




## Article

# Fractional-Order Control Method Based on Twin-Delayed Deep Deterministic Policy Gradient Algorithm

Guangxin Jiao <sup>1</sup>, Zhengcai An <sup>1</sup>, Shuyi Shao <sup>1,\*</sup>  and Dong Sun <sup>2</sup>

<sup>1</sup> College of Automation Engineering, Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing 211106, China; 17863960870@163.com (G.J.); azc0802@163.com (Z.A.)

<sup>2</sup> Jincheng Nanjing Engineering Institute of Aircraft Systems, Nanjing 211199, China; dongsun\_njust@126.com

\* Correspondence: shaosy@nuaa.edu.cn

**Abstract:** In this paper, a fractional-order control method based on the twin-delayed deep deterministic policy gradient (TD3) algorithm in reinforcement learning is proposed. A fractional-order disturbance observer is designed to estimate the disturbances, and the radial basis function network is selected to approximate system uncertainties in the system. Then, a fractional-order sliding-mode controller is constructed to control the system, and the parameters of the controller are tuned using the TD3 algorithm, which can optimize the control effect. The results show that the fractional-order control method based on the TD3 algorithm can not only improve the closed-loop system performance under different operating conditions but also enhance the signal tracking capability.

**Keywords:** FODOB; FOSMC; radial basis function network; TD3 algorithm



**Citation:** Jiao, G.; An, Z.; Shao, S.; Sun, D. Fractional-Order Control Method Based on Twin-Delayed Deep Deterministic Policy Gradient Algorithm. *Fractal Fract.* **2024**, *8*, 99. <https://doi.org/10.3390/fractalfract8020099>

Academic Editors: Germán Ardúl Muñoz Hernández and Fermi Guerrero-Castellanos

Received: 30 December 2023

Revised: 23 January 2024

Accepted: 3 February 2024

Published: 6 February 2024



**Copyright:** © 2024 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

Fractional-order calculus stands out for its flexible description of the behavior of nonlocal and non-Markovian dynamics, providing a rich mathematical tool for modeling complex systems. Its main applications include the modeling of nonlinear and nonsmooth system dynamics [1], the simulation of multiscale complex systems [2], the analysis of non-Markovian processes [3], as well as in the fields of signal processing, control systems and financial modeling [4]. In the field of control, fractional-order control has been combined with many traditional control schemes, such as fractional-order PID control [5], fractional-order robust control [6], and fractional-order sliding-mode control [7]; these fractional-order control methods have been developed in depth at the theoretical level to form a sound theoretical system and have achieved extensive and powerful results in practical applications.

As a traditional control strategy, fractional-order sliding-mode control is able to cope with the nonlinearity and uncertainty of the system more flexibly [8–10]. However, with the increase in system complexity and nonlinearity, choosing the optimal values of controller parameters often becomes a great difficulty [11]. Faced with this problem, different scholars have used a variety of optimization algorithms over the years, such as early adaptive control [12], the particle swarm algorithm [13], the genetic algorithm [14], and the wolf-pack algorithm [15]. These algorithms have had some success, but as technology advances, they struggle to handle increasingly complex problems.

In recent years, with the continuous development of artificial intelligence technology, more and more advanced algorithms have been applied to the control field [16]. Among them, reinforcement learning algorithms based on neural networks are increasingly becoming a focus of research because their flexibility and adaptability make them more adaptable to complex environments [17–23]. Reinforcement learning efficiently masters system dynamics by learning through the interaction of an agent with the environment [24]. This approach transforms the system control problem into a process in which the agent learns the optimal control strategy through continuous trial and error [25], and it can also

learn without prior knowledge of the system model [26,27], which offers a flexible approach for real-time controller parameter optimization. Specifically, under the framework of reinforcement learning, the intelligent body adjusts the control strategy according to feedback signals by interacting with the environment and gradually optimizes the controller parameters [28,29]. This learning approach is more adaptive, especially in the face of a high-order system complexity, significant nonlinearities, and rapid changes in dynamic characteristics, where traditional static parameter optimization methods may appear to be inadequate [30]. Therefore, reinforcement learning provides a more intelligent solution for controller parameter optimization, which is expected to show more significant performance improvements in practical applications.

In this paper, a fractional-order control method based on TD3 reinforcement learning is proposed to optimize the parameters. A fractional-order disturbance observer is designed for estimating the disturbance signals present in the system, while an RBF network is designed for approximating the possible uncertainties in the system, and finally, a fractional-order sliding-mode controller is designed for controlling the system. For the parameter optimization part, it is performed by the TD3 reinforcement learning algorithm, which consists of six deep neural network networks designed to inhibit the bootstrap phenomenon in the reinforcement learning algorithm [31], so that the output of the network can converge to the optimal solution quickly. On this theoretical basis, a valve-controlled hydraulic system is selected for design and simulation in Matlab/Simulink to verify the effectiveness of the method proposed in this paper, and at the same time, in order to reflect the advantages and disadvantages of the proposed method, a series of comparative experiments are designed.

In summary, the main work of this paper can be succinctly summarized in the following three areas:

- i. The fractional-order disturbance observer is designed to estimate the system disturbance signal, and the RBF network is selected to approximate the uncertainties of the system; then, a fractional-order sliding-mode controller is designed to control the system according to the estimated value of the disturbance observer;
- ii. The TD3 algorithm is introduced to optimize the parameters of the controller, and an improved loss function is designed to improve the learning performance of the algorithm, accelerate the convergence of the network output, and optimize the control effect;
- iii. It is verified through simulation that not only the control effect of the proposed method is better than the selected comparison control method, but that its also has a great robustness and generalization capability.

This paper is organized as follows. Section 2 introduces the proposed fractional-order control method, including the system state equation generalization and the designed fractional-order disturbance observer with an RBF network structure, based on which the fractional-order sliding-mode controller is designed. Section 3 introduces the fundamentals of the TD3 algorithm [32] and defines the reward function and loss function used. Section 4 demonstrates the stability of the designed method to verify its theoretical correctness. In Section 5, the proposed method is simulated in Matlab/Simulink using a valve-controlled hydraulic system as a model, and a series of control simulations are performed to verify the practical feasibility of the proposed method. Finally, conclusions and future work are presented.

## 2. Design of Fractional-Order Control Method

In this section, the fractional-order control part of the proposed method is accomplished for a system generalization including a fractional-order disturbance observer, an RBF network, and a fractional-order sliding-mode controller.

For a common system, the equation of state can be expressed as

$$\begin{cases} \dot{x} = Ax + Bu + f(x) + d \\ y = Cx \end{cases} \quad (1)$$

where  $x$  is the system state, and  $x \in R^n$ ;  $A$  is the system parameter matrix,  $A \in R^{n \times n}$ ,  $B$  is the control matrix, and  $B \in R^{n \times q}$ ;  $u$  is the control signal,  $u \in R^q$ ,  $C \in R^{p \times n}$  is the output matrix,  $d \in R^{n \times 1}$  is the disturbance signal, and  $f(x) \in R^{n \times 1}$  is the uncertainty of system.

In this paper, the Riemann–Liouville fractional-order calculus formula is defined as [33]:

$$D^\alpha g(x) = \frac{1}{\Gamma(n-\alpha)} \frac{d^n}{dx^n} \int_a^x \frac{g(t)}{(x-t)^{\alpha+1-n}} dt \quad (2)$$

where  $\alpha$  represents the differential,  $0 < \alpha < 1$ ,  $\Gamma(\cdot)$  represents the Gamma function,  $n$  represents the smallest integer larger than  $\alpha$ , usually taken as 1, and  $a, x$  represents the lower and upper limits of the integral.

For the disturbance signal, the fractional-order disturbance observer is given by

$$\begin{cases} D^\lambda z = -L(D^{\lambda-1}z + LD^{\lambda-1}x) - L(AD^{\lambda-1}x + BD^{\lambda-1}u + D^{\lambda-1}\hat{f}(x)) \\ \hat{d} = z + Lx \end{cases} \quad (3)$$

where  $z$  is the disturbance observer auxiliary vector,  $L$  is the gain matrix,  $-L$  is the Hurwitz matrix with  $n$  distinct eigenvalues,  $\hat{d}$  is the estimate of the disturbance, and  $\hat{f}(x)$  is an estimate of the model uncertainty.

**Theorem 1.** Assume that the disturbance is bounded and its derivative is also bounded, which means that  $\|d(t)\| \leq \varsigma$ ,  $\|\dot{d}(t)\| \leq \zeta$ . Define the estimation error of the disturbance  $e_d(t) = d(t) - \hat{d}(t)$ . Based on the structure of the designed disturbance observer, the estimation error of the disturbance is bounded when the system uncertainty estimation error  $\tilde{f}(x)$  is bounded, which means that  $\|e_d(t)\| \leq \xi$ , where  $\xi$  is a very small constant greater than zero.

