 30$, $T = 500$, $w_{\max} = 0.9$, $w_{\min} = 0.4$, $b_0 = 0.75$, $k_0 = 0.2$. The number of iterations to obtain the optimal solution is 29, and the optimized

sliding mode control parameters are $\alpha = 2.7$, $\beta = 1.8$, $k_1 = 0.1$, $k_2 = 0.8$, $k_3 = 1.4$, $k_4 = 1.5$.

The ALSSA-RBFTSM algorithm designed in this paper is compared with the global fast terminal sliding mode (GFTSM) algorithm proposed in the literature [34] and the RBF neural network fast nonsingular terminal sliding mode (RBF-FNTSM) algorithm proposed in the literature [35]. From Figure 2, it can be seen that the GFTSM algorithm has the smallest initial control torques, but it experiences the most serious control torque jitter. The RBF-FNTSM algorithm effectively suppresses control torque jitter, but there is still a certain amount of jitter that remains. At the same time, its initial control torques increase. The ALSSA-RBFTSM algorithm demonstrates smooth control torques with little jitter, but it has the largest initial control torques. From Figure 3, it can be observed that all three control algorithms enable the robotic arm to track the desired trajectory within a certain time period. However, the GFTSM algorithm deviates from the desired trajectory at certain time intervals, while the RBF-FNTSM algorithm and the proposed algorithm in this paper remain mostly on the desired trajectory. Figure 4 demonstrates that when subjected to uncertainties such as friction and external disturbances, the position tracking errors of the proposed algorithm remain consistently at zero. In contrast, the GFTSM algorithm and the RBF-FNTSM algorithm exhibit fluctuating position tracking errors. The GFTSM algorithm is most affected, exhibiting significant fluctuations in position tracking error and poor position tracking performance. The RBF-FNTSM algorithm demonstrates a certain level of disturbance rejection capability, suppressing uncertainties to some extent, but still experiences a certain degree of position tracking error fluctuations. In comparison to the GFTSM and RBF-FNTSM algorithms, the ALSSA-RBFTSM algorithm exhibits improved performance, demonstrating better robustness and position tracking effectiveness when faced with uncertainties such as friction and external disturbances.

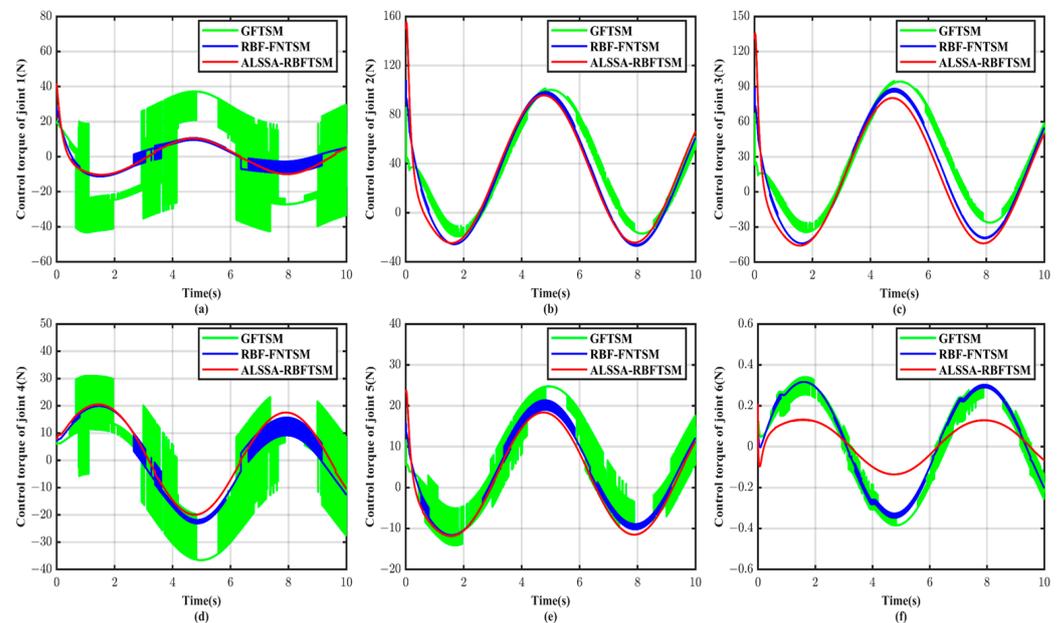


Figure 2. Control torques of different controllers. (a) Joint 1; (b) Joint 2; (c) Joint 3; (d) Joint 4; (e) Joint 5; (f) Joint 6.

