

Article

Nonlinear Nonsingular Fast Terminal Sliding Mode Control Using Deep Deterministic Policy Gradient

Zefeng Xu ^{1,*}, Wenkai Huang ^{1,*} , Zexuan Li ², Linkai Hu ¹ and Puwei Lu ¹

¹ School of Mechanical and Electrical Engineering, Guangzhou University, Guangzhou 510006, China; 2111807041@e.gzhu.edu.cn (Z.X.); 1707700071@e.gzhu.edu.cn (L.H.); 2111907042@e.gzhu.edu.cn (P.L.)

² School of Electronics and Communication Engineering, Guangzhou University, Guangzhou 510006, China; lzx@e.gzhu.edu.cn

* Correspondence: smallkat@gzhu.edu.cn

Featured Application: The control strategy proposed to this paper can be applied to the joint position and velocity tracking down industrial robots (series or parallel manipulators). Theoretically, it is suitable for general second-order nonlinear systems, such as inverted pendulum control, motor coupling control, dual manipulator cooperative control, etc.

Abstract: Background: As a control strategy of industrial robots, sliding mode control has the advantages of fast response and simple physical implementation, but it still has the problems of chattering and low tracking accuracy caused by chattering. This paper proposes a new sliding mode control strategy for the application of industrial robot control, which effectively solves these problems. Methods: In this paper, a deep deterministic policy gradient–nonlinear nonsingular fast terminal sliding mode control (DDPG–NNFTSMC) strategy is proposed for industrial robot control. In order to improve the tracking control accuracy and anti-interference ability, DDPG is used to approach the uncertainties of the system in real time, which ensures the robustness of the system in various uncertain environments. Lyapunov function is used to prove the stability and finite time convergence of the system. Compared with the nonsingular terminal sliding mode control (NTSMC), the time to reach the equilibrium point is shorter. With the help of MATLAB/Simulink, the tracking accuracy and control effects are compared with traditional terminal sliding mode control (TSMC), NTSMC and radial basis function–sliding mode control (RBF–SMC), the results showed that it had the advantages of nonsingularity, finite time convergence, small tracking error. The motion accuracy and anti-interference ability of the uncertain manipulator system was further improved, and the chattering problem of the system in the motion process is effectively eliminated.

Keywords: nonlinear nonsingular fast terminal sliding mode control; deep deterministic policy gradient; dynamic uncertainty; disturbance; manipulator



Citation: Xu, Z.; Huang, W.; Li, Z.; Hu, L.; Lu, P. Nonlinear Nonsingular Fast Terminal Sliding Mode Control Using Deep Deterministic Policy Gradient. *Appl. Sci.* **2021**, *11*, 4685. <https://doi.org/10.3390/app11104685>

Academic Editor: Luigi Fortuna

Received: 20 April 2021

Accepted: 17 May 2021

Published: 20 May 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 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/>).

1. Introduction

In recent years, with the development of industrial robots, nonlinear, external interference, a variety of uncertainty problems appear, and the performance requirements of the control system are more and more strict. At present, there are many control methods of general nonlinear systems—e.g., adaptive control, fuzzy control, neural network control, sliding mode control (SMC) [1–4], etc. Among them, in the dynamic process, sliding mode control (SMC) is subject to the continuous changes of the system according to the current state of the system, which forces the system to move according to the state trajectory of the predetermined sliding mode, the hypersurface in state space is defined as sliding surface [5], which has strong robustness to the external interference and uncertainty of the nonlinear system [6,7]. Therefore, SMC is usually used in the control system of industrial robots and manipulators [8–11]. However, the traditional sliding mode variable structure controls have some disadvantages, such as singularity, uncertain convergence time,

and the error is difficult to converge on 0. When the state trajectory reaches the sliding surface, it is difficult to slide strictly into the sliding surface to the equilibrium point, it passes back and forth on both sides of the sliding surface, which is called chattering [5,12], and it needs an accurate dynamic model for the control object. To solve these problems, some research effects propose a series of solutions to eliminate the uncertainty, chatter, and error of the system as much as possible [13–21]. For the uncertain dynamic model, neural network [13–15], fuzzy logic system [16–18], RBF [19,20] are used to approach the uncertainty infinitely. In [21,22], the TSMC is introduced, and the nonlinear function is led into the sliding surface to construct the terminal sliding surface, so that the sliding surface's tracking error converges on 0 in finite time T . However, the convergence speed of TSMC is slow, and the singularity still exists.

Therefore, the authors of [23,24] proposed fast terminal sliding mode control (FTSMC) to solve the convergence rate problem, while [25,26] proposed NTSMC to solve the singularity problem. On these bases, the authors of [9,27] proposed nonsingular fast terminal sliding mode control (NFTSMC), which overcomes the single problem that [21–26] proposed a solution only for one of the shortcomings of SMC. However, the above TSMC, FTSMC and NFTSMC have not eliminated the chattering of the system. Therefore, some scholars proposed global terminal sliding mode control technology [24,28]. The control law is continuous and does not contain switching term (control transformation parameter) [5,12], so the chattering of the system is eliminated, but the tracking accuracy is less precise.

Deep reinforcement learning has been developed rapidly and applied to the field of control in recent years. For example, in [29,30] the authors applied the reinforcement learning neural network algorithm to the control of flexible manipulator and multi-body system with unknown dynamic model, inspired by [29,30], this paper combined deep reinforcement learning algorithm with SMC, proposed DDPG–NNFTSMC. It had the following characteristics:

- It is proved that NNFTSMC has the characteristics of nonsingularity and finite time convergence by mathematical derivation, and the robustness and stability of the system are verified by Lyapunov theorem;
- DDPG is used to adaptively approximate the uncertainty of the system model, and the chattering is eliminated to realize the control system's smooth input, to improve the anti-interference ability which ensures the robustness of the system in various environments such as quality change, friction factor, external disturbance, and modeling uncertainty of the system [5,6,12];
- On the basis of eliminating chattering with DDPG, it brings the advantages of low steady-state error and high-precision position tracking.

Compared with the nonlinear control methods such as TSMC, NTSMC and RBF–SMC, the proposed method has better tracked performance and stronger anti-interference and uncertainty effects. The effectiveness and superiority of the proposed control method is verified. Finally, the conclusion is given. Table 1 provides the definition of acronyms.

Table 1. Definition of acronyms.

Acronyms	Definition
SMC	Sliding mode control
TSMC	Terminal sliding mode control
FTSMC	Fast terminal sliding mode control
NTSMC	Nonsingular terminal sliding mode control
RBF–SMC	Radial basis function–sliding mode control
DDPG–NNTFSMC	Deep deterministic policy gradient–nonlinear nonsingular fast terminal sliding mode control

2. Manipulator Model

According to [5,22,31], the dynamic differential equation model of n-DOF manipulator system is as follows:

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

where: $q, \dot{q}, \ddot{q} \in \mathbb{R}^n$ correspond to the position, velocity, and acceleration of the manipulator, $M(q) = \tilde{M}(q) + \delta M(q) \in \mathbb{R}^{n \times n}$ is the actual inertia matrix of order $n \times n$, $C(q, \dot{q}) = \tilde{C}(q, \dot{q}) + \delta C(q, \dot{q}) \in \mathbb{R}^{n \times 1}$ is the $n \times 1$ order inertia matrix of centrifugal force and Coriolis force, $\tilde{M}(q)$, $\tilde{C}(q, \dot{q})$ is the standard model, $\delta M(q)$, $\delta C(q, \dot{q})$ is the error of the real dynamic model. $G(q) \in \mathbb{R}^{n \times 1}$ is the inertia vector of order $n \times 1$ and represents the gravity matrix; $F(\dot{q})$ is the $n \times 1$ order inertia vector, which represents the friction force and disturbance load; $\tau_d(t) \in \mathbb{R}^{n \times 1}$ is interference term and uncertainty term; $\tau(t)$ is the system control input.