**Proof.** Define  $M = -L$ ; then,  $M$  is a Hurwitz matrix with  $n$  distinct eigenvalues, so there exists an invertible matrix  $X$  such that  $X^{-1}MX = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$ . Therefore, there exists a positive constant  $\sigma$  such that  $\|e^{Mt}\| \leq \sigma e^{\lambda_{\max}(M)t}$ , where  $\sigma = \|X^{-1}\| \|X\|$  [34]. Differentiating the estimation error  $e_d(t)$  for the disturbance yields the following equation:

$$\begin{aligned} \frac{d(e_d(t))}{dt} &= \dot{d}(t) - \dot{\hat{d}}(t) = \dot{d}(t) - \frac{d}{dt}(z + Lx) \\ &= \dot{d}(t) + L(z + Lx) + L(Ax + Bu + \hat{f}(x)) - L(Ax + Bu + f(x) + d) \\ &= \dot{d}(t) + L\hat{d}(t) - Ld(t) + L\hat{f}(x) - Lf(x) \\ &= \dot{d}(t) - Le_d(t) - L\tilde{f}(x) \end{aligned}$$

It can be obtained that  $\dot{e}_d(t) = \dot{d}(t) + Me_d(t) + M\tilde{f}(x)$ . The subsequent proof is given in the stability analysis on  $e_d(t)$ .  $\square$

For the model uncertainty that may exist in the system, a radial basis function (RBF) network is used for approximation. The network structure of an RBF has the ability for high-speed learning, the ability to approximate a nonlinear function, which improves the model's fitting ability, and it can adapt to a variety of complex input–output mapping, so it is very flexible in practical applications. Moreover, compared with other types of neural networks, the RBF network structure is relatively simple, which makes it easy to realize and adjust [35]. The structure of the RBF network used in this paper is shown in Figure 1.

The input–output relationship for each layer can be expressed as an input layer,  $\theta_i = x_i$ , ( $i = 1, 2, \dots, n$ ), an implicit layer,  $h_j = e^{net_j}$ ,  $net_j = -\sum_{i=1}^n \frac{\|\theta_i - c_j\|^2}{b_j^2}$ , ( $j = 1, 2, \dots, m$ ), and an output layer,  $Y_i = W_i^T H = \omega_{i1}h_1 + \omega_{i2}h_2 + \dots + \omega_{im}h_m$ , where  $\|\cdot\|$  is the Euclidean paradigm,  $H$  is an implicit layer function vector  $H = [h_1 h_2 \dots h_m]^T$ ,  $c_j$  is the center vector,  $b_j$  is the width of the radial basis function, and  $W_i$  is the  $i$ th set of column vectors of the weight matrix  $W$  from the implicit layer to the output layer, which means that

$$W = \begin{bmatrix} \omega_{11} & \omega_{12} & \cdots & \omega_{1m} \\ \omega_{21} & \omega_{22} & \cdots & \omega_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \omega_{n1} & \omega_{n2} & \cdots & \omega_{nm} \end{bmatrix}^T, \text{ where } i \text{ is the number of neurons in the input and output layer, and } j \text{ is the number of neurons in the hidden layer.}$$

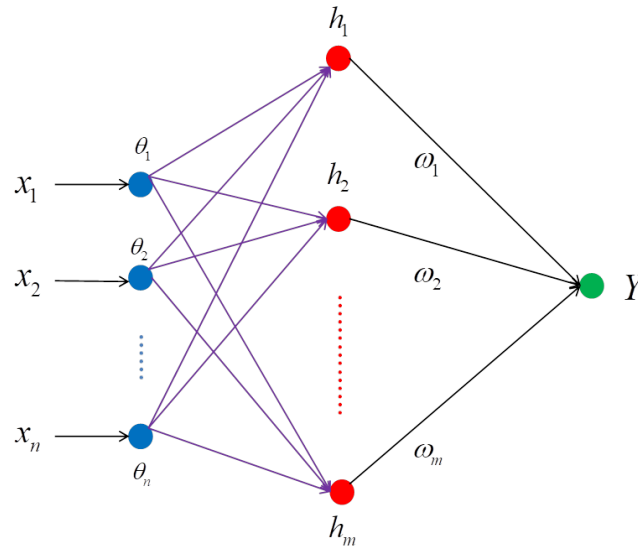


Figure 1. Structure of the RBF network.

Defining the optimal weights vector as  $W^*$ , the system uncertainty part function  $f(x)$  can be expressed as [36]

$$f(x) = W^{*T}H + \varepsilon \tag{4}$$

where  $\varepsilon = [\varepsilon_1 \ \varepsilon_2 \ \cdots \ \varepsilon_n]^T$  is the smallest approximation error of the RBF network,  $H = \aleph(x, c, b)$ ,  $\aleph(\cdot)$  is the arithmetic function from the input layer to the output of the implicit layer; since the system uncertainty  $f(x)$  is bounded, there exists an upper bound  $W_{max}$  for  $W^*$ , which means that  $\|W^*\| \leq W_{max}$ .

Defining the estimate of the RBF network for the system uncertainty  $f(x)$  as  $\hat{f}(x) = \hat{W}^T H$ , the estimation error can be written as [36]

$$\begin{aligned} \tilde{f}(x) &= f(x) - \hat{f}(x) \\ &= W^{*T}H - \hat{W}^T H + \varepsilon \\ &= (W^{*T} - \hat{W}^T)H + \varepsilon \\ &= \check{W}^T H + \varepsilon \end{aligned} \tag{5}$$

where  $\varepsilon$  satisfies  $\|\varepsilon\| \leq \varepsilon_{max}$ , with  $\varepsilon_{max}$  a bounded constant.

Based on the fractional-order disturbance observer's estimate of the disturbance signal  $\hat{d}(x)$  and the RBF network's estimate of the system's uncertainty  $\hat{f}(x)$ , the sliding-mode surface of the fractional-order sliding-mode controller is given by

$$s = C_1 e + C_2 D^{\lambda-1} e \tag{6}$$

where  $e$  is the system tracking error,  $e = y_d - y$ , determined by the desired output  $y_d$  and the actual output  $y$  of the system,  $C_1$  and  $C_2$  are positive real numbers, and  $0 < \lambda < 1$  is the order of the fractional-order differentiation.

Deriving the sliding-mode surface and taking the sliding-mode convergence law as  $-ks - k_s \text{sgn}(s)$ , where  $k_s$  and  $k$  are the sliding-mode convergence law parameters and  $k_s, k > 0$ , and substituting into (1), the control law can be obtained by

$$u = (CB)^{-1} \left( \dot{y}_d + \frac{C_2}{C_1} D^\lambda e + \frac{ks + k_s \text{sgn}(s)}{C_1} \right) - B^{-1}Ax - B^{-1}d - B^{-1}f(x) \quad (7)$$

Substituting the disturbance signal estimator of (3) with the system uncertainty estimator  $\hat{f}(x)$  yields the following final control law:

$$u = (CB)^{-1} \left( \dot{y}_d + \frac{C_2}{C_1} D^\lambda e + \frac{ks + k_s \text{sgn}(s)}{C_1} \right) - B^{-1}Ax - B^{-1}\hat{d} - B^{-1}\hat{f}(x) \quad (8)$$

### 3. TD3 Algorithm Based on Fractional-Order Control Method

In this section, a TD3 algorithm based on the designed fractional-order control method is proposed, and we design the reward function and loss function of the TD3 algorithm based on the proposed fractional-order control method, which in turn makes it effective for the optimization of the parameters of the fractional-order control method.

As a kind of reinforcement learning algorithm, the basic principle of the TD3 algorithm is also composed of a critic and an actor, two kinds of networks of the agent and the environment for the interactive operation. More specifically, Figure 2 shows the agent interacting with the environment to obtain the state  $s_t$  and rewards  $r_t$ , the output of the action  $a_t$  at moment  $t$ , and the update of the two kinds of networks within the agent is based on the loss function.

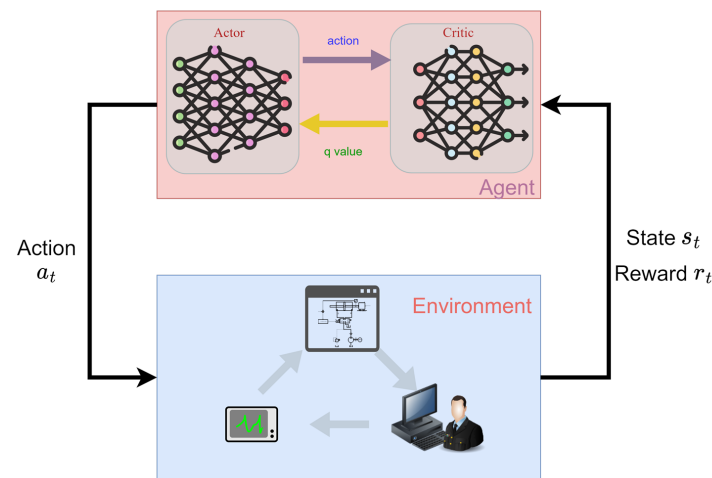


Figure 2. The relationship between environment and agent.

In this paper, the state of the environment  $s_t$  is set as the error signal for each state of the system, which means that  $s_t = [e_1(t) \ e_2(t) \ \dots \ e_n(t)]^T$ , and the reward signal  $r_t$  is set to the negative of the weighted sum of the absolute values of the error signals for each state of the system, which means  $r_t = -\sum_{i=1}^n \eta_i |e_i(t)|$ , where  $\eta_i > 0$  denotes the weight of the  $i$ th error signal in the reward function, which needs to be set according to the actual situation; the action  $a_t$  of the actor network's output are the parameters of the fractional-order controller which should be optimized.

In the reinforcement learning algorithm, the Bellman equation is defined as [37]

$$U_t = r_t + \gamma U_{t+1} \quad (9)$$

where  $r_t$  denotes the reward acquired by the intelligence in the environment,  $U_t = \sum_{k=1}^n \gamma^{k-1} r_{t+k-1}$  is a value function that represents the sum of the rewards that the intelligence expects to acquire in a continuous process after moment  $t$ , and where  $0 < \gamma < 1$  denotes the discount rate.