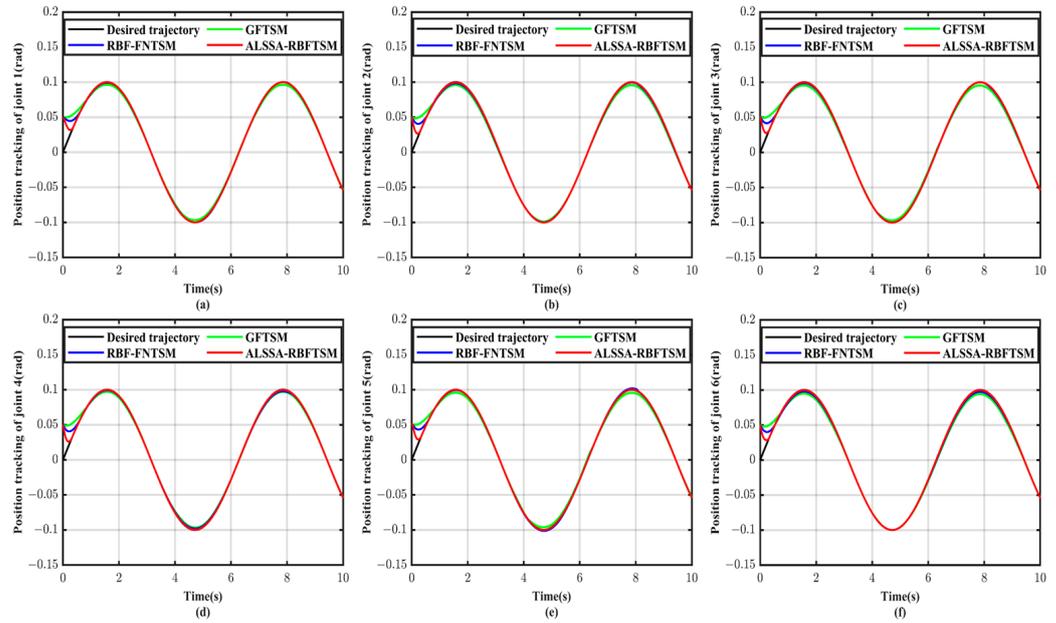


Figure 3. Position tracking of different controllers. (a) Joint 1; (b) Joint 2; (c) Joint 3; (d) Joint 4; (e) Joint 5; (f) Joint 6.