The actual dynamic equation of the manipulator can be written as follows:

$$\tilde{M}(q)\ddot{q} + \tilde{C}(q, \dot{q})\dot{q} + D = \tau(t) \quad (2)$$

where:

$$D(q, \dot{q}, \ddot{q}) = \delta M(q)\ddot{q} + \delta C(q, \dot{q})\dot{q} + G(q) + F(\dot{q}) + \tau_d(t) \quad (3)$$

The change of joint rotation, angular velocity and angular acceleration are defined as: $q_d, \dot{q}_d, \ddot{q}_d$, then the tracking position error of the system is:

$$e = q(t) - q_d(t) \quad (4)$$

The speed error is:

$$\dot{e} = \dot{q}(t) - \dot{q}_d(t) \quad (5)$$

The dynamic error corresponding to Equation (2) is:

$$\begin{cases} \dot{e}_1 = e_2 \\ \dot{e}_2 = -\tilde{M}(q)^{-1}\tilde{C}(q, \dot{q})\dot{q} + \tilde{M}(q)^{-1}\tau + D - \ddot{q}_d \end{cases} \quad (6)$$

where: $D = -d\tilde{M}(q)^{-1}$ is the vector of uncertainty (including unknown disturbance, uncertainty, and approximation error).

According to the physical characteristics of industrial robot [9,31,32], the following hypotheses are put forward:

Assumption 1. $M(q)$ is a positive definite, invertible symmetric matrix and bounded:

$$\phi_1 \leq M(q) \leq \phi_2 \quad (7)$$

Assumption 2. The uncertainty D in the set is a bounded function satisfying the constraint:

$$\|D\| \leq A_0 + A_1\|q\| + A_2\|\dot{q}\| \quad (8)$$

where: $\phi_1, \phi_2, A_0, A_1, A_2$ are unknown constants, and $\phi_1, \phi_2, \Lambda > 0$; $\|D\|$ represents the Euclidean norm of a matrix.

In this paper, the control strategy is proposed to improve the tracking control accuracy of the manipulator with uncertain dynamic models further. The nonlinear sliding surface is established by using dynamic error, then the feedback control loop is developed. DDPG algorithm is used to adaptively approximate the uncertainties of the system. Therefore, for the control system with uncertainties, the tracking error can converge to zero synchronously and remain stable for a finite amount of time.

3. DDPG–NNFTSMC Control Design

In this part, a DDPG–NNFTSMC control method is proposed for the nonlinear second-order system of the general manipulator. Then the sliding surface and DDPG algorithm design are given.

3.1. Design of the NNFTSMC

In this part, we design a new NNFTSMC sliding surface for the manipulator with uncertain dynamic model based on traditional NTSMC, and then solve the reaching law and design the controller based on the sliding surface.

According to [33], the function $\text{sig}(x)^a$ is introduced, when $a > 0, \forall x \in \mathbb{R}, \text{sig}(x)^a$ monotonically increasing, and always returns a real number.

$$\text{sig}(x)^a = |x|^a \text{sgn}(x) \tag{9}$$

It can be seen from [5] that the traditional ntsmc sliding surface and control rate are as follows:

$$s = x_1 + \frac{1}{\beta} x_2^\eta \tag{10}$$

where: $\beta > 0, 2 > \eta > 1$, because the exponential term of e_2 is greater than 0, the singular problem is avoided, but in the region far away from the equilibrium point, the state derivative of the system is smaller than the linear sliding surface with the same parameters, which affects the convergence rate of the system state. To accelerate the convergence speed, the tracking position error and change rate of Equations (4) and (5) are defined as NNFTSMC variables, combined with Equations (9) and (10), the sliding surface function is designed as follows:

$$s = e_1 + \frac{1}{\alpha} \text{sig}(e_1)^\gamma + \frac{1}{\beta} \text{sig}(e_2)^\eta \tag{11}$$

where: $\alpha, \beta > 0, 2 > \eta > 1$.

According to the dynamic error Equation (6), the NNFTSMC control law is designed as:

$$\tau = -\tilde{M}(q) \left[-\tilde{M}(q)^{-1} \tilde{C}(q, \dot{q}) \dot{q} + \frac{\gamma}{\alpha} \text{sig}(e_1)^{\gamma-1} + \frac{\beta}{\eta} \text{sig}(e_2)^{2-\eta} \left(1 + \frac{\gamma}{\alpha} |e_1|^{\gamma-1} \right) + (\Lambda + \varepsilon) \text{sgn}(s) - \ddot{q}_d \right] - kd \cdot s \tag{12}$$

Theorem 1. For the system Equation (1), using Equation (11) as the sliding surface, the NNFTSMC control law is designed as Equation (12), then the system will reach the sliding surface in finite time t_s , and the tracking error on the sliding surface will converge to 0 in t_s .

Proof of Theorem 1. The stability analysis of the controller is as follows:

Equation (11) is taken as the first derivative of time to obtain the exponential reaching law, then Equation (12) is substituted into the calculation:

$$\begin{aligned} \dot{s} &= \dot{e}_1 + \frac{1}{\alpha} \gamma |e_1|^{\gamma-1} \dot{e}_1 + \frac{1}{\beta} \eta |e_2|^{\eta-1} \dot{e}_2 \\ &= e_2 + \frac{\gamma}{\alpha} |e_1|^{\gamma-1} e_2 + \frac{\eta}{\beta} |e_2|^{\eta-1} \left(-\tilde{M}(q)^{-1} \tilde{C}(q, \dot{q}) \dot{q} + \tilde{M}(q)^{-1} \tau + D - \ddot{q}_d \right) \\ &= e_2 + \frac{\gamma}{\alpha} |e_1|^{\gamma-1} e_2 + \frac{\eta}{\beta} |e_2|^{\eta-1} \left\{ -\frac{\beta}{\eta} \text{sig}(e_2)^{2-\eta} \left(1 + \frac{\gamma}{\alpha} |e_1|^{\gamma-1} \right) - (\Lambda + \varepsilon) \text{sgn}(s) + D \right\} \\ &= \frac{\eta}{\beta} |e_2|^{\eta-1} [-(\Lambda + \varepsilon) \text{sgn}(s) + D] \end{aligned} \tag{13}$$

Equation (13) makes the speed of the system fast before reaching the switching surface. When reaching the switching surface, the speed decreases and the chattering is weakened, so that the system has better adaptability, and robustness to parameter perturbation and external disturbance.