For both sides of (9), the expectation based on the state  $S$  and action  $A$  at a moment can be obtained, and the value function of the action at that moment can be obtained by

$$E_{\pi}[U_t | S_t = s_t, A_t = a_t] = E_{\pi}(r(s_t, a_t)) + \gamma E_{\pi}[U_{t+1} | S_{t+1} = s_{t+1}, A_{t+1} = a_{t+1}] \quad (10)$$

In the reinforcement learning algorithm, the estimation of the value function of the action is carried out through the critic network, so there is a certain bias, at which point the Bellman equation's equal sign does not hold. Defining the difference between the left and right sides of (10) as the temporal-difference error (TD error)  $\delta_{TD}(t)$ , it can be written as [38]

$$\delta_{TD}(t) = Q_{\pi}(s_t, a_t | \omega_t) - (r(s_t, a_t) + \gamma Q_{\pi}(s_{t+1}, a_{t+1} | \omega_t)) \quad (11)$$

where  $\omega_t$  is the parameter of the critic network at moment  $t$ .

In order to solve the bootstrapping of the overestimation in traditional reinforcement learning algorithms, the TD3 algorithm adopts two sets of critic networks and target critic networks with identical structure and parameters; the corresponding network outputs are called Critic1 network  $Q_{1,\pi}(s, a | \omega_1)$ , Critic2 network  $Q_{2,\pi}(s, a | \omega_2)$ , Target Critic1 network  $Q'_{1,\pi}(s, a | \omega'_1)$ , and Target Critic2 network  $Q'_{2,\pi}(s, a | \omega'_2)$ . The algorithm adopts the strategy of smoothing regularization to reduce the estimation error, adding a noise signal  $\epsilon \sim \text{clip}(\mathcal{N}(0, \sigma); -c, c)$  to the action output from the actor network, so that the target value  $y_{target} = r(s_t, a_t) + \gamma \cdot Q_{\pi}(s_{t+1}, a_{t+1})$  can be expressed as [31]

$$y_{target} = r(s_t, a_t) + \gamma \min_{i=1,2} (Q'_{i,\pi}(s_{t+1}, (a'_{t+1} + \epsilon) | \omega'_i)) \quad (12)$$

The agent in the TD3 algorithm stores each state transfer pair  $(s, a, r, s')$  in the replay buffer during each interaction with the environment, and since in the process of learning by the intelligent body, the selection of the experience should be considered to prioritize the learning value, in this paper, a nonuniform sampling was designed as

$$P(X = r_i) = e^{\alpha(|r_i|)} \left( \sum_{i=1}^n e^{\alpha(|r_i|)} \right)^{-1} \quad (13)$$

where  $P(X = r_i)$  denotes the probability of drawing a state transfer pair corresponding to reward  $r_i$ ,  $\alpha$  is a parameter that controls the shape of the distribution, and  $n$  denotes the total number of samples in the replay buffer.

Based on the sampling probability in (13),  $N$  samples are taken from the replay buffer and the loss function of the critic network is constructed using the mean squared error (MSE), which can be written as

$$J_{\omega}(t) = \sum_{j=1}^2 J_{\omega_j}(t) = -N^{-1} \sum_{j=1}^2 \sum_{i=1}^N \left( Q_{j,\pi}(s, a | \omega_j) - y_{target_{j,i}} \right)^2 \quad (14)$$

For the actor network, in order to maximize the future expected return of the current strategy, which means that  $\pi(a | \theta) = \arg \max_{\theta} E_{s_t, a_t} \left( \sum_{i=t}^{\infty} \gamma^{i-t} R_i \right)$ , and in order to improve the explorability of the action space, the loss function was designed as

$$J_{\theta}(t) = - \left[ N^{-1} \sum_{j=1}^N E_{s_{t_j}, a_{t_j}} \left( \sum_{i=t_j}^{\infty} \gamma^{i-t_j} R_i \right) + \alpha H_{\theta}(\pi) \right] \quad (15)$$

where  $\theta$  is the parameter of the actor network,  $H_\theta(\pi) = \sum_a \pi(s|\theta) \log \pi(s|\theta)$  is a measure of uncertainty in the distribution of strategies, and  $\alpha$  is the weight parameter for that item.

For both critic network and actor network parameters  $\omega$ ,  $\theta$  are updated using the gradient descent method, which can be expressed as [31]

$$\begin{aligned}\omega_{t+1} &= \omega_t - \alpha \nabla_\omega J_\omega(t) \\ \theta_{t+1} &= \theta_t - \beta \nabla_\theta J_\theta(t) \\ &= \theta_t - \beta (N^{-1} \sum \nabla_a Q_\pi(s, a|\omega) \nabla_\theta \pi(s|\theta) + \alpha \nabla_\theta H_\theta(\pi))\end{aligned}\quad (16)$$

where  $\alpha$  and  $\beta$  are the learning rate of the critic network and actor network.

For the target network parameter, a soft update is performed by introducing the update shift  $\tau$  of the network parameter, which means [31]:

$$\begin{aligned}\omega'_{i,t+1} &= \tau \omega_{i,t+1} + (1 - \tau) \omega'_{i,t} \quad (i = 1, 2) \\ \theta'_{t+1} &= \tau \theta_{t+1} + (1 - \tau) \theta'_t\end{aligned}\quad (17)$$

Finally, the TD3 algorithm uses lagged updating for the actor network, which means that the actor network is updated once when the critic network is updated multiple times. Based on the above description, a structure of a fractional-order control method based on the TD3 algorithm is proposed, and the corresponding pseudocode is shown in Algorithm 1.

---

#### Algorithm 1 FOCS pseudocode based on the TD3 algorithm

---

```

Initialize critic networks  $Q_{1,\pi}(s, a|\omega_1)$  and  $Q_{2,\pi}(s, a|\omega_2)$  and actor network  $\pi_\theta$  with
random parameters  $\omega_1, \omega_2, \theta$ 
Initialize target networks  $\omega'_1 \leftarrow \omega_1, \omega'_2 \leftarrow \omega_2, \theta' \leftarrow \theta$ 
Initialize replay buffer  $\mathcal{B}$ 
for  $t = t_0$  to  $t = t_f$  do
  Output action based on current parameters  $a_t \sim \pi_\theta(s_t) + \epsilon, \epsilon \sim \text{clip}(\mathcal{N}(0, \sigma); -c, c)$ ,
  Take  $a_t$  for FOCS, observe the error  $e(t)$ , calculate  $r$ , and observe new state  $s_{t+1}$ 
  Store the transfer pair  $(s, a, r, s')$  in  $\mathcal{B}$ 
  if  $n \geq N \setminus \setminus n$  is the number of transfer pair in replay buffer  $\mathcal{B}$ .
    Sample  $N$  transfer pairs and calculate the loss function of critic network  $J_\omega(t)$ 
    Update  $\omega$  by  $\omega_{t+1} \leftarrow \omega_t - \alpha \nabla_\omega J_\omega(t)$ 
    if Critic network after  $d$  updates
      Calculate the loss function of critic network  $J_\theta(t)$ 
      Update  $\theta$  by  $\theta_{t+1} \leftarrow \theta_t - \beta \nabla_\theta J_\theta(t)$ 
      Update target network by
         $\omega'_{t+1} \leftarrow \tau \omega_{t+1} + (1 - \tau) \omega'_t$ 
         $\theta'_{t+1} \leftarrow \tau \theta_{t+1} + (1 - \tau) \theta'_t$ 
    end if
  end if
end for

```

---

#### 4. Stability Analysis

In this section, the stability of the proposed control method is substantiated through a detailed proof, emphasizing the theoretical feasibility of the proposed approach.

For the proposed fractional-order disturbance observer, the Lyapunov function is defined as

$$V_d = \frac{1}{2} e_d^T e_d \quad (18)$$

The derivation of (18) and substitution into (1), (3), and (5) yield

$$\begin{aligned}
\dot{V}_d &= e_d^T \dot{e}_d = e_d^T (\dot{d} - \hat{d}) = e_d^T (\dot{d} - (\dot{z} + L\dot{x})) \\
&= e_d^T (\dot{d} - (D^{-\lambda+1} D^\lambda z + L(Ax + Bu + f(x) + d))) \\
&= e_d^T (\dot{d} - Le_d - L\tilde{f}(x)) = e_d^T Me_d + e_d^T M\tilde{f}(x) + e_d^T \dot{d} \\
&\leq \lambda_{\max}(M) e_d^T e_d + \lambda_{\max}(M) e_d^T (\tilde{W}^T H) + \lambda_{\max}(M) e_d^T \varepsilon + \frac{1}{2} (e_d^T e_d + \dot{d}^T \dot{d}) \\
&\leq \left( 2\lambda_{\max}(M) + \frac{1}{2} \right) e_d^T e_d + \frac{\lambda_{\max}(M)}{2} \text{tr}(H^T H) \text{tr}(\tilde{W}^T \tilde{W}) + \frac{1}{2} (\lambda_{\max}(M) \varepsilon^T \varepsilon + \dot{d}^T \dot{d}) \\
&\leq \left( 2\lambda_{\max}(M) + \frac{1}{2} \right) e_d^T e_d + \frac{\lambda_{\max}(M)m}{2} \text{tr}(\tilde{W}^T \tilde{W}) + \frac{1}{2} (\lambda_{\max}(M) \varepsilon_{\max}^2 + \zeta^2)
\end{aligned} \tag{19}$$

where  $\lambda_{\max}(M)$  is the largest eigenvalue of  $M$  which is a Hurwitz matrix so it can be obtained that  $\lambda_{\max}(M) < 0$ . In order to make the estimation error of the fractional-order disturbance observer for the disturbance signal stable,  $M$  should satisfy  $\lambda_{\max}(M) < -1/4$ .

In order to keep the approximation error of the designed RBF network for the system uncertainty within a certain range, the adaptive law of the RBF network weights and the implicit layer function update law are taken as

$$\dot{\hat{W}} = \eta^{-1} (-\mathcal{K}\hat{W} - C_1 H s^T C) \tag{20}$$

where  $\eta \in R^{m \times m}$  is the designed positive-definite diagonal matrix,  $\mathcal{K} > 0$  are the designed parameters of the adaptive law,  $s$  and  $C_1$  are the sliding-mode surface and the parameter which was designed in (6), and  $H$  is the implicit layer function vector.