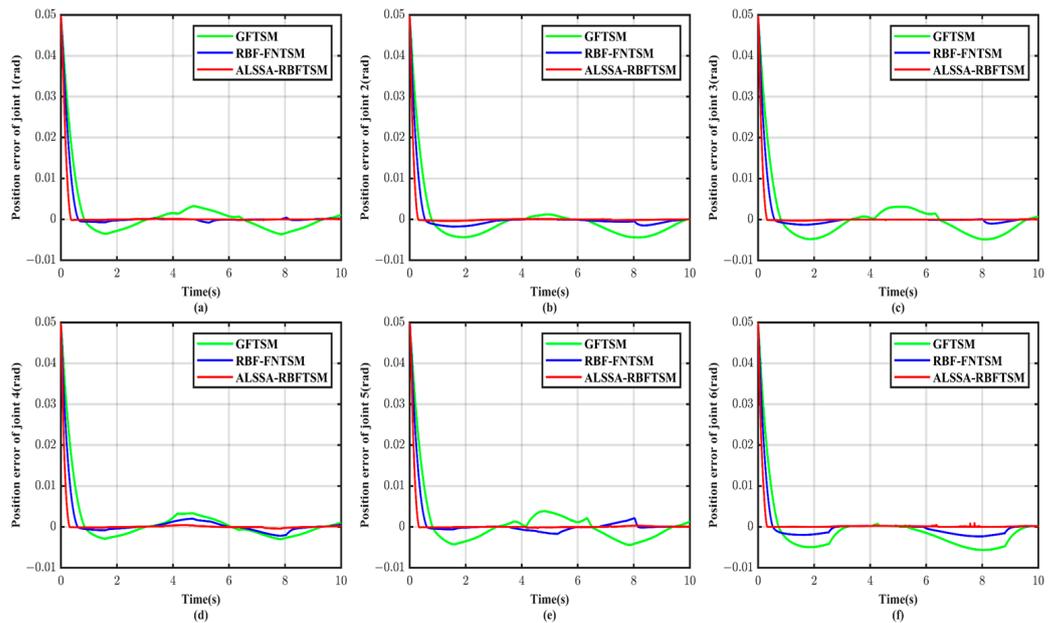


Figure 4. Position tracking errors of different controllers. (a) Joint 1; (b) Joint 2; (c) Joint 3; (d) Joint 4; (e) Joint 5; (f) Joint 6.

Figure 5 demonstrates that the velocities of all three control algorithms can effectively track the desired trajectories. The GFTSM algorithm exhibits the longest convergence time, while the RBF-FNTSM algorithm shows a slightly accelerated convergence speed. The ALSSA-RBFTSM control algorithm designed in this study demonstrates the shortest convergence time. Figure 6 shows that the velocity tracking errors of all three control algorithms converge to nearly zero, indicating a favorable control performance. The GFTSM algorithm has relatively significant fluctuations in trajectory tracking errors. The error fluctuations in the RBF-FNTSM algorithm are less noticeable. In contrast, the ALSSA-RBFTSM control algorithm designed in this study demonstrates the fastest convergence speed and the smallest tracking errors, showcasing superior steady-state characteristics.

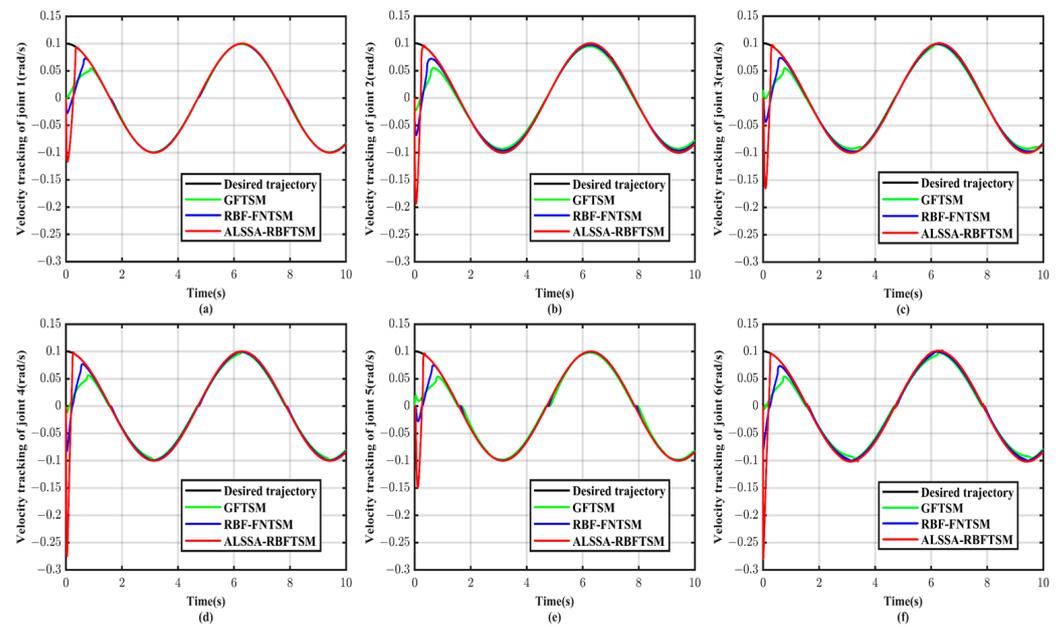


Figure 5. Velocity tracking of different controllers. (a) Joint 1; (b) Joint 2; (c) Joint 3; (d) Joint 4; (e) Joint 5; (f) Joint 6.