The Lyapunov function is selected as:

$$V = \frac{1}{2}s^2 \quad (14)$$

Equations (11) and (13) are substituted into Equation (14) and their derivatives are obtained:

$$\begin{aligned} \dot{V} &= s \dot{s} \\ &= s \left(\frac{\eta}{\beta} |e_2|^{\eta-1} [-(\Lambda + \varepsilon) \operatorname{sgn}(s) + D] \right) \\ &= \frac{\eta}{\beta} |e_2|^{\eta-1} [Ds - (\Lambda + \varepsilon)|s|] \end{aligned} \quad (15)$$

Due to $1 < \eta < 2$, $0 < \eta - 1 < 1$, and $\beta > 0$, when $e_2 \neq 0$, $\frac{\eta}{\beta} |e_2|^{\eta-1} > 0$ is established. Equation (15) combined with Equation (8) can be obtained as follows: (16)

$$\begin{aligned} \dot{V} &\leq \frac{\eta}{\beta} |e_2|^{\eta-1} [|D||s| - (\Lambda + \varepsilon)|s|] \\ &\leq \frac{\eta}{\beta} |e_2|^{\eta-1} [\Lambda |s| - (\Lambda + \varepsilon)|s|] \\ &\leq \frac{\eta}{\beta} |e_2|^{\eta-1} (-\varepsilon |s|) \\ &\leq 0 \end{aligned} \quad (16)$$

Therefore, when $e_2 \neq 0$ is applied, the controller satisfies Lyapunov stability condition [34] and has good stability and robustness. The system will arrive at sliding mode surface in finite time. The time for sliding mode surface $s(0) \neq 0$ to $s = 0$ is t_r , when $t = t_r$, $s = 0$, that is $s(t_r) = 0$, we can get:

$$\dot{s} = -\varepsilon \frac{|s|}{s} = \pm \varepsilon \quad (17)$$

By integrating both sides of the above equation, we get the following results:

$$t_r = \left| \frac{s(0)}{\varepsilon} \right| \quad (18)$$

In stage $s = 0$, supposed $e_1(t_r) \neq 0$ passes through finite time t_s from s to $e_1(t_r + t_s) = 0$, \dot{e}_1 can be obtained by Equation (6), Equation (11) and $s = 0$:

$$\dot{e}_1 = -e_1^{\frac{1}{\eta}} \beta^{\frac{1}{\eta}} \left(1 + \frac{1}{\alpha} |e_1|^{\eta-1} \right)^{\frac{1}{\eta}} \quad (19)$$

By integrating and simplifying Equation (19) at the same time, the following results can be obtained:

$$\int_{e_1=e_1(t_r)}^{e_1=e_1(t_r+t_s)=0} \left(\frac{1}{e_1} \right)^{\frac{1}{\eta}} de_1 \leq - \int_{t_r}^{t_r+t_s} \beta^{\frac{1}{\eta}} dt \quad (20)$$

$$t_s \leq \frac{\eta}{\beta^{\frac{1}{\eta}} (\eta - 1)} e_1(t_r)^{1-\frac{1}{\eta}} \quad (21)$$

The total convergence time is as follows:

$$t = t_r + t_s \leq \left| \frac{s(0)}{\varepsilon} \right| + \frac{\eta}{\beta^{\frac{1}{\eta}} (\eta - 1)} e_1(t_r)^{1-\frac{1}{\eta}} \quad (22)$$

□

Theorem 2. Was proved.

Remark 1. Due to the existence of $\frac{1}{\alpha} \operatorname{sig}(e_1)^\eta$ term in Equation (11), the total convergence time of Equation (22) is less than that of NTSMC proposed in [26].

3.2. Design of DDPG Network

According to the derivation process of Section 3.1, the uncertainty D in assumption 2 satisfies: $|D| \leq \Lambda$ is an important condition of Lyapunov stability.

The modified DDPG algorithm can improve the stability and anti-interference of the control system. The delay of control parameters' return and the update makes the action calculated according to the current state take effect in the next control phase, the weights of the neural network and the parameters of experience pool are updated synchronously. During the execution of the algorithm, the data onto the previous step combined with the error information is converted into a reward parameter, which is combined with a data area with N steps. At the beginning of training, the actor network $Q(\Gamma, a|\theta^Q)$ (using hyper-parameter θ^Q) and critic network $\mu(\Gamma|\theta^\mu)$ (using hyperparameter theta θ^μ) are initialized, and the experience set κ is initialized. For the critic algorithm, the current round monitoring data and state and the next action (combined as vectors: Γ) are used as the input parameters of the target network, and then scalar values are output to calculate target values. The formula for target value was:

$$y_i = r_i + \gamma Q'[\Gamma_{i+1}, \mu'(\Gamma_{i+1}|\theta^{\mu'})|\theta^Q] \tag{23}$$

The current network in the critic algorithm takes the latest state action history and current states Γ as input and outputs the new action to be taken. This paper uses a real-time update feedback training program with multiple training rounds. In each execution cycle, each batch (in random order) of training data is used to update the weights of the neural network using gradient descent. The critic network is updated by minimizing the mean square error between the output of Q and the original reward data. When taking action in the current state, according to the output of μ , the policy network is updated by minimizing the output of Q. For the action-value function of critic network output, one part is used to calculate the mean square error:

$$L = \frac{1}{N} \sum_i [y_i - Q(\Gamma_i, a_i|\theta^Q)]^2 \tag{24}$$

Network used to update actor part at the same time:

$$\nabla_{\theta^\mu} J \approx \sum_i V_a Q(\Gamma, a|\theta^Q) \Big|_{\Gamma=\Gamma_i, \alpha=\mu(\Gamma_i)} \nabla_{\theta^\mu} \mu(\Gamma|\theta^\mu) \Big|_{\Gamma_i} \tag{25}$$

The key of the reward function is to make the network give correct feedback to the network according to the execution results after making decision actions. The merits of the reward function directly affect the training effect and convergence speeds. In reinforcement learning, there is no strict definition of the reward function. It only needs to be able to correctly evaluate the advantages and disadvantages of network output actions. There are two kinds of common reward functions. One is a sparse reward, which is often used in the game to score after completing the task. This kind of reward does not respond well to nondiscrete actions, so it is difficult to quantify the size of the reward. The other is a formal reward, which is only given in the target state, and not anywhere else. This formal reward is easier to promote neural network learning, even if the strategy does not find a solution to the problem, the reward function can also provide positive feedback.

In this paper, the input value of the theoretical signal (including but not limited to position, velocity, acceleration, etc.) and the parameters of feedback adjustment (including but not limited to tracking position, tracking velocity, tracking error, control rate, etc.) are combined. According to the actual situation, the correlation signal is multiplied by the adjustment coefficient and superimposed, and the results are provided for the reinforcement learning network as the reward function value.

In actor-network, the role of actor current network is responsible for the iterative update of policy network parameter θ , according to the current state Γ_t selects the current action a and interacts with the environment to generate Γ_{t+1} . The actor target network is responsible for selecting the optimal next action a_t according to the next state Γ_{t+1} sampled in the experience playback pool. At the same time, m samples are sampled in the experience set κ to calculate the Q value of the current state, and soft update is used to update the network weight periodically.

$$\begin{cases} \theta^{Q'} \leftarrow \tau\theta^{Q'} + (1 - \tau)\theta^Q \\ \theta^{\mu'} \leftarrow \tau\theta^{\mu'} + (1 - \tau)\theta^{\mu} \end{cases} \quad (26)$$

When the neural network is needed to output the action, the current state Γ_t is inputted into the actor target network to get the output after selecting the action with the largest reward, a_t corresponds to a variable within a predefined range. The formula is as follows:

$$a_t = \mu(\Gamma_t | \theta^{\mu}) + N_t \quad (27)$$

To improve the foresight of action selection, the random noise N_t was added to ensure that the output action has certain randomness, which could be added or removed dynamically according to the need in practical application.