For the proposed RBF network, the Lyapunov function is defined as

$$V_f = \frac{1}{2} \text{tr}(\tilde{W}^T \eta \tilde{W}) \tag{21}$$

Derive Equation (21), and substitute into the RBF network's adaptive law (20):

$$\begin{aligned}
\dot{V}_f &= \text{tr}(\tilde{W}^T \eta \dot{\tilde{W}}) = \text{tr}[\tilde{W}^T \eta (\dot{W}^* - \dot{\hat{W}})] \\
&= -\text{tr}[\tilde{W}^T \eta (\eta^{-1} (-\mathcal{K}\hat{W} - C_1 H s^T C))] \\
&= \text{tr}(\tilde{W}^T \mathcal{K} \hat{W}) + \text{tr}(\tilde{W}^T C_1 H s^T C) \\
&= \mathcal{K} \text{tr}[\tilde{W}^T (W^* - \hat{W})] + \text{tr}(\tilde{W}^T C_1 H s^T C) \\
&\leq -\frac{\mathcal{K}}{2} \text{tr}(\tilde{W}^T \tilde{W}) + \frac{\mathcal{K}}{2} \|W^*\|^2 + \text{tr}(\tilde{W}^T C_1 H s^T C) \\
&\leq -\frac{\mathcal{K}}{2} \text{tr}(\tilde{W}^T \tilde{W}) + \frac{\mathcal{K}}{2} W_{\max}^2 + \text{tr}(\tilde{W}^T C_1 H s^T C)
\end{aligned} \tag{22}$$

Based on the designed sliding-mode surface function, define the Lyapunov function

$$V_s = \frac{1}{2} s^T s \tag{23}$$

Deriving Equation (23) and substituting the system state equation shown in (1) with the control law shown in (8) yield

$$\begin{aligned}
\dot{V}_s &= s^T \dot{s} = s^T (C_1 \dot{e} + C_2 D^\lambda e) = s^T (C_1 (\dot{y}_d - C\dot{x}) + C_2 D^\lambda e) \\
&= s^T (C_1 (\dot{y}_d - C(Ax + Bu + f(x) + d)) + C_2 D^\lambda e) \\
&= s^T \left( C_1 \left( \dot{y}_d - C \left( Ax + B \left( (CB)^{-1} \left( \dot{y}_d + \frac{ks + k_s \text{sgn}(s) + C_2 D^\lambda e}{C_1} \right) - B^{-1} Ax - B^{-1} \hat{d} - B^{-1} \hat{f}(x) \right) + f(x) + d \right) \right) + C_2 D^\lambda e \right) \\
&= s^T (-ks - k_s \text{sgn}(s) - C_1 C \tilde{f}(x) - C_1 C e_d) \\
&= -ks^T s - s^T (k_s \text{sgn}(s) + C_1 C \tilde{f}(x) + C_1 C e_d) \\
&\leq \left( -k + \frac{k_s}{2} + \frac{C_1 \|C\|}{2} \right) s^T s + \frac{C_1 \|C\|}{2} e_d^T e_d + \frac{k_s}{2} - s^T C_1 C (\tilde{W}^T H + \varepsilon)
\end{aligned} \tag{24}$$

To prove the overall stability of the original system with the addition of a fractional-order control method, define the Lyapunov function

$$V = V_d + V_f + V_s \tag{25}$$



A derivation of Equation (25) and a substitution of the Lyapunov derivation of (19), (22), and (24) lead to

$$\begin{aligned}
 \dot{V} &= \dot{V}_d + \dot{V}_f + \dot{V}_s \\
 &\leq \left(2\lambda_{\max}(M) + \frac{C_1\|C\|+1}{2}\right)e_d^T e_d + \left(-k + \frac{k_s}{2} + \frac{C_1\|C\|}{2}\right)s^T s - \left(\frac{\mathcal{K}}{2} - \frac{m\lambda_{\max}(M)}{2}\right)\text{tr}(\tilde{W}^T \tilde{W}) \\
 &\quad + \frac{C_1\|C\|}{2}s^T s + \frac{C_1\|C\|}{2}\varepsilon_{\max}^2 + \frac{\mathcal{K}}{2}W_{\max}^2 + \frac{1}{2}(\zeta^2 + \lambda_{\max}(M)\varepsilon_{\max}^2) + \frac{k_s}{2} \\
 &\leq \left(-k + \frac{k_s}{2} + C_1\|C\|\right)s^T s + \left(2\lambda_{\max}(M) + \frac{C_1\|C\|+1}{2}\right)e_d^T e_d \\
 &\quad - \left(\frac{\mathcal{K}}{2} - \frac{m\lambda_{\max}(M)}{2}\right)\text{tr}(\tilde{W}^T \tilde{W}) + \frac{1}{2}(\zeta^2 + \lambda_{\max}(M)\varepsilon_{\max}^2 + C_1\|C\|\varepsilon_{\max}^2 + \mathcal{K}W_{\max}^2 + k_s) \\
 &\leq -\left(k - \frac{k_s}{2} - C_1\|C\|\right)s^T s - \left(-2\lambda_{\max}(M) - \frac{C_1\|C\|+1}{2}\right)e_d^T e_d \\
 &\quad - \left(\frac{\mathcal{K}}{2} - \frac{m\lambda_{\max}(M)}{2}\right)\text{tr}(\tilde{W}^T \tilde{W}) + \delta \\
 &\leq -\kappa V + \delta
 \end{aligned} \tag{26}$$

where  $\kappa = \min\left(\left(k - \frac{k_s}{2} - C_1\|C\|\right), \left(-2\lambda_{\max}(M) - \frac{C_1\|C\|+1}{2}\right), \left(\frac{\mathcal{K}}{2} - \frac{m\lambda_{\max}(M)}{2}\right)\right)$  to satisfy the system stability, the control system parameter  $0 < C_1 < \frac{-4\lambda_{\max}(M)-1}{\|C\|}$ ,  $k - \frac{k_s}{2} > C_1\|C\|$ ,  $k_s > 0$  should be selected, and  $\delta = \frac{1}{2}(\zeta^2 + \lambda_{\max}(M)\varepsilon_{\max}^2 + C_1\|C\|\varepsilon_{\max}^2 + \mathcal{K}W_{\max}^2 + k_s)$ . At this point, the system is boundedly stable.

### 5. Simulation Result

In this section, in order to validate the effectiveness of the proposed methodology, a valve-controlled hydraulic system is selected as the object of study to design a series of simulations. Firstly, to verify the online learnability of the TD3 algorithm, three rounds of training states during the training process are selected for the comparison. Secondly, the control effect is verified for the optimized fractional-order control method of the TD3 algorithm by employing the prescribed performance fractional-order sliding-mode controller (PPC-FOSMC), an unoptimized fractional-order sliding-mode controller (FOSMC), and a sliding-mode controller (SMC). Furthermore, the antidisturbance is proved for the proposed method, and the simulation verification is carried out for different input signals and disturbance signals. The results show that the fractional-order control method based on the TD3 algorithm not only has good online learning ability and generalization ability but also has a better control effect than the traditional FOSMC, SMC, and PPC-FOSM. Finally, in order to verify that the designed method can still maintain a better control effect under noise disturbance, a Gaussian noise is selected as the disturbance signal to simulate using the online learning method.

#### 5.1. Simulation Results of System Online Learning

A valve-controlled hydraulic system is a third-order system [39] which can be described approximately as

$$\begin{cases} \dot{x} = Ax + Bu + B_d d + f(x) \\ y = Cx \end{cases} \tag{27}$$

where  $x = [x_v, x_s, x_a]^T$  denotes the position, velocity, and acceleration, respectively,  $u$  denotes the control input, and  $y$  denotes the output of the position. The state matrix, input matrix, disturbance matrix, and output matrix  $A, B, B_d$ , and  $C$  are, respectively,

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -R_1 & -R_2 & -R_3 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0 \\ b \end{bmatrix}, \tag{28}$$

$$B_d = [0 \ 0 \ 1]^T, \quad C = [1 \ 0 \ 0],$$

where  $R_1 = \frac{4kk_{ce}\beta}{mV_t}$ ,  $R_2 = \frac{4B_p k_{ce}\beta}{mV_t} + \frac{k}{m}$ ,  $R_3 = \frac{4k_{ce}\beta}{V_t} + \frac{B_p}{m}$ ,  $b = \frac{4k_v k_{sv} k_q \beta A_p}{mV_t}$ , and the values of the parameters of the valve-controlled hydraulic system are shown in Table 1.