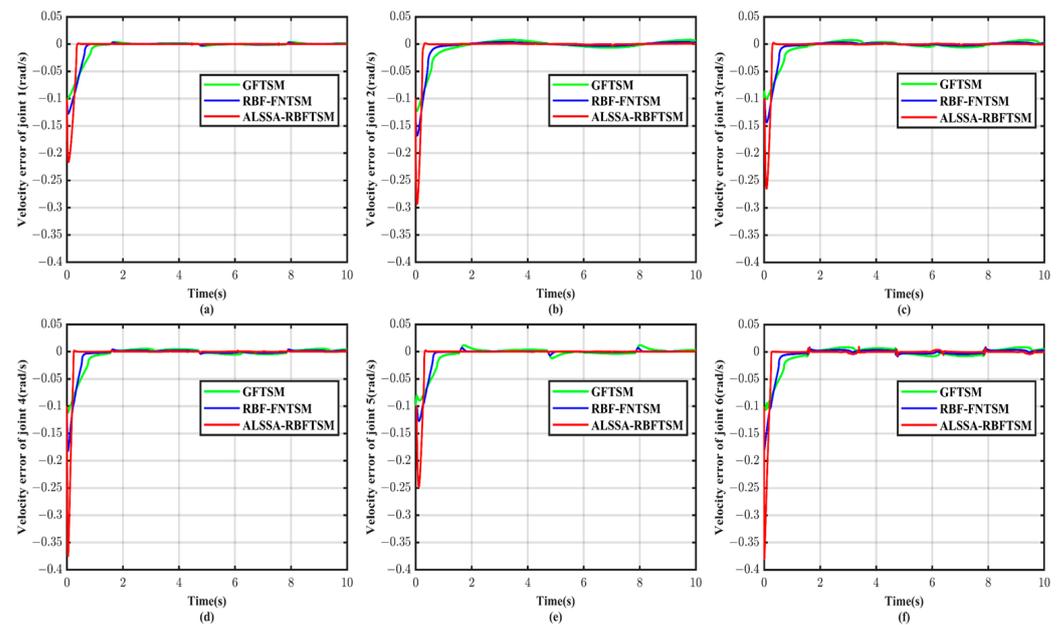


Figure 6. Velocity tracking errors of different controllers. (a) Joint 1; (b) Joint 2; (c) Joint 3; (d) Joint 4; (e) Joint 5; (f) Joint 6.

According to the results presented in Tables 3 and 4, it can be observed that the ALSSA-RBFTSM algorithm designed in this study exhibits the best control performance. It demonstrates the smallest average steady-state error and maximum steady-state error in position tracking for each joint of the robotic arm. Comparative analysis reveals that, compared to the GFTSM control algorithm, the ALSSA-RBFTSM control algorithm reduces the average steady-state error in position tracking for each joint by 97.6%, 96.5%, 99.1%, 90.7%, 95.1%, and 95.3% respectively. Furthermore, it reduces the maximum steady-state error in position tracking for each joint by 95.5%, 95.0%, 97.0%, 86.5%, 92.7%, and 83.9%, respectively. In comparison to the RBF-FNTSM control algorithm, the ALSSA-RBFTSM control algorithm reduces the average steady-state error in position tracking for each joint

by 68.5%, 84.2%, 85.7%, 82.6%, 82.3%, and 87.3%, respectively. Additionally, it reduces the maximum steady-state error in position tracking for each joint by 79.4%, 85.3%, 85.5%, 79.2%, 84.8%, and 60.9%, respectively.

Table 3. Position tracking average steady-state error.

Joint	GFTSM	RBF-FNTSM	ALSSA-RBFTSM
Joint 1	1.75×10^{-3}	1.31×10^{-4}	4.13×10^{-5}
Joint 2	1.83×10^{-3}	4.10×10^{-4}	6.48×10^{-5}
Joint 3	2.38×10^{-3}	1.53×10^{-4}	2.19×10^{-5}
Joint 4	1.68×10^{-3}	9.02×10^{-4}	1.57×10^{-4}
Joint 5	2.79×10^{-3}	6.03×10^{-4}	1.07×10^{-4}
Joint 6	2.49×10^{-3}	9.10×10^{-4}	1.16×10^{-4}

Table 4. Position tracking maximum steady-state error.