The DDPG algorithm used in this paper has the following improvements compared with the traditional DDPG algorithm: 1. In the critic network, the output signal was changed from the single numerical to the combined control rate and error, so as to improve the performance requirements of control rate for the early stage of the model; 2. In the last layer of the performer network, the output was mapped from the discrete action output with the highest probability selected from the experience pool to the numerical output with the highest probability selected. The control diagram is as Figure 1:

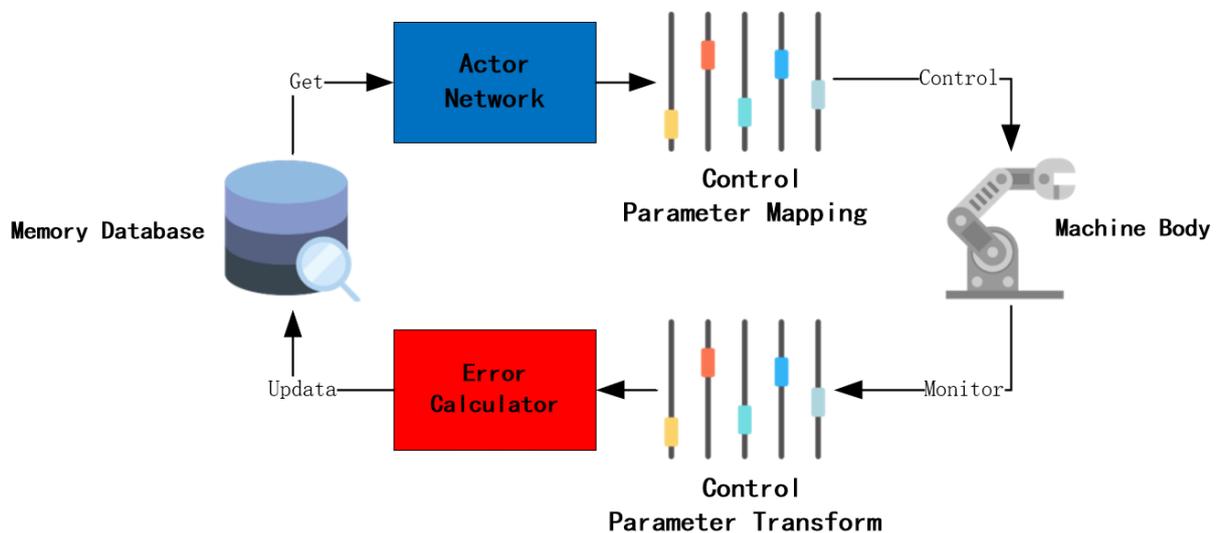


Figure 1. DDPG network control structure chart.

Aiming at the two joint control model used in this experiment, two lightweight DDPG networks are used to jointly control the target, which respectively undertakes the error compensation output and control adjustment of the two joints. In the experiment, using the traditional network architecture will make the training model too complex, the convergence speed is significantly reduced, and there will be an overfitting phenomenon. Therefore, the network uses two layers of network with 32 and 64 neurons as actor-network, two layers of critic-network with 32 neurons each are used to judge and correct the performer network. The control flow chart of DDPG–NNFTSMC is shown in Figure 2.

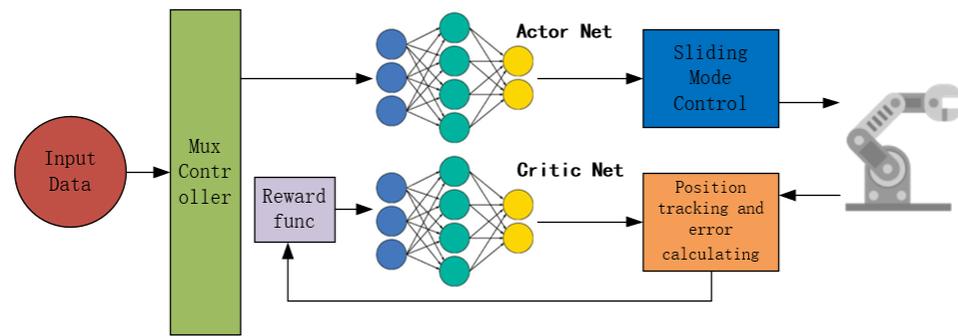


Figure 2. DDPG–NNFTSMC control flow chart.

4. Experimental Results and Discussion

4.1. Simulation Comparison with Control Algorithms

To verify the effectiveness of DDPG–NNFTSMC proposed in this paper, the dynamic model of two joint manipulator was introduced into this section. The simulation analysis was carried out by using MATLAB/Simulink, and the sampling rate was set to 10^{-3} s, the sensor was used to measure the corresponding position accuracy, response speed, and the path tracking control of each joint. Considering the characteristics of the control system, the external disturbance and uncertain friction were modeled. The control effect was compared with TSMC, NTSMC, and RBF–SMC. Figure 3 is the pseudo-code of the proposed algorithm.

```

Main:
do Init Parameters and Simulink model.
IF Init Parameter and Simulink models successfully
{
do open agent mdl.
LOOP until Epochs reach the preset time
{
do Get the Input data base on time.
do input data to the NNFTSMC Control Function and get the output data.
do input the data to the Simulink model and get its output to input DDPG algorithm.
LOOP until output error approach enough to the ideal data
{
Compare DDPG outputs and NNFTSMC outputs.
IF ABS (DDPG outputs – NNFTSMC outputs) > MAX_ERROR
{
Calculate the reward via the value function.
Train and update the DDPG network.
}
do add the NNFTSMC output to the newly input data.
}
}
ELSE
{
Exit the program.
}
}
}
    
```

Figure 3. Pseudo-code of the proposed algorithm.

The second-order manipulator model designed according to [5,35] was shown in Equation (1), where:

$$M(q) = \begin{bmatrix} \rho_1 + 2\rho_3 \cos(q_2) + 2\rho_4 \sin(q_2) & \rho_2 + \rho_3 \cos(q_2) + \rho_4 \sin(q_2) \\ \rho_2 + \rho_3 \cos(q_2) + \rho_4 \sin(q_2) & \rho_2 \end{bmatrix} \quad (28)$$

$$C(q, \dot{q}) = \begin{bmatrix} (-2\rho_3 \sin(q_2) + 2\rho_4 \cos(q_2)) \dot{q}_2 & (-\rho_3 \sin(q_2) + \rho_4 \cos(q_2)) \dot{q}_2 \\ (\rho_3 \sin(q_2) - \rho_4 \cos(q_2)) \dot{q}_1 & 0 \end{bmatrix} \quad (29)$$

$$G(q) = \begin{bmatrix} \rho_3 St_2 \cos(q_1 + q_2) + \rho_4 St_2 \sin(q_1 + q_2) + (\rho_1 - \rho_2 + St_1) St_2 \cos(q_1) \\ \rho_3 St_2 \cos(q_1 + q_2) + \rho_4 St_2 \sin(q_1 + q_2) \end{bmatrix} \quad (30)$$

The friction force was:

$$F(\dot{q}) = 2 \operatorname{sgn}(\dot{q}) \quad (31)$$

The external interference is as follows:

$$\tau_d = [10 \sin(\dot{q}) \quad 10 \sin(\dot{q})]^T \quad (32)$$

The uncertainty parameters are as follows:

$$\rho = [\rho_1 \quad \rho_2 \quad \rho_3 \quad \rho_4]^T \quad (33)$$

The physical parameters of the two joints manipulator are shown in Table 2. Table 3 was the control parameter selection for the control strategies.