**Table 1.** Parameters of the valve-controlled hydraulic system.

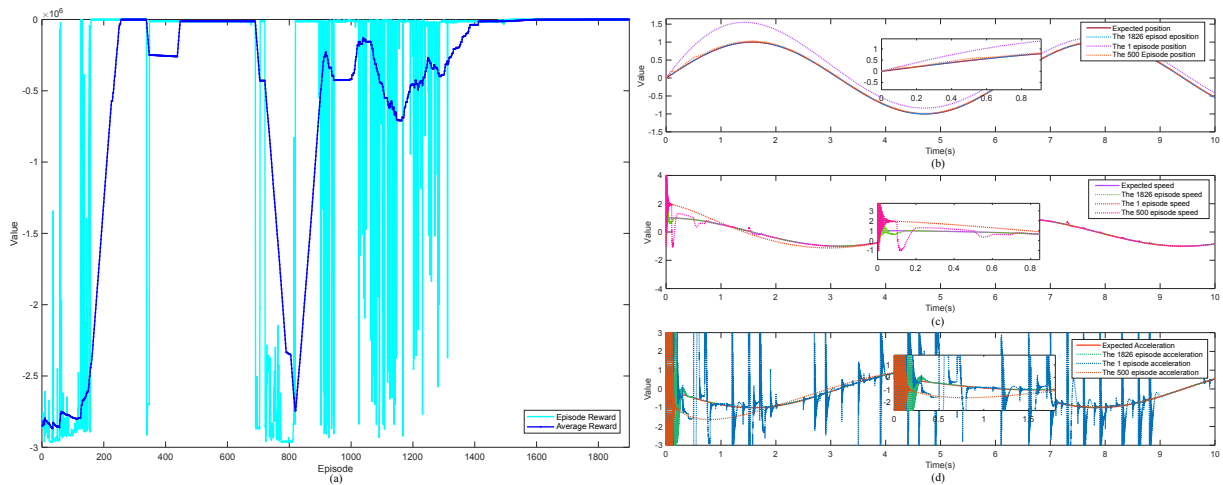
Name	Symbol	Numerical Value
Total mass of piston and load converted to piston	$m$	5 kg
Piston area of hydraulic cylinder	$A_p$	0.4734 m <sup>2</sup>
Springiness	$k$	1
Flow-pressure amplification factor	$k_{ce}$	$3.005 \times 10^{-13}$
Modulus of elasticity of hydraulic fluid	$\beta$	0.005
Total volume of hydraulic cylinder	$V_t$	1.5 m <sup>3</sup>
Viscous damping coefficient of piston and load	$B_p$	5
Servo amplifier gain	$k_v$	1
Servo valve gain	$k_{sv}$	$4.733 \times 10^{-3}$
Flow gain of slide valve	$k_q$	1

The disturbance signal during the TD3 training was set to be  $d = 0.5\sin(5\pi t)$ , the system uncertainty was  $f(x) = \sin(2x_a)\cos(3x_a)$ , and the system expected the output position signal to be  $x_{pd} = \sin(t)$ ; the training parameters are shown in Table 2.

**Table 2.** Parameters of the TD3 algorithm.

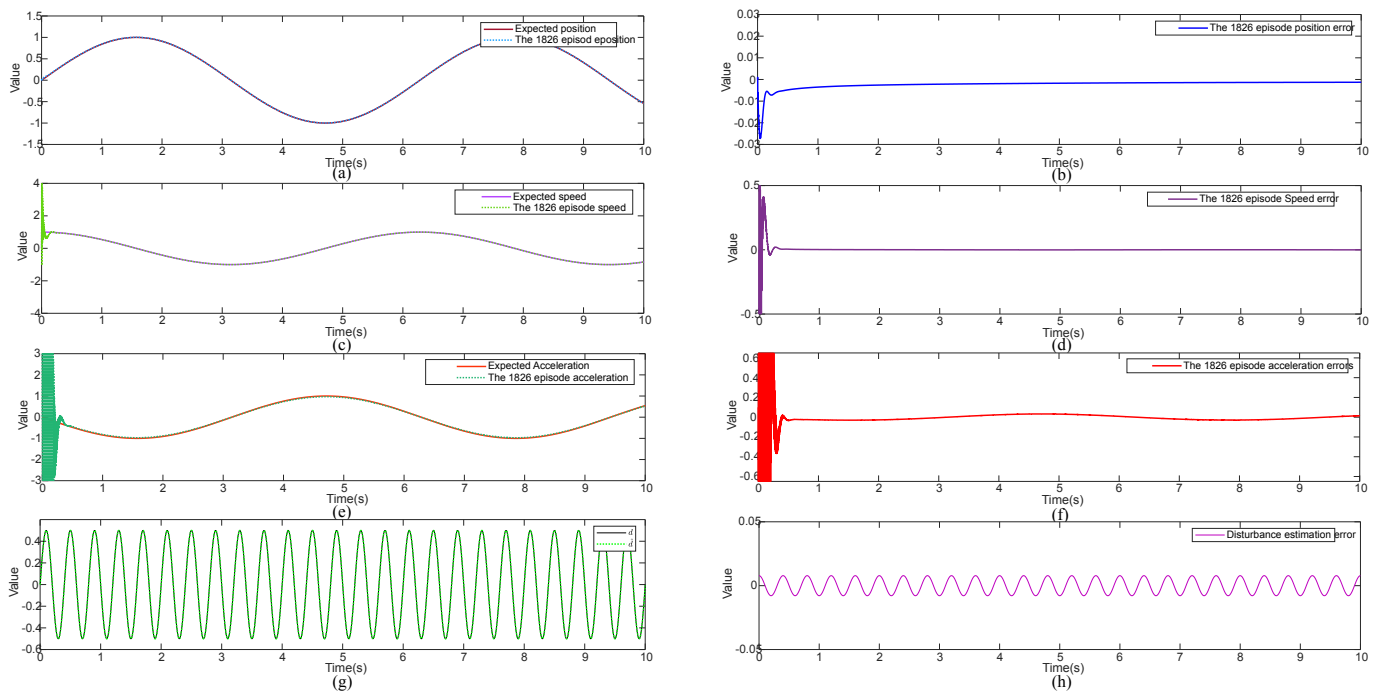
Name	Numerical Value
Episode	1300
Episode step	100
Sampling time	0.1 s
Final time	10 s
Learning rate	0.001
Replay buffer	$1 \times 10^6$
Minimum batch size	128
Number of actor network layers	4
Number of critic network layers	9

Learning to train the system with the above parameters, the episode rewards and average rewards obtained after 1900 episodes are shown in Figure 3a, and Figure 3b–d show the state of the system for the selected episode 1, episode 500, and episode 1826. It can be noticed that in the first 420 episodes, the episode reward fluctuates a lot, and after episode 420, the episode reward and the average reward stabilize for the first time, but at that point, the network still cannot achieve a particularly good training effect. From episode 630 to episode 1300, the agent further explores the action space, and after episode 1300, the episode reward and average reward converge to more optimal values, corresponding to the system state. It can be found that the position and velocity signals of the system corresponding to episode 1 have the worst tracking performance with both large errors and desired signals, while the acceleration signals have the worst tracking performance with more noise. Compared to episode 1, episode 500's systematic position signal can already track the desired signal at the 0.8th second but with a small error, the velocity signal can track the desired signal after 2 s but with a small noise in between, and the acceleration signal can only track the desired signal after the 4th second. Finally, compared to the previous two, episode 1826 is definitely the best performer, with the position signal tracking the desired signal in the first 0.1 s with very little error, and the velocity signal tracking the desired signal with little error after that, despite oscillating in the first 0.1 s, and with the acceleration signal tracking the desired signal after oscillating in the first 0.4 s with an acceptable error.



**Figure 3.** Episode reward and average reward with the training results of the 1826th episode: (a) episode reward and average reward of 1900 episodes, (b) the 1st, 500th, and 1826th episode’s position and expected position signals; (c) the 1st, 500th, and 1826th episode’s speed and expected signals; (d) the 1st, 500th, 1826th episode’s acceleration and expected signals.

In order to better visually represent the training state of set 1826, the system’s individual signals and desired signals and their errors at that point in time are shown in Figure 4. It can be seen that the position, velocity, and acceleration signals of the system have good tracking performance, and the errors are all within a small range. The same is true for the estimation of disturbance signals.