Joint	GFTSM	RBF-FNTSM	ALSSA-RBFTSM
Joint 1	3.65×10^{-3}	8.07×10^{-4}	1.66×10^{-4}
Joint 2	4.43×10^{-3}	1.50×10^{-3}	2.21×10^{-4}
Joint 3	4.84×10^{-3}	9.98×10^{-4}	1.45×10^{-4}
Joint 4	3.33×10^{-3}	2.16×10^{-3}	4.49×10^{-4}
Joint 5	4.42×10^{-3}	2.12×10^{-3}	3.22×10^{-4}
Joint 6	5.65×10^{-3}	2.33×10^{-3}	9.12×10^{-4}

3.2. Experiment Results

In this section, we implement the proposed control strategy on the packaging unit of an ABB robot in the intelligent manufacturing production line in our laboratory. The intelligent manufacturing production line comprises six experimental stations: unordered screening unit, visual sorting unit, dimension detection unit, laser marking unit, assembly unit, and packaging unit. These stations utilize automation equipment and robots to carry out production tasks, reducing the requirement for manual operations and enhancing production efficiency. The ALSSA-RBFTSM algorithm designed in this paper is applied to the ABB six-axis robotic arm in the packing unit of the production line to design the robot grasping and placing test. Figures 7 and 8 illustrates the process of robot grasping and placing.



Figure 7. Trajectory tracking control for robotic arm gripping tests.



Figure 8. Trajectory tracking control for robotic arm placement tests.

The system conducted 50 grasping and placing experiments on black boxes. The success rate for grasping was 92%, while the success rate for placing was 86%. The main reason for the grasping failures is the positional deviation of the black boxes on the tray, which caused the robotic arm to misalign or have unstable grasping. The primary reason for the placing failures is the angular deviation of the lid of the yellow paper box, resulting in interference with the black boxes during the placing process. The experimental results above demonstrate that, for multiple grasping and placing tasks, the ABB robot can reliably track the trajectory, successfully completing the grasping and placing operations. The system operates smoothly during the workflow, without significant occurrences of excessive vibrations.

4. Conclusions

This paper proposes a neural network terminal sliding mode control algorithm for robotic arms, based on a reaching law and an improved salp swarm algorithm. A fast nonsingular terminal sliding surface is utilized to achieve finite-time convergence, and the traditional multi-power reaching law is improved to reduce system oscillations. An RBF neural network is employed to estimate the model uncertainties and external disturbances of the system. The improved salp swarm algorithm is utilized to optimize controller parameters, further enhancing control performance. A robust term is designed to compensate for estimation errors in the RBF neural network. The stability of the system is proven, using the Lyapunov stability theory. Simulation results demonstrate that, compared to the GFTSM control algorithm, the ALSSA-RBFTSM control algorithm reduces the average steady-state position tracking error of the robotic arm by up to 99.1% and the maximum steady-state position tracking error by up to 97.0%. Compared to the RBF-FNTSM control algorithm, the ALSSA-RBFTSM control algorithm reduces the average steady-state position tracking error by up to 87.3% and the maximum steady-state position tracking error by up to 85.5%. The experimental results show that the designed control algorithm enables the robot to stably track trajectories and complete grasping and placing tasks. The simulation and experimental results demonstrate the effectiveness and reliability of the proposed control algorithm, showcasing its excellent disturbance rejection capability and trajectory tracking performance.

Author Contributions: Conceptualization, H.Z. and Q.Z.; methodology, H.Z. and J.D.; software, H.Z.; validation, H.Z., J.D. and J.Q.; writing—original draft preparation, H.Z.; writing—review and editing, H.Z., J.D. and J.Q. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data used to support the findings of this study are included within the article and are also available from the corresponding authors upon request.