Table 2. Physical parameters of two joint manipulator.

m_1	l_1	l_{c1}	I_1	m_e	l_{ce}	I_e	δ_e	C_1	C_2
1 kg	1 m	0.5 m	1/12 kg	3 kg	1 m	0.4 kg	0	-7/12	9.81

Table 3. The control parameter selection for the control strategies.

Control Strategy	Control Parameters	Parameter Value
TSMC	$(\alpha, \beta, \gamma, \delta, \varphi)$	(0.75, 0.9, 1.2, -0.1, 160)
NTSMC	$(\alpha, \beta, \gamma, \delta, \varphi, \epsilon, \mu)$	(0.75, 0.9, 1.1, 0.1, 160, 22/11, 25/11)
RBF-SMC	$(p^1, p^2, p^3, p^4, p^5)$ $M, \varphi, \epsilon, \mu$	(2.9, 0.76, 0.87, 3.04, 0.87, 1, 0.1, 5, 10)
DDPG-NNFTSMC	$(\rho_1, \rho_2, \rho_3, \rho_4, \alpha, \beta)$ $, \gamma, \delta, \eta, \varphi, \Lambda, \epsilon$	$(\rho_1, \rho_2, \rho_3, \rho_4, 10, 10,$ 25/11, 21/11, 160, 2, 0.1)

This paper compared Equation (1) with TSMC and NTSMC proposed in [5,26], RBF-SMC [5,19], NNFTSMC and DDPG-NNFTSMC proposed in this paper. In order to further compare the effects of each control strategy, Equation (28) was led to calculate the average position error \bar{E}_s and the average speed error \bar{E}_v according to [9], where n was the simulation step size, and the results are shown in Tables 4 and 5.

$$\bar{E} = \sqrt{\frac{1}{n} \sum_{k=1}^n (||e_i||)^2} \quad (i = 1, 2, 3 \dots) \quad (34)$$

Table 4. The control parameter selection for the control strategies.

Error Control Strategy	\bar{E}_{s1}	\bar{E}_{s2}
TSMC	0.6083	0.3986
NTSMC	0.6449	0.4288
RBF-SMC	0.4459	0.1151
DDPG-NNFTSMC	0.0690	0.0757

Table 5. The average speed errors under control input signals of the control strategy.

Error Control Strategy	\bar{E}_{v1}	\bar{E}_{v2}
TSMC	3.1022	2.0766
NTSMC	3.9493	2.6833
RBF-SMC	3.5170	0.9385
DDPG-NNFTSMC	0.5416	0.4756

Figures 4–7 correspond to the tracking performance of four controllers, including tracking position and tracking error of each joint, tracking speed and speed error. Figure 8 showed the control inputted signals of each controller compared with this paper, including the traditional TSMC, NTSMC, and RBF-SMC, NNFTSMC, and DDPG-NNFTSMC designed in this paper.

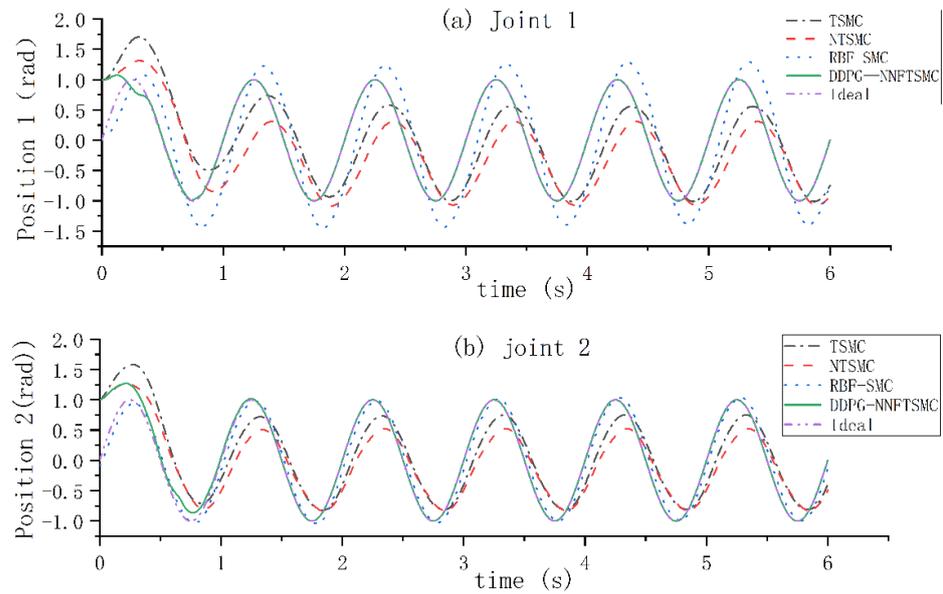


Figure 4. Trajectory tracking positions.

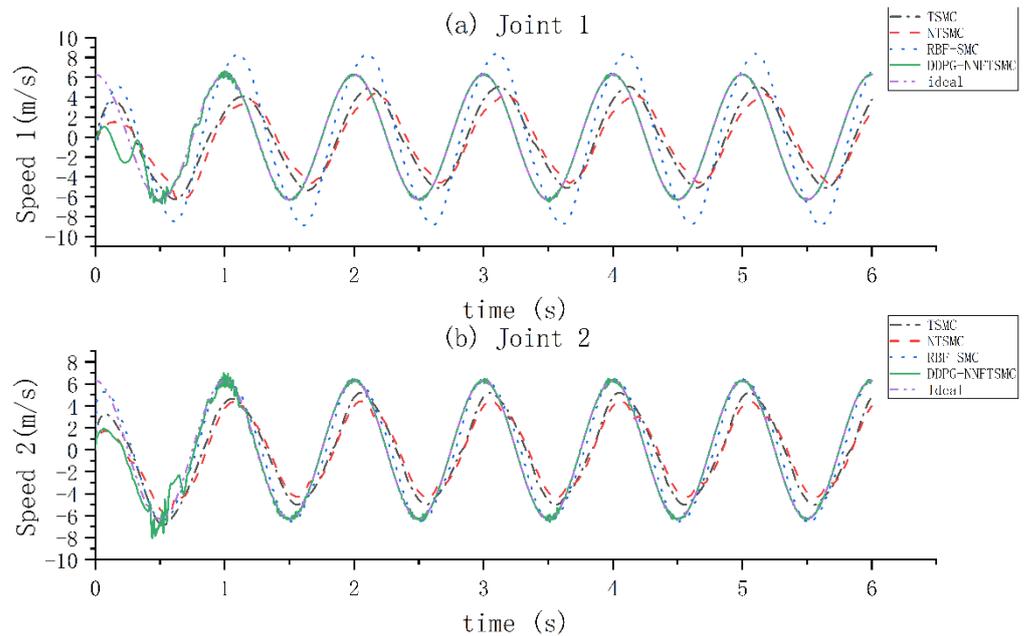


Figure 5. Trajectory tracking speeds.