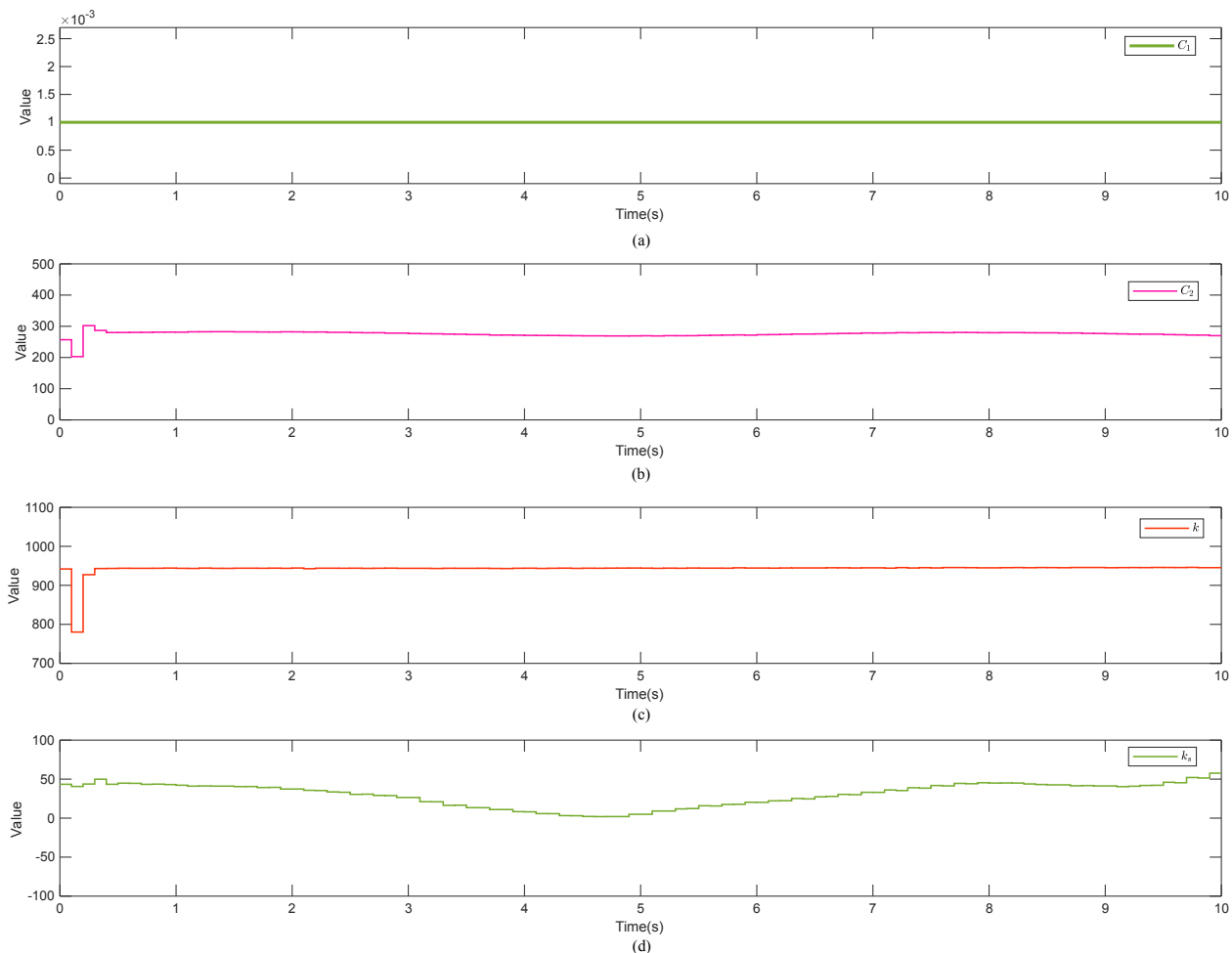


**Figure 4.** The 1st, 500th, and 1826th episode’s system signal and expected signals with the error: (a) position signal and expected signal, (b) position tracking error, (c) speed signal and expected signal, (d) speed tracking error, (e) acceleration signal and expected signal, (f) acceleration tracking error, (g) estimation of the disturbance signal and the actual signal, (h) estimation error of the disturbance signal.

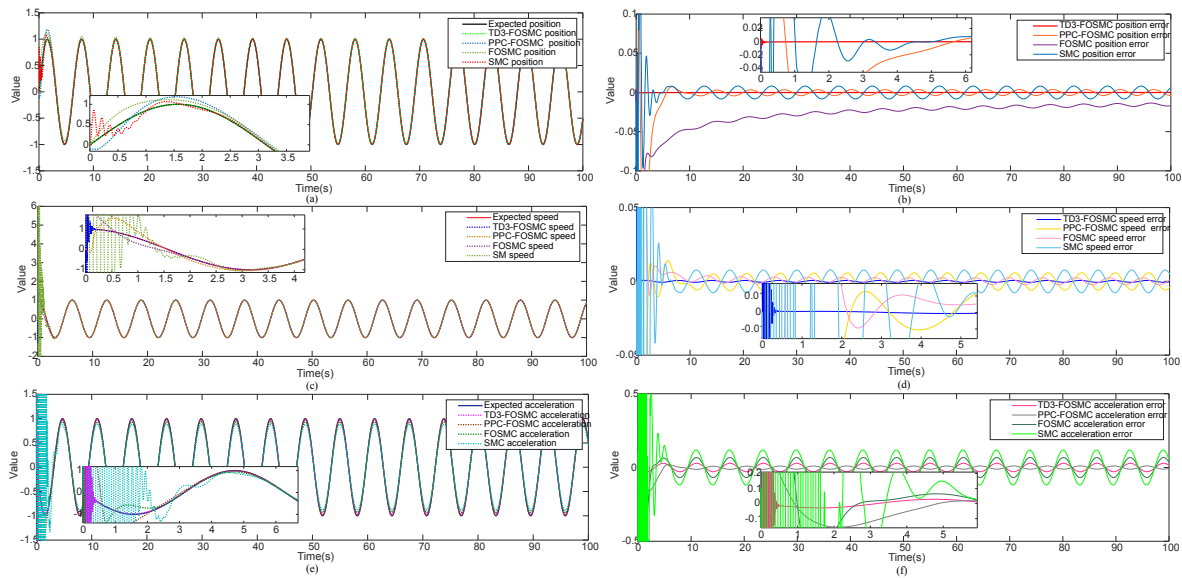
### 5.2. Simulation Results of TD3-FOCS

For the online learning results of the valve-controlled hydraulic fractional-order control method, the training results of the 1826th episode are shown in Figure 5 with  $C_1 = 0.001$ ,  $C_2 = 290$ ,  $k = 945$ ,  $k_s = 43$ ,  $\lambda = 0.55$  selected as the controller parameters, and the parameters of FODOB and the RBF network were set to  $L = 157$ ,  $\mathcal{K} = 7$ , and  $\eta = I$ , which is a single-unit diagonal matrix for the subsequent comparative simulation experiments; the simulation time was set to 100 s, and the final results are shown in Figure 6.

As can be seen from Figure 6, for the position signal, the proposed TD3-FOSMC had the fastest convergence speed (first 0.03 s) and maintained the final steady-state error in the range of  $(-0.001, +0.001)$ , which was the best among the experimental results. For the velocity signal, it also had the fastest convergence speed (first 0.25 s) and maintained the final steady-state error in the range of  $(-0.005, +0.005)$ , which was the best performance among all the experimental methods. For acceleration signals, although the prescribed performance of the fractional-order sliding-mode controller ended up with a smaller error range, the convergence time was greater than the proposed TD3-FOSMC, which converged in the first 0.3 s. Combining all the above control effects, it can be concluded that the proposed TD3-FOSMC has a better speed and stability.



**Figure 5.** Training results of the 1826th episode: (a) parameter  $C_1$ , (b) parameter  $C_2$ , (c) parameter  $k$ , (d) parameter  $k_s$ .



**Figure 6.** TD3-optimized fractional-order control vs. unoptimized control method's system signals and expected signals and their errors: (a) position signal and expected signal, (b) position tracking error, (c) speed signal and expected signal, (d) speed tracking error, (e) acceleration signal and expected signal, (f) acceleration tracking error.

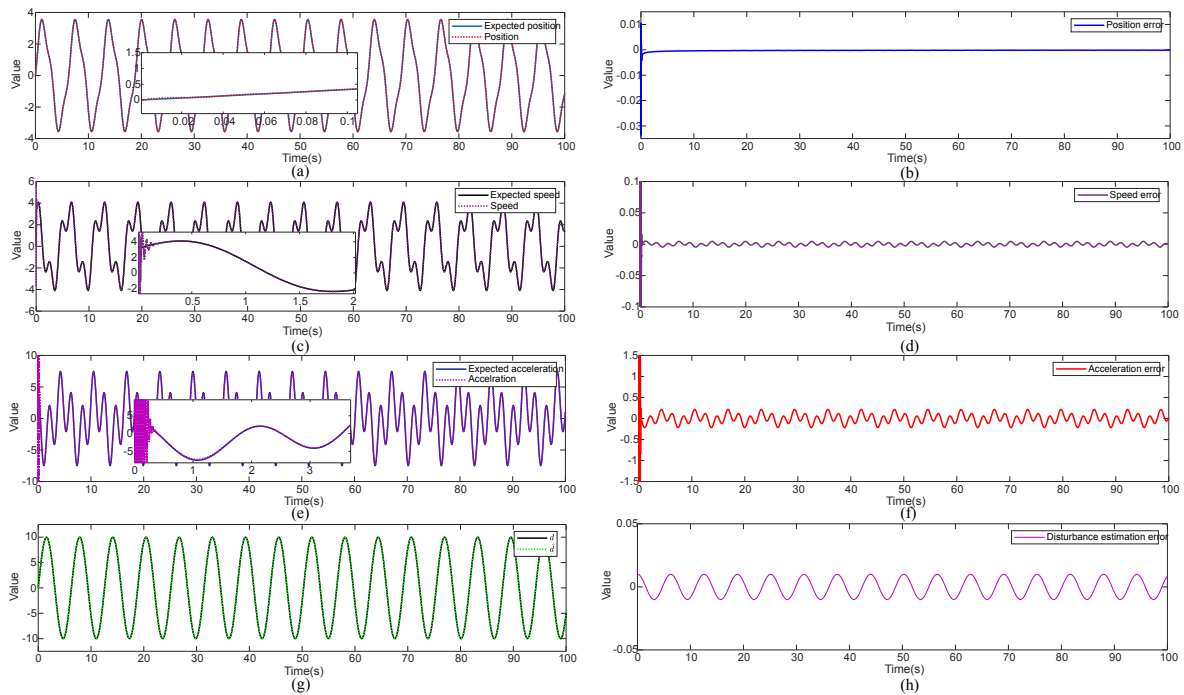
### 5.3. Simulation Results of Different Control Signals and Disturbance Signals

In order to demonstrate the robustness and versatility of the method, two control signals (a triangular wave composite signal and a step signal) and two disturbance signals (a strong sine wave signal disturbance and a triangular wave composite signal) were selected for the simulation verification. This allowed a more comprehensive assessment of the method's antidisturbance performance under different control and disturbance conditions, thus demonstrating the effectiveness of the method in mitigating the effects of an external disturbance in a variety of practical situations.