Acknowledgments: We thank Haodong Li, Lei Yang, and Shanlin Li of Shanghai Maritime University for their valuable discussions and useful suggestions in this work.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yang, Q.; Ma, X.; Wang, W.; Peng, D. Adaptive Non-Singular Fast Terminal Sliding Mode Trajectory Tracking Control for Robot Manipulators. *Electronics* **2022**, *11*, 3672. [[CrossRef](#)]
2. Kelly, R.; Haber, R.; Haber-Guerra, R.E.; Reyes, F. Lyapunov Stable Control of Robot Manipulators: A Fuzzy Self-Tuning Procedure. *Intell. Autom. Soft Comput.* **1999**, *5*, 313–326. [[CrossRef](#)]
3. Guerra, R.E.H.; Schmitt-Braess, G.; Haber, R.H.; Alique, A.; Alique, J.R. Using circle criteria for verifying asymptotic stability in PI-like fuzzy control systems: Application to the milling process. *IEE Proc.-Control Theory Appl.* **2003**, *150*, 619–627. [[CrossRef](#)]
4. Lyu, X.; Lin, Z. Design of PID control for planar uncertain nonlinear systems with input delay. *Int. J. Robust Nonlinear Control* **2022**, *32*, 9407–9420. [[CrossRef](#)]
5. Burkan, R.; Uzmay, İ. Logarithmic based robust approach to parametric uncertainty for control of robot manipulators. *Int. J. Robust Nonlinear Control* **2005**, *15*, 427–436. [[CrossRef](#)]
6. Wei, B. Adaptive Control Design and Stability Analysis of Robotic Manipulators. *Actuators* **2018**, *7*, 89. [[CrossRef](#)]
7. Raković, M.; Anil, G.; Mihajlović, Ž.; Srdjan, S.; Naik, S.; Borovac, B.; Gottscheber, A. Fuzzy position-velocity control of underactuated finger of FTN robot hand. *J. Intell. Fuzzy. Syst.* **2018**, *34*, 2723–2736. [[CrossRef](#)]
8. Yovchev, K.; Delchev, K.; Krastev, E. State Space Constrained Iterative Learning Control for Robotic Manipulators. *Asian J. Control* **2018**, *20*, 1145–1150. [[CrossRef](#)]
9. Xu, K.; Wang, Z. The design of a neural network-based adaptive control method for robotic arm trajectory tracking. *Neural Comput. Appl.* **2022**, *35*, 8785–8795. [[CrossRef](#)]
10. Wu, H.; Zhang, X.; Song, L.; Zhang, Y.; Wang, C.; Zhao, X.; Gu, L. Parallel Network-Based Sliding Mode Tracking Control for Robotic Manipulators with Uncertain Dynamics. *Actuators* **2023**, *12*, 187. [[CrossRef](#)]
11. Guo, Q.; Sun, P.; Yin, J.M.; Yu, T.; Jiang, D. Parametric adaptive estimation and backstepping control of electro-hydraulic actuator with decayed memory filter. *ISA Trans.* **2016**, *62*, 202–214. [[CrossRef](#)] [[PubMed](#)]
12. Zhao, D.; Li, S.; Gao, F. A new terminal sliding mode control for robotic manipulators. *Int. J. Control* **2009**, *82*, 1804–1813. [[CrossRef](#)]
13. Doan, Q.V.; Vo, A.T.; Le, T.D.; Kang, H.J.; Nguyen, N.H.A. A Novel Fast Terminal Sliding Mode Tracking Control Methodology for Robot Manipulators. *Appl. Sci.* **2020**, *10*, 3010. [[CrossRef](#)]
14. Jin, M.; Lee, J.; Chang, P.; Choi, C. Practical Nonsingular Terminal Sliding-Mode Control of Robot Manipulators for High-Accuracy Tracking Control. *IEEE Trans. Ind. Electron.* **2009**, *56*, 3593–3601. [[CrossRef](#)]
15. Liang, X.; Wang, H.; Zhang, Y. Adaptive nonsingular terminal sliding mode control for rehabilitation robots. *Comput. Electr. Eng.* **2022**, *99*, 107718. [[CrossRef](#)]
16. Wang, A.; Jia, X.; Dong, S. A New Exponential Reaching Law of Sliding Mode Control to Improve Performance of Permanent Magnet Synchronous Motor. *IEEE Trans. Magn.* **2013**, *49*, 2409–2412. [[CrossRef](#)]
17. Xia, Y.; Xie, W.; Ma, J. Research on trajectory tracking control of manipulator based on modified terminal sliding mode with double power reaching law. *Int. J. Adv. Robot. Syst.* **2019**, *16*, 1729881419847899. [[CrossRef](#)]
18. Ba, K.; Yu, B.; Liu, Y.; Jin, Z.; Gao, Z.; Zhang, J.; Kong, X. Fuzzy Terminal Sliding Mode Control with Compound Reaching Law and Time Delay Estimation for HDU of Legged Robot. *Complexity* **2020**, *2020*, 5240247. [[CrossRef](#)]
19. Zhang, Y.; Wang, Y.J.; Yu, J.Q. A Novel MPPT Algorithm for Photovoltaic Systems Based on Improved Sliding Mode Control. *Electronics* **2022**, *11*, 2421. [[CrossRef](#)]
20. Cruz, Y.J.; Rivas, M.; Quiza, R.; Villalonga, A.; Haber, R.E.; Beruvides, G. Ensemble of convolutional neural networks based on an evolutionary algorithm applied to an industrial welding process. *Comput. Ind.* **2021**, *133*, 103530. [[CrossRef](#)]
21. Liu, A.; Zhang, Y.; Zhao, H.; Wang, S.; Sun, D. Neural network control system of cooperative robot based on genetic algorithms. *Neural Comput. Appl.* **2020**, *33*, 8217–8226. [[CrossRef](#)]
22. He, W.; Dong, Y. Adaptive Fuzzy Neural Network Control for a Constrained Robot Using Impedance Learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 1174–1186. [[CrossRef](#)] [[PubMed](#)]
23. Tlijani, H.; Jouila, A.; Nouri, K. Wavelet neural network sliding mode control of two rigid joint robot manipulator. *Adv. Mech. Eng.* **2022**, *14*, 16878132221119886. [[CrossRef](#)]
24. Sun, C.; He, W.; Ge, W.; Chang, C. Adaptive Neural Network Control of Biped Robots. *IEEE Trans. Syst. Man Cybern. Syst.* **2016**, *47*, 315–326. [[CrossRef](#)]
25. Fan, B.; Zhang, Y.; Chen, Y.; Meng, L. Intelligent vehicle lateral control based on radial basis function neural network sliding mode controller. *CAAI Trans. Intell. Technol.* **2022**, *7*, 455–468. [[CrossRef](#)]
26. Chen, Z.; Yang, X.; Liu, X. RBFNN-based nonsingular fast terminal sliding mode control for robotic manipulators including actuator dynamics. *Neurocomputing* **2019**, *362*, 72–82. [[CrossRef](#)]
27. Ahmed, S.; Azar, A.T.; Tounsi, M. Adaptive Fault Tolerant Non-Singular Sliding Mode Control for Robotic Manipulators Based on Fixed-Time Control Law. *Actuators* **2022**, *11*, 353. [[CrossRef](#)]
28. Zhang, Z.; Zhang, K.; Han, Z. Three-dimensional nonlinear trajectory tracking control based on adaptive sliding mode. *Aerosp. Sci. Technol.* **2022**, *128*, 107734. [[CrossRef](#)]