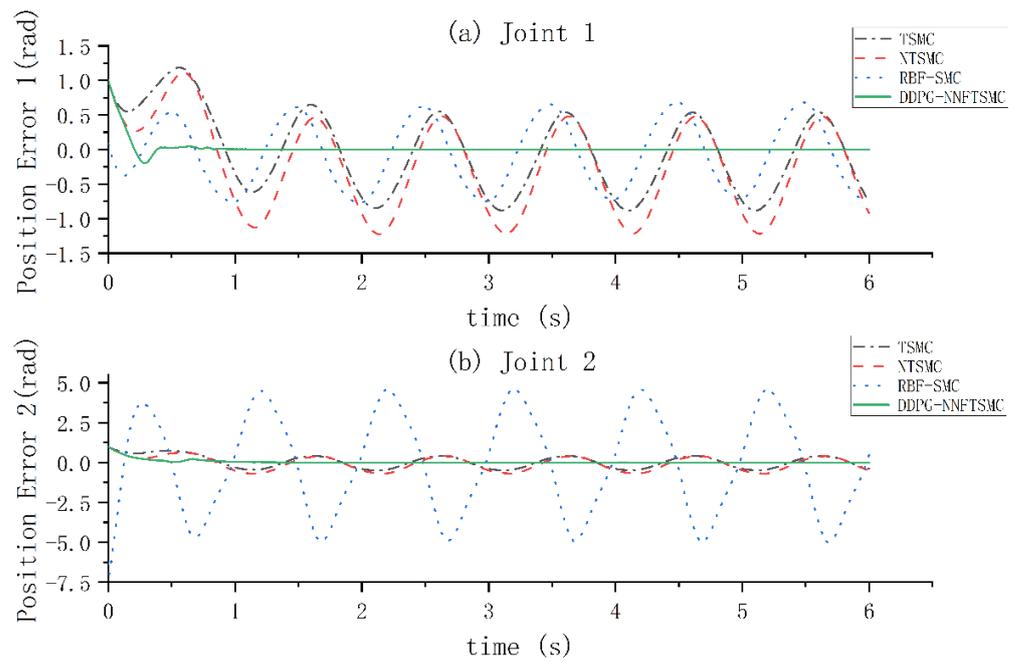


Figure 6. Position tracking errors.

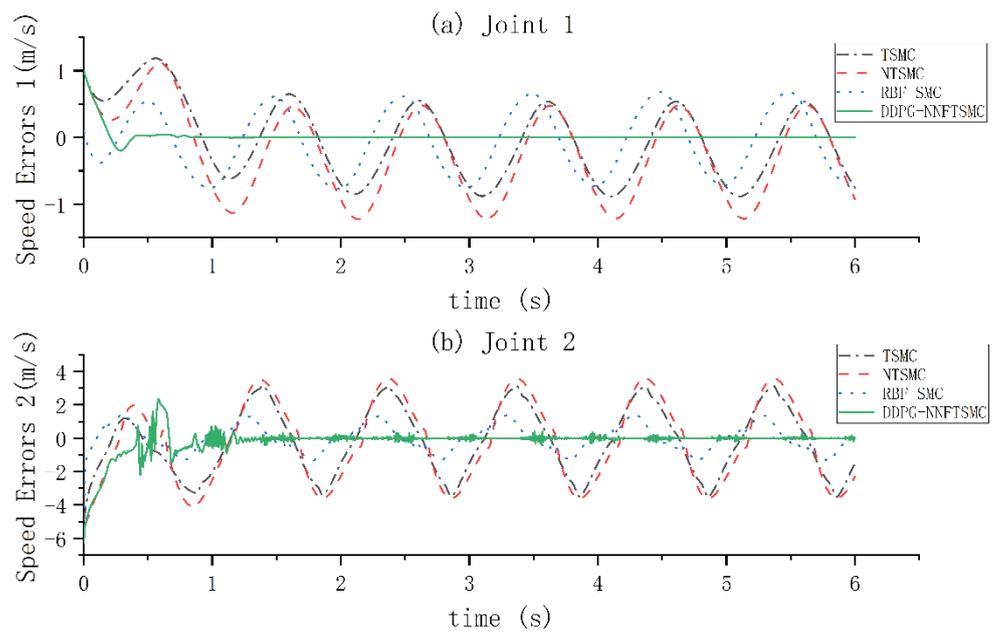


Figure 7. Speed tracking errors.

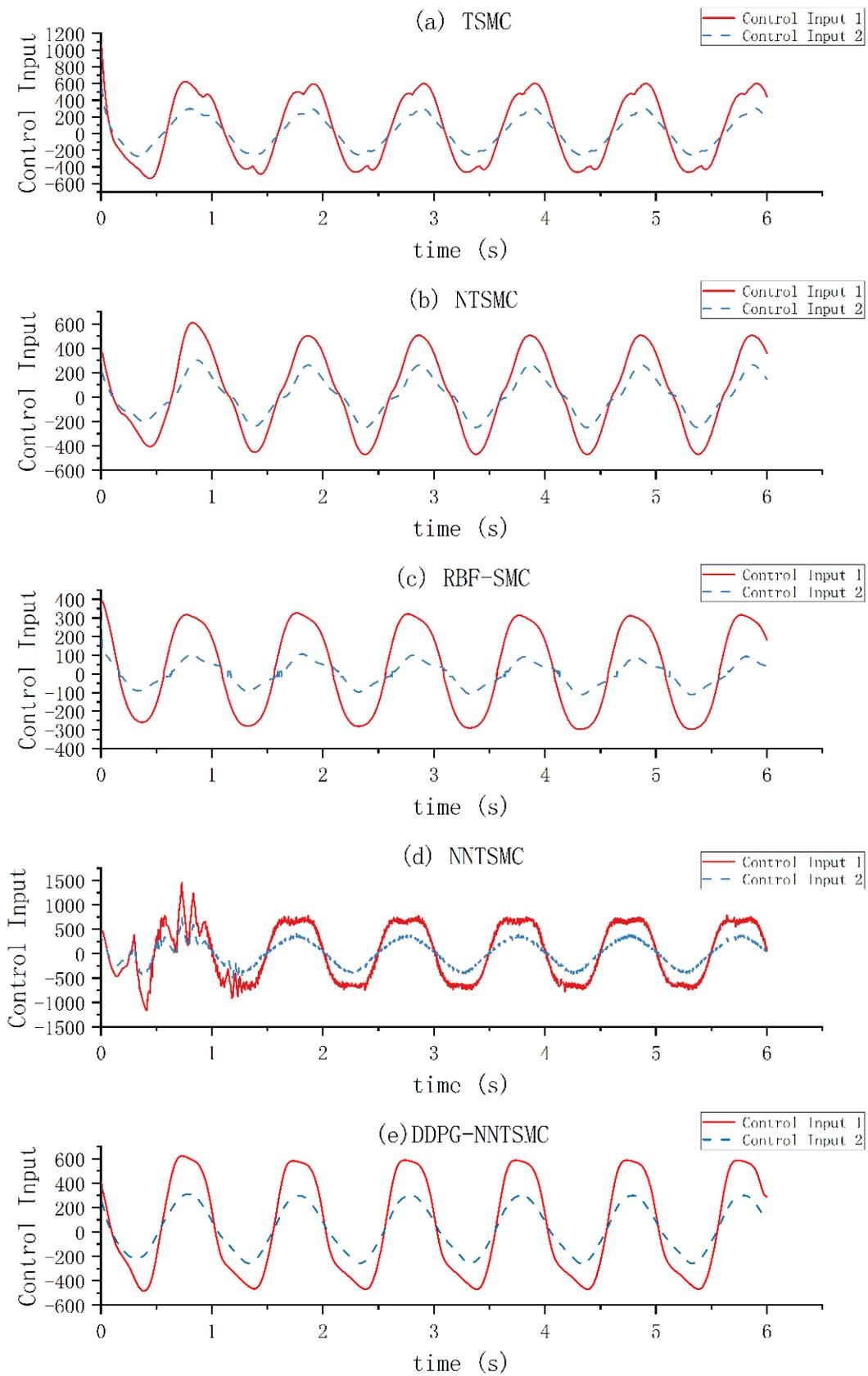


Figure 8. Control input signals of each controllers.