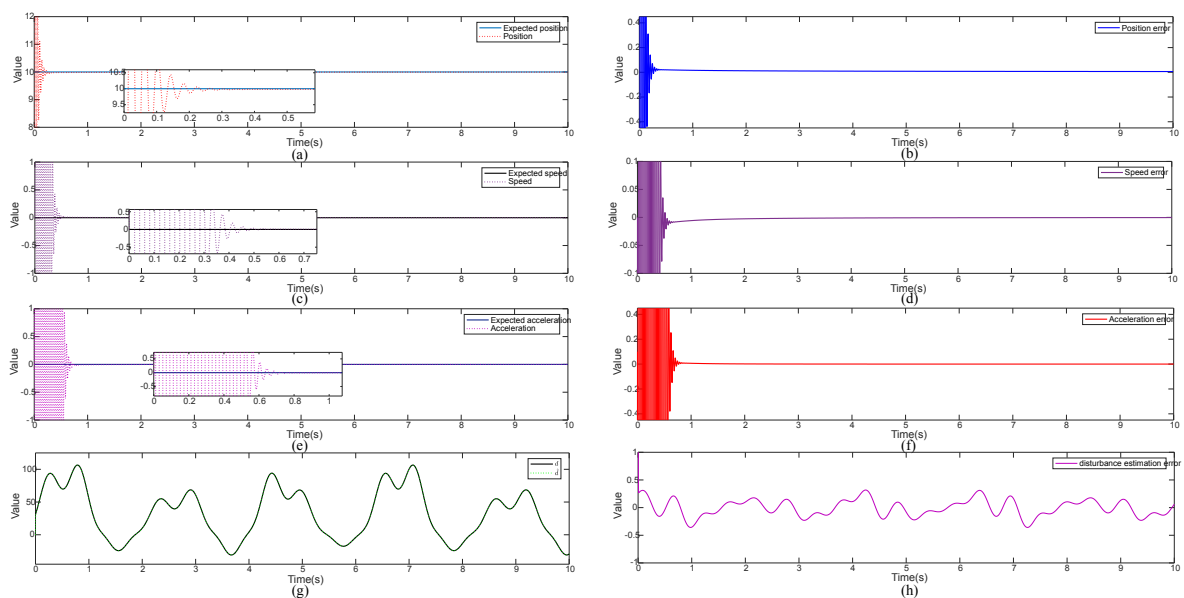
As observed in Figure 7, the simulation results revealed the capability of accurately tracking the desired signal under the influence of a triangular wave composite signal and a strong sine wave signal disturbance. Specifically, the position signal achieved tracking within a remarkably short duration of 0.07 s, with very small overshoots (less than 1%) in the first 0.02 s, and the final steady-state error confined to a narrow range of  $(-0.001, 0.001)$ . In the first 0.3 s, the velocity signal had large to small fluctuations in the range of  $(-3.7, 6)$ ; after that, it achieved synchronization with the desired signal and kept the steady-state error within the range of  $(-0.01, 0.01)$ . On the other hand, the acceleration signal fluctuated in the range of  $(-12, 12)$  for the first 0.5 s and was after synchronized with the desired signal and kept the steady-state error within the range of  $(-0.25, 0.25)$ . The results show that despite the large overshoot in the system response for the different input signals and disturbance signals, the system still had good stability and fast performance.

Analyzing the results shown in Figure 8, when subjected to a step signal and a strong triangular wave composite signal disturbance during the simulation, the obtained results exhibited a marginally reduced performance compared to the previous scenario. Taking the simulation time of 10 s as an example, it can be found that for the position, velocity, and acceleration signals, there were large fluctuations in the first 1 s, and their stabilization time was slowing down sequentially. For the position signal, there were fluctuations within the range of  $(5, 15)$  within the first 0.3 s; after that, it could track the desired signal, and the final steady-state error was limited to the range of  $(0, 0.004)$ . The velocity signal exhibited fluctuations within the range of  $(-10, 10)$  during the first 0.5 s, after which it synchronized with the desired signal, and the final steady-state error was constrained to  $(-0.002, 0)$ . For the acceleration signal, fluctuations within the range of  $(-20, 20)$  occurred during the initial 0.8 s, following which it converged to the desired value. The final steady-state error

remained confined to a smaller range of  $(-0.0001, 0.0001)$ . Despite a slight decrease in performance, the results underscore the method's ability to maintain effective control even in the presence of challenging input signals and disturbance conditions.



**Figure 7.** Simulation results of triangular wave composite signals and strongly interfering sine wave signals: (a) position signal and expected signal, (b) position tracking error, (c) speed signal and expected signal, (d) speed tracking error, (e) acceleration signal and expected signal, (f) acceleration tracking error, (g) estimation of the disturbance signal and actual signal, (h) estimation error of the disturbance signal.

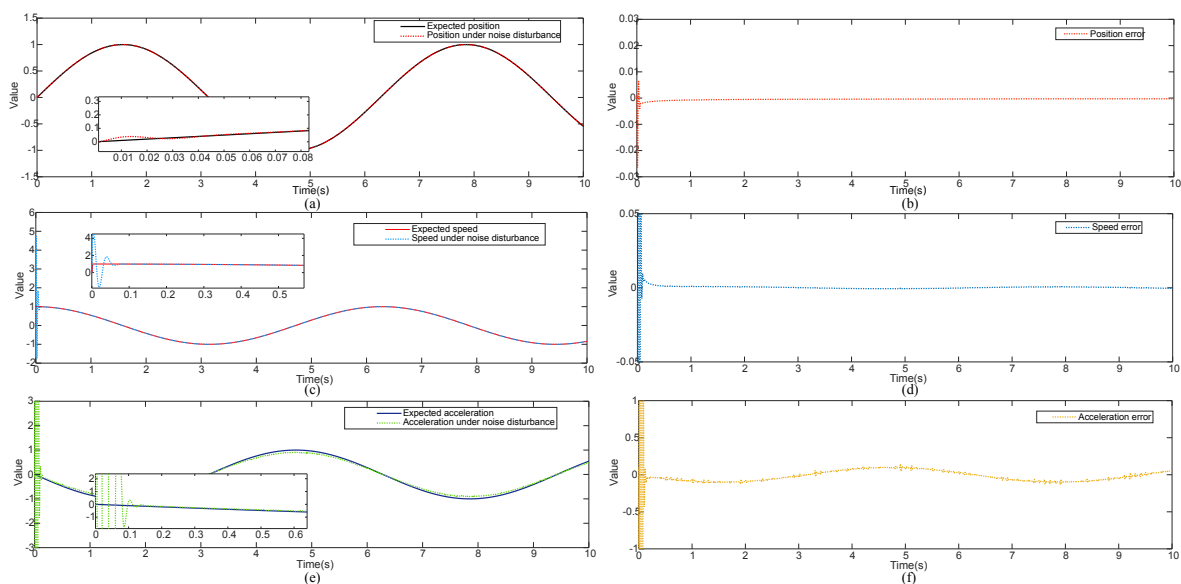


**Figure 8.** Simulation results of step signals and triangular wave composite signal: (a) position signal and expected signal, (b) position tracking error, (c) speed signal and expected signal, (d) speed tracking error, (e) acceleration signal and expected signal, (f) acceleration tracking error, (g) estimation of the disturbance signal and actual signal, (h) estimation error of the disturbance signal.

### 5.4. Simulation Results under Noise Disturbance

The control performance is illustrated for the method under noise disturbance. The noise disturbance signal was selected with a mean value of zero, a variance of five, and a frequency of 10 Hz, and the simulation verification was carried out by using the online learning of the agent. The results are shown in Figure 9.

Analyzing Figure 9, it can be seen that the position signal exhibited minimal susceptibility from the noise disturbance signal, with an overshoot of less than 3% observed only within the initial 0.03 s. Subsequently, the position signal could track the desired signal, and the final steady-state error could be maintained within the range of  $(-0.001, 0)$ . The velocity signal demonstrated minimal susceptibility to noise disturbances, with oscillations confined to the range of  $(-1.6, 4.7)$  within the initial 0.08 s. Following that period, the signal could adeptly track the desired trajectory, and the final steady-state error could satisfy the narrow bounds of  $(-0.0015, 0.0015)$ . Comparing with the earlier online learning outcomes, the performance of the acceleration signal was degraded under the noise signal. The acceleration signal was oscillatory within the range of  $(-6, 6)$  during the initial 0.15 s. Despite these challenges, it could approximately track the desired signal. However, due to the influence of noise, it maintained a final steady-state error within the range of  $(-0.13, 0.13)$ . Thus, based on the analysis above, the proposed control method could achieve the control of system within the bounded range under the noise signal.



**Figure 9.** Simulation results under noise disturbance: (a) position signal and expected signal, (b) position tracking error, (c) speed signal and expected signal, (d) speed tracking error, (e) acceleration signal and expected signal, (f) acceleration tracking error.

## 6. Conclusions

In this paper, a fractional-order control method based on the TD3 algorithm was introduced. A fractional-order disturbance observer was designed to estimate the system's disturbance signal, and an RBF network was selected to approximate the uncertainties in the system, and the fractional-order sliding-mode control method was also adopted to design the controller. A valve-controlled hydraulic system was simulated and validated in Matlab/Simulink using the agent online learning and the optimization parameters. Different control signals and disturbance signals were used in the optimized fractional-order control system. The results showed that the limitations of this method mainly lay in the difficulty of setting up the training environment comprehensively. Despite some shortcomings, the proposed method was generally fast and had good antidisturbance ability.

**Author Contributions:** Conceptualization, G.J. and S.S.; methodology, G.J. and Z.A.; software, G.J. and D.S.; validation, G.J. and D.S.; formal analysis, G.J. and Z.A.; writing—original draft preparation, G.J.; writing—review and editing, G.J. and S.S.; supervision, S.S.; funding acquisition, S.S. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the Fund of Science and Technology on Space Intelligent Control Laboratory Foundation (grant number HTKJ2023KL502002); the China Postdoctoral Science Foundation (grant number 2020M681587); the Jiangsu Province Postdoctoral Science Foundation (grant number 2020Z112).