29. Chen, Y.; Tan, R.; Zheng, Y.; Zhou, Z. Sliding-Mode Control With Multipower Approaching Law for DC-Link Voltage of Z-Source Photovoltaic Inverters. *IEEE Access* **2019**, *7*, 133812–133821. [[CrossRef](#)]
30. Qu, C.; Hu, Y.; Guo, Z.; Han, F.; Wang, X. New Sliding Mode Control Based on Tracking Differentiator and RBF Neural Network. *Electronics* **2022**, *11*, 3135. [[CrossRef](#)]
31. Pham, P.C.; Kuo, Y.L. Robust Adaptive Finite-Time Synergetic Tracking Control of Delta Robot Based on Radial Basis Function Neural Networks. *Appl. Sci.* **2022**, *12*, 10861. [[CrossRef](#)]
32. Khajezadeh, M.; Iraj, A.; Majdi, A.; Keawsawasvong, S.; Nehdi, M.L. Adaptive Salp Swarm Algorithm for Optimization of Geotechnical Structures. *Appl. Sci.* **2022**, *12*, 6749. [[CrossRef](#)]
33. Ding, L.; Ma, R.; Wu, Z.; Qi, R.; Ruan, W. Optimal Joint Space Control of a Cable-Driven Aerial Manipulator. *Comput. Model. Eng. Sci.* **2023**, *135*, 441–464. [[CrossRef](#)]
34. Fang, Q.; Mao, P.; Shen, L.; Wang, J. A global fast terminal sliding mode control for trajectory tracking of un-manned aerial manipulation. *Meas. Control* **2022**, *56*, 763–776. [[CrossRef](#)]
35. Guo, L.; Liu, W.; Li, L.; Lou, Y.; Wang, X.; Liu, Z. Neural Network Non-Singular Terminal Sliding Mode Control for Target Tracking of Underactuated Underwater Robots with Prescribed Performance. *J. Mar. Sci. Eng.* **2022**, *10*, 252. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.