4.2. Results and Discussion

From Tables 4 and 5 and Figures 4–7, it could be seen that the position error and velocity error of traditional TSMC, NTSMC, and RBF–SMC changed with the application of uncertain interference and friction. Compared with the three strategies, DDPG–NNFTSMC was much smaller in position error, velocity error, average position error and average velocity error, the corresponding tracking position and speed almost fitted the theoretical value. In addition, the proposed sliding surface function s was designed based on the control function in Equation (12), which played an important role in providing fast convergence and robustness to uncertainties and disturbances. Compared with other control strategies, the control strategy proposed in this paper provided the best path tracking performance and the fastest convergence speed.

In Figure 8a,b compared with the traditional TSMC and NTSMC which converged on finite time, but for nonlinear and nonsingular systems, there were still system chattering and errors, so we must choose between chattering elimination and path tracking accuracy. Therefore, the robustness of the system was reduced, and the tracking error was increased. As shown in Figure 8c, although the control input 1 of RBF–SMC had the characteristics of fast convergence speed and elimination of chattering phenomenon, the tracking error was greatly increased; control input 2 provided a continuous control signal with partial chattering behavior, but the tracking error also decreases, as shown in Figures 4–7. The NNFTSMC designed in this paper provided continuous control signals for the manipulator, and led DDPG network into the controller, the tracking accuracy was guaranteed, and the chattering phenomenon of Figure 8d was eliminated, and the convergence time of the control system signal was effectively reduced without losing its effectiveness, which is shown in Figure 8e. These adaptive feedbacks were estimated according to the change of system disturbance and uncertain disturbance terms. Once the error variables converge to the sliding surface, they would be close to a constant value. From the simulation results, the controller was superior to the traditional TSMC, NTSMC, and RBF–SMC in tracking accuracy, convergence speed, and chattering elimination.

5. Conclusions

In this paper, a DDPG–NNFTSMC control strategy was proposed to solve the problem of chattering and chattering caused by traditional sliding mode control, which is successfully applied to manipulator systems with uncertain dynamic characteristics. Based on NNFTSMC, the sliding surface was proposed, and the error function can converge on the sliding surface quickly. Then the Lyapunov stability condition was used to prove that the NNFTSMC has good stability and finite time convergence. Compared with the traditional TSMC, NTSMC, and RBF–SMC, the DDPG–NNFTSMC has the following advantages: (1) DDPG network is used to train and update the control parameters in real time to estimate the model uncertainties (including unknown disturbance, dynamic model uncertainty, and approximation error), which effectively eliminates the chattering phenomenon of the system and ensures the robustness of the system under various uncertain disturbances; (2) the chattering elimination greatly improves the tracking accuracy, reduces the tracking average error, and enhances the ability of antisturbance and uncertainty of the system. Therefore, it can be concluded that DDPG–NNFTSMC proposed in this paper has excellent control performance, and theoretically has good application prospects for industrial manipulators with uncertain dynamic models and general second-order nonlinear systems. Next, based on the research of scholars [12,31], we will try to apply DDPG–NNFTSMC to other different scenarios and combine it with deep reinforcement learning to further improve the control performance [32,36,37].

Author Contributions: Conceptualization, Z.X. and W.H.; methodology, Z.X. and W.H.; software, Z.X. and Z.L.; validation, Z.X. and L.H.; W.H.; results analysis, Z.X. and P.L.; investigation, Z.X. and L.H. and Z.L.; data curation, Z.X. and P.L.; writing—original draft preparation, Z.X. and W.H.; writing—review and editing, Z.X. and W.H.; visualization, W.H.; supervision, Z.X. and W.H.; project administration, Z.X. and W.H.; funding acquisition, W.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Guangzhou Science and Technology Planning Project, grant number 202002030279.

Acknowledgments: This work was supported by the Basic Science Research Program through the Guangzhou Science and Technology Planning Project (202002030279).

Conflicts of Interest: All authors announce that they have no conflict of interest in relation to the publication of this article.