**Data Availability Statement:** Data are contained within the article.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

- Spanos, P.D.; Malara, G. Nonlinear vibrations of beams and plates with fractional derivative elements subject to combined harmonic and random excitations. *Probabilistic Eng. Mech.* **2020**, *59*, 142–149. [\[CrossRef\]](#)
- Magin, R.L. Fractional calculus models of complex dynamics in biological tissues. *Comput. Math. Appl.* **2010**, *36*, 1586–1593. [\[CrossRef\]](#)
- Atangana, A. Non validity of index law in fractional calculus: A fractional differential operator with Markovian and non-Markovian properties. *Phys. A Stat. Mech. Its Appl.* **2018**, *505*, 688–706. [\[CrossRef\]](#)
- Gonzalez, E.A.; Petráš, I. Advances in fractional calculus: Control and signal processing applications. In Proceedings of the 2015 16th International Carpathian Control Conference (ICCC), Szilvasvarad, Hungary, 27–30 May 2015; pp. 147–152. [\[CrossRef\]](#)
- Warrier, P.; Shah, P. Optimal Fractional PID Controller for Buck Converter Using Cohort Intelligent Algorithm. *Appl. Syst. Innov.* **2021**, *4*, 50. [\[CrossRef\]](#)
- Razzaghian, A. A fuzzy neural network-based fractional-order Lyapunov-based robust control strategy for exoskeleton robots: Application in upper-limb rehabilitation. *Math. Comput. Simul.* **2022**, *193*, 567–583. [\[CrossRef\]](#)
- Fei, J.; Wang, Z.; Pan, Q. Self-Constructing Fuzzy Neural Fractional-Order Sliding Mode Control of Active Power Filter. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, *34*, 10600–10611. [\[CrossRef\]](#)
- Jakovljević, B.; Pisano, A.; Rapaić, M.R.; Usai, E. On the sliding-mode control of fractional-order nonlinear uncertain dynamics. *Int. J. Robust Nonlinear Control* **2016**, *24*, 782–798. [\[CrossRef\]](#)
- Mirrezapour, S.Z.; Zare, A.; Hallaji, M. A new fractional sliding mode controller based on nonlinear fractional-order proportional integral derivative controller structure to synchronize fractional-order chaotic systems with uncertainty and disturbances. *J. Vib. Control* **2022**, *28*, 773–785. [\[CrossRef\]](#)
- Deepika, D. Hyperbolic uncertainty estimator based fractional-order sliding mode control framework for uncertain fractional-order chaos stabilization and synchronization. *ISA Trans.* **2022**, *123*, 76–86. [\[CrossRef\]](#) [\[PubMed\]](#)
- Delavari, H.; Ghaderi, R.; Ranjbar, A.; Momani, S. Fuzzy fractional-order sliding mode controller for nonlinear systems. *Commun. Nonlinear Sci. Numer. Simul.* **2010**, *15*, 963–978. [\[CrossRef\]](#)
- Sun, G.; Ma, Z. Practical tracking control of linear motor with adaptive fractional-order terminal sliding mode control. *IEEE/ASME Trans. Mechatron.* **2017**, *22*, 2643–2653. [\[CrossRef\]](#)
- Djari, A.; Bouden, T.; Boukroune, A. Design of fractional-order sliding mode controller (FSMC) for a class of fractional-order non-linear commensurate systems using a particle swarm optimization (PSO) Algorithm. *J. Control Eng. Appl. Inform.* **2014**, *16*, 46–55.
- Karthikeyan, A.; Rajagopal, K. Chaos control in fractional-order smart grid with adaptive sliding mode control and genetically optimized PID control and its FPGA implementation. *Complexity* **2017**, *2017*, 3815146. [\[CrossRef\]](#)
- Han, S. Modified grey-wolf algorithm optimized fractional-order sliding mode control for unknown manipulators with a fractional-order disturbance observer. *IEEE Access* **2020**, *8*, 18337–18349. [\[CrossRef\]](#)
- Salman, G.A.; Jafar, A.S.; Ismael, A.I. Application of artificial intelligence techniques for LFC and AVR systems using PID controller. *Int. J. Power Electron. Drive Syst.* **2019**, *10*, 1694. [\[CrossRef\]](#)
- Perrusquía, A.; Yu, W. Identification and optimal control of nonlinear systems using recurrent neural networks and reinforcement learning: An overview. *Neurocomputing* **2021**, *438*, 145–154. [\[CrossRef\]](#)
- Liu, Y.J.; Zhao, W.; Liu, L.; Li, D.; Tong, S.; Chen, C.P. Adaptive neural network control for a class of nonlinear systems with function constraints on states. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, *34*, 2732–2741. [\[CrossRef\]](#) [\[PubMed\]](#)
- Chen, Z.; Niu, B.; Zhang, L.; Zhao, J.; Ahmad, A.M.; Alassafi, M.O. Command filtering-based adaptive neural network control for uncertain switched nonlinear systems using event-triggered communication. *Int. J. Robust Nonlinear Control* **2022**, *32*, 6507–6522. [\[CrossRef\]](#)
- Katz, S.M.; Corso, A.L.; Strong, C.A.; Kochenderfer, M.J. Verification of image-based neural network controllers using generative models. *J. Aerosp. Inf. Syst.* **2022**, *19*, 574–584. [\[CrossRef\]](#)
- Kiumarsi, B.; Vamvoudakis, K.G.; Modares, H.; Lewis, F.L. Optimal and autonomous control using reinforcement learning: A survey. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *29*, 2042–2062. [\[CrossRef\]](#)



22. Duraisamy, P.; Nagarajan, Santhanakrishnan, M.; Rengarajan, A. Design of deep reinforcement learning controller through data-assisted model for robotic fish speed tracking. *J. Bionic Eng.* **2023**, *20*, 953–966. [[CrossRef](#)]
23. Wei, C.; Xiong, Y.; Chen, Q.; Xu, D. On adaptive attitude tracking control of spacecraft: A reinforcement learning based gain tuning way with guaranteed performance. *Adv. Space Res.* **2023**, *71*, 4534–4548. [[CrossRef](#)]
24. Barto, A.G. Reinforcement learning. In *Neural Systems for Control*; Academic Press: Cambridge, MA, USA, 1997; pp. 7–30.
25. Nian, R.; Liu, J.; Huang, B. A review on reinforcement learning: Introduction and applications in industrial process control. *Comput. Chem. Eng.* **2020**, *139*, 106886. [[CrossRef](#)]
26. Zhu, Y.; Zhao, D.; Li, X. Using reinforcement learning techniques to solve continuous-time non-linear optimal tracking problem without system dynamics. *IET Control Theory Appl.* **2016**, *10*, 1339–1347. [[CrossRef](#)]
27. Syafiie, S.; Tadeo, F.; Martinez, E. Model-free learning control of neutralization processes using reinforcement learning. *Eng. Appl. Artif. Intell.* **2007**, *20*, 767–782. [[CrossRef](#)]
28. Dogru, O.; Velswamy, K.; Ibrahim, F.; Wu, Y.; Sundaramoorthy, A.S.; Huang, B.; Xu, S.; Nixon, M.; Bell, N. Reinforcement learning approach to autonomous PID tuning. *Comput. Chem. Eng.* **2022**, *161*, 107760. [[CrossRef](#)]
29. Taherian, S.; Kuutti, S.; Visca, M.; Fallah, S. Self-adaptive torque vectoring controller using reinforcement learning. In Proceedings of the 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), Indianapolis, IN, USA, 19–22 September 2021; IEEE: Piscataway, NJ, USA, 2021; pp 172–179.
30. Yan, L.; Webber, J.L.; Mehbodniya, A.; Moorthy, B.; Sivamani, S.; Nazir, S.; Shabaz, M. Distributed optimization of heterogeneous UAV cluster PID controller based on machine learning. *Comput. Electr. Eng.* **2022**, *101*, 108059. [[CrossRef](#)]
31. Wu, J.; Wu, Q.J.; Chen, S.; Pourpanah, F.; Huang, D. A-TD3: An Adaptive Asynchronous Twin Delayed Deep Deterministic for Continuous Action Spaces. *IEEE Access* **2022**, *10*, 128077–128089. [[CrossRef](#)]
32. Dankwa, S.; Zheng, W. Twin-delayed ddpg: A deep reinforcement learning technique to model a continuous movement of an intelligent robot agent. In Proceedings of the 3rd International Conference on Vision, Image and Signal Processing, Vancouver, BC, Canada, 26–28 August 2019; pp. 1–5.
33. Li, C.; Qian, D.; Chen, Y.Q. On Riemann-Liouville and caputo derivatives. *Discret. Dyn. Nat. Soc.* **2011**, *2011*, 562494. [[CrossRef](#)]
34. Walsh, G.C.; Ye, H.; Bushnell, L.G. Stability analysis of networked control systems. *IEEE Trans. Control Syst. Technol.* **2002**, *10*, 438–446. [[CrossRef](#)]
35. Er M.J.; Wu, S.; Lu, J.; Toh, H.L. Face recognition with radial basis function (RBF) neural networks. *IEEE Trans. Neural Netw.* **2002**, *13*, 697–710. [[PubMed](#)]
36. Fei, J.; Wang, H.; Fang, Y. Novel neural network fractional-order sliding-mode control with application to active power filter. *IEEE Trans. Syst. Man, Cybern. Syst.* **2021**, *52*, 3508–3518. [[CrossRef](#)]
37. Fei, Y.; Yang, Z.; Chen, Y.; Wang, Z. Exponential bellman equation and improved regret bounds for risk-sensitive reinforcement learning. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 20436–20446.
38. Doya, K. Reinforcement learning in continuous time and space. *Neural Comput.* **2000**, *12*, 219–245. [[CrossRef](#)]
39. Duan, Z.; Sun, C.; Li, J.; Tan, Y. Research on servo valve-controlled hydraulic motor system based on active disturbance rejection control. *Meas. Control.* **2024**, *57*, 113–123. [[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.