References

1. Lopez, B.T.; Slotine, J.-J.E. Adaptive Nonlinear Control with Contraction Metrics. *IEEE Control. Syst. Lett.* **2021**, *5*, 205–210. [[CrossRef](#)]
2. Yu, X.; Zhang, S.; Fu, Q.; Xue, C.; Sun, W. Fuzzy Logic Control of an Uncertain Manipulator with Full-State Constraints and Disturbance Observer. *IEEE Access* **2020**, *8*, 24284–24295. [[CrossRef](#)]
3. Zhang, Z.; Yan, Z. An Adaptive Fuzzy Recurrent Neural Network for Solving the Nonrepetitive Motion Problem of Redundant Robot Manipulators. *IEEE Trans. Fuzzy Syst.* **2019**, *28*, 684–691. [[CrossRef](#)]
4. Wang, Y.; Karimi, H.R.; Shen, H.; Fang, Z.; Liu, M. Notice of Violation of IEEE Publication Principles: Fuzzy-Model-Based Sliding Mode Control of Nonlinear Descriptor Systems. *IEEE Trans. Cybern.* **2019**, *49*, 3409–3419. [[CrossRef](#)] [[PubMed](#)]
5. Liu, J. *Sliding Mode Control and MATLAB Simulation. The Basic Theory and Design Method*, 3rd ed.; Tsing-Hua University Press: Beijing, China, 2015.
6. Bucolo, M.; Buscarino, A.; Famoso, C.; Fortuna, L.; Frasca, M. Control of imperfect dynamical systems. *Nonlinear Dyn.* **2019**, *98*, 2989–2999. [[CrossRef](#)]
7. Ahmed, S.; Wang, H.; Tian, Y. Adaptive High-Order Terminal Sliding Mode Control Based on Time Delay Estimation for the Robotic Manipulators with Backlash Hysteresis. *IEEE Trans. Syst. Man Cybern. Syst.* **2021**, *51*, 1128–1137. [[CrossRef](#)]
8. Wang, Y.; Xia, Y.; Shen, H.; Zhou, P. SMC Design for Robust Stabilization of Nonlinear Markovian Jump Singular Systems. *IEEE Trans. Autom. Control.* **2017**, *63*, 219–224. [[CrossRef](#)]
9. Vo, A.T.; Kang, H.-J. An Adaptive Neural Non-Singular Fast-Terminal Sliding-Mode Control for Industrial Robotic Manipulators. *Appl. Sci.* **2018**, *8*, 2562. [[CrossRef](#)]
10. Fallaha, C.; Saad, M.; Ghommam, J.; Kali, Y. Sliding Mode Control with Model-Based Switching Functions applied on a 7-DOF Exoskeleton Arm. *IEEE/ASME Trans. Mechatron.* **2020**, *26*, 1. [[CrossRef](#)]
11. Rahmani, M.; Ghanbari, A.; Etefagh, M.M. Hybrid neural network fraction integral terminal sliding mode control of an Inchworm robot manipulator. *Mech. Syst. Signal Process.* **2016**, *80*, 117–136. [[CrossRef](#)]
12. Yoerger, D.R.; Slotine, J.-J.E. Adaptive sliding control of an experimental underwater vehicle. In Proceedings of the 1991 IEEE International Conference on Robotics and Automation, Sacramento, CA, USA, 9–11 April 1991; Volume 3, pp. 2746–2751. [[CrossRef](#)]
13. Wang, J.; Zhai, A.; Xu, F.; Zhang, H.; Lu, G. Dual feedforward neural networks based synchronized sliding mode controller for cooperative manipulator system under variable load and uncertainties. *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.* **2020**, *234*, 3859–3872. [[CrossRef](#)]
14. Chen, M.; Ge, S.S. Adaptive Neural Output Feedback Control of Uncertain Nonlinear Systems with Unknown Hysteresis Using Disturbance Observer. *IEEE Trans. Ind. Electron.* **2015**, *62*, 7706–7716. [[CrossRef](#)]
15. Wang, H.; Shi, P.; Li, H.; Zhou, Q. Adaptive Neural Tracking Control for a Class of Nonlinear Systems with Dynamic Uncertainties. *IEEE Trans. Cybern.* **2017**, *47*, 3075–3087. [[CrossRef](#)]
16. Zhao, X.; Yang, H.; Xia, W.; Wang, X. Adaptive Fuzzy Hierarchical Sliding-Mode Control for a Class of MIMO Nonlinear Time-Delay Systems with Input Saturation. *IEEE Trans. Fuzzy Syst.* **2016**, *25*, 1062–1077. [[CrossRef](#)]
17. Yang, Y.; Yan, Y. Attitude regulation for unmanned quadrotors using adaptive fuzzy gain-scheduling sliding mode control. *Aerosp. Sci. Technol.* **2016**, *54*, 208–217. [[CrossRef](#)]
18. Tao, X.; Yi, J.; Pu, Z.; Xiong, T. Robust Adaptive Tracking Control for Hypersonic Vehicle Based on Interval Type-2 Fuzzy Logic System and Small-Gain Approach. *IEEE Trans. Cybern.* **2021**, *51*, 2504–2517. [[CrossRef](#)] [[PubMed](#)]
19. Yang, H.-J.; Tan, M. Sliding Mode Control for Flexible-link Manipulators Based on Adaptive Neural Networks. *Int. J. Autom. Comput.* **2018**, *15*, 239–248. [[CrossRef](#)]
20. Han, S.; Wang, H.; Tian, Y.; Christov, N. Time-delay estimation based computed torque control with robust adaptive RBF neural network compensator for a rehabilitation exoskeleton. *ISA Trans.* **2020**, *97*, 171–181. [[CrossRef](#)]

21. Kamal, S.; Moreno, J.A.; Chalanga, A.; Bandyopadhyay, B.; Fridman, L.M. Continuous terminal sliding-mode controller. *Automatica* **2016**, *69*, 308–314. [[CrossRef](#)]
22. Le, Q.D.; Kang, H.-J. Finite-Time Fault-Tolerant Control for a Robot Manipulator Based on Synchronous Terminal Sliding Mode Control. *Appl. Sci.* **2020**, *10*, 2998. [[CrossRef](#)]
23. 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](#)]
24. Pan, H.; Zhang, G.; Ouyang, H.; Mei, L. A Novel Global Fast Terminal Sliding Mode Control Scheme for Second-Order Systems. *IEEE Access* **2020**, *8*, 22758–22769. [[CrossRef](#)]
25. Su, Y.; Zheng, C. A new nonsingular integral terminal sliding mode control for robot manipulators. *Int. J. Syst. Sci.* **2020**, *51*, 1418–1428. [[CrossRef](#)]
26. Zhai, J.; Xu, G. A Novel Non-Singular Terminal Sliding Mode Trajectory Tracking Control for Robotic Manipulators. *IEEE Trans. Circuits Syst. II Express Briefs* **2021**, *68*, 391–395. [[CrossRef](#)]
27. Mobayen, S.; Mostafavi, S.; Fekih, A. Non-singular fast terminal sliding mode control with disturbance observer for underactuated robotic manipulators. *IEEE Access* **2020**, *8*, 1. [[CrossRef](#)]
28. Truong, T.N.; Vo, A.T.; Kang, H.-J. A Backstepping Global Fast Terminal Sliding Mode Control for Trajectory Tracking Control of Industrial Robotic Manipulators. *IEEE Access* **2021**, *9*, 31921–31931. [[CrossRef](#)]
29. Hu, Y.; Wang, W.; Liu, H.; Liu, L. Reinforcement Learning Tracking Control for Robotic Manipulator with Kernel-Based Dynamic Model. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *31*, 3570–3578. [[CrossRef](#)]
30. Wen, G.; Chen, C.L.P.; Li, B. Optimized Formation Control Using Simplified Reinforcement Learning for a Class of Multiagent Systems with Unknown Dynamics. *IEEE Trans. Ind. Electron.* **2019**, *67*, 7879–7888. [[CrossRef](#)]
31. Spong, M.W.; Vidyasagar, M. *Robot Dynamics and Control*; John Wiley & Sons: New York, NY, USA, 1989.
32. Wang, Y.; Li, S.; Wang, D.; Ju, F.; Chen, B.; Wu, H. Adaptive Time-Delay Control for Cable-Driven Manipulators with Enhanced Nonsingular Fast Terminal Sliding Mode. *IEEE Trans. Ind. Electron.* **2021**, *68*, 2356–2367. [[CrossRef](#)]
33. Deng, B.; Shao, K.; Zhao, H. Adaptive Second Order Recursive Terminal Sliding Mode Control for a Four-Wheel Independent Steer-by-Wire System. *IEEE Access* **2020**, *8*, 75936–75945. [[CrossRef](#)]
34. Mendoza-Avila, J.; Efimov, D.; Ushirobira, R.; Moreno, J.A. Numerical design of Lyapunov functions for a class of homogeneous discontinuous systems. *Int. J. Robust Nonlinear Control* **2021**. [[CrossRef](#)]
35. Liu, J. *Sliding Mode Control and MATLAB Simulation. The Design Method of Advanced Control System*, 3rd ed.; Tsing-Hua University Press: Beijing, China, 2015.
36. Kumar, A.; Sharma, R. Linguistic Lyapunov reinforcement learning control for robotic manipulators. *Neurocomputing* **2018**, *272*, 84–95. [[CrossRef](#)]
37. Lin, Y.; Huang, J.; Zimmer, M.; Guan, Y.; Rojas, J.; Weng, P. Invariant Transform Experience Replay: Data Augmentation for Deep Reinforcement Learning. *IEEE Robot. Autom. Lett.* **2020**, *5*, 6615–6622. [[CrossRef](#)]