Hindawi
Complexity
Volume 2020, Article ID 2837916, 19 pages
https://doi.org/10.1155/2020/2837916

Research Article

Attitude Control with Auxiliary Structure Based on Adaptive Dynamic Programming for Reentry Vehicles

Xu Li, Zhongtao Cheng, Bo Wang, Yongji Wang, and Lei Liu

National Key Laboratory of Science and Technology on Multispectral Information Processing, School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan 430074, China

Correspondence should be addressed to Zhongtao Cheng; ztcheng@hust.edu.cn

Received 6 April 2020; Revised 8 August 2020; Accepted 19 August 2020; Published 2 September 2020

Academic Editor: Raúl Villafuerte-Segura

This paper presents an attitude control scheme combined with adaptive dynamic programming (ADP) for reentry vehicles with high nonlinearity and disturbances. Firstly, the nonlinear attitude dynamics is divided into inner and outer loops according to the time scale separation and the cascade control principle, and a general sliding mode control method is employed to construct the main controllers for the double loops. Considering the shortage of main controllers in handling nonlinearity and sudden disturbances, an ADP structure is introduced into the outer attitude loop as an auxiliary. And the ADP structure utilizes neural network estimators to minimize the cost function and generate optimal signals through online learning, so as to compensate defect of the main controllers' adaptability speed and accuracy. Then, the stability is analyzed by the Lyapunov method, and the parameter selection strategy of the ADP structure is derived to guide implementation. In addition, this paper puts forward skills to speed up ADP training. Finally, simulation results show that the control strategy with ADP possesses stronger adaptability and faster response than that without ADP for the nonlinear vehicle system.

1. Introduction

Attitude control for reentry vehicles has been a hotspot in the field of aerospace. The complex operating conditions and the high nonlinearity of vehicles themselves bring great challenges to attitude control. Fortunately, around these focuses, researchers continue to explore and ameliorate control schemes, developing a series of available control technologies.

For the control of space vehicles, some schemes have been investigated one after another. Some linear control methods, such as linear parameter varying (LPV) [1] and linear quadratic regulator (LQR), focus on linearizing the aircraft model. However, due to the highly nonlinear and coupling dynamic characteristics, to be honest, the capabilities of these linear control methods on actual nonlinear coupling vehicles are limited. Besides, some nonlinear control methods are widely employed, such as nonlinear dynamic inversion [2], sliding mode control, and backstepping method [3, 4]. Although these nonlinear control techniques can also effectively deal with the nonlinear nature of vehicles, they will still be slightly embarrassed and lack adaptability in the face of complex and changeable disturbances if without other auxiliary means. Therefore, in the recent development of vehicle control, more and more adaptive technologies have been favoured by researchers [5].

For the purpose of ameliorating the robustness of the controller by designing adaptive mechanism, observer-based adaptive control technology and other intelligent methods (as adaptive fuzzy control and iterative learning) have emerged one after another [6–8]. Especially, in recent years, thanks to the vigorous development of new artificial intelligence, reinforcement learning (RL) has attracted more and more attention, which has shown strong performance in solving adaptive and optimal control problems [9–11]. In the control domain, reinforcement learning is transformed into approximate or adaptive dynamic programming (ADP), which learns by interacting with the environment to determine what optimal actions to take to minimize a cost function over a period of time [12]. One of the core

approaches is the critic-action (CA) design, which approximates the cost function and obtains the optimal actions by solving the Hamilton–Jacobi–Bellman equation with function estimators [13]. ADP contains a variety of structural classifications, including heuristic [14], dual heuristic [15], and action-dependent dynamic programming (ADHDP), etc., which have been made preliminary explorations and achievements in the field of vehicle control [16]. Specifically, Luo et al. developed a direct heuristic dynamic programming (dHDP) for longitudinal control of hypersonic vehicles and introduced fuzzy neural networks to enhance the learning ability and robustness of dHDP [17]. There is also an application of ADDHP to study the optimal control of attitude maneuver for three-axis spacecraft [18]. Some creative researchers improve ADP by redefining the two optimization objectives and apply ADP to the in-orbit reconfiguration of the vehicle attitude system under multitask constraints through dual optimization indexes [19]. Moreover, ADP can be associated with traditional methods, such as nonlinear filter [20] and sliding mode control [21], to implement a data-driven ADHDP auxiliary control scheme for the speed and altitude system of an air-breathing hypersonic vehicle [21]. In [22], a switching adaptive active anti-interference control technique based on reduced-order observer technique and ADP is proposed, considering the parameter uncertainty and external disturbance of variable structure near-space vehicles. Furthermore, aiming at the guidance and control problem of the vertical take-off and landing (VTOL) system with multivariable disturbances, an online kernel DHP robust control strategy based on the sparse kernel theory is designed for VTOL vehicles [23]. Most of the above control strategies with ADP utilize neural network estimators to approximate the cost function and optimal control law online, while Zhou et al. creatively put forward an incremental ADP (iADP) combining the advantages of the incremental control method and ADP [24]. This iADP is based on Markov decision-making process and Bellman optimal principle to directly derive the explicit expression of optimal control law, greatly simplifying the design process of ADP, and successfully exploited to satellite [25] and aircraft [26]. Similarly, Sun and van Kampen also come up with an incremental model-based DHP technology for vehicle control, replacing the model network in traditional DHP with an incremental model [27, 28].

In a word, the development of ADP in the field of vehicle control is rapidly deepening and expanding [16], but as far as the current literature is concerned, ADP is still rarely applied to the control of all three channels' attitude angles of the vehicle. Moreover, most of the literature rarely mentions the internal weight convergence, parameter selection, training speed, and other issues of ADP based on critic-action networks, but these are problems to be concerned about. Therefore, this paper contributes to employ the ADP framework to the control of all three-channel attitude angles of a reentry vehicle. Inspired by the ADP as an auxiliary controller [21], this paper presents a framework combining conventional controller and ADP, and ADP is as the auxiliary means to enhance the rapidity and adaptivity of the whole attitude system. In addition, the internal convergence of the ADP structure and its parameter selection rules are discussed in depth. Aiming at the implementation problem, this paper considers the improvement measures to speed up ADP training, which will be provided to interested researchers for future discussion.

The rest of this paper is organized as follows. Firstly, the nonlinear dynamics of the three-channel attitude control system of the reentry vehicle is established in Section 2. Then, in Section 3, the control strategy based on the dual-loop main controller plus ADP is elaborated in detail. In Section 4, some issues about implementation are taken into consideration. Finally, the simulations and conclusions are presented in Sections 5 and 6, respectively.

2. Nonlinear Model

To describe the attitude change of the reentry phase, we give the rotation equations of the vehicle around the center of mass, including rotation dynamics and attitude kinematics. They determine the attitude angles of the vehicle around the center of mass and the angular rate of the three channels during the flight. Considering the influence of Earth rotation on attitude control, a three degree of freedom nonlinear attitude model in the body coordinate system can be obtained [29]:

$$\dot{\alpha} = -p\cos\alpha\tan\beta + q - r\sin\alpha\tan\beta$$

$$+\left(\frac{\sin\mu}{\cos\beta}\right)\left(-\dot{\phi}\sin\chi\sin\vartheta + \dot{\chi}\cos\vartheta + \left(\dot{\theta}+\Omega_{\mathrm{E}}\right)(\cos\phi\cos\chi\sin\vartheta - \sin\phi\cos\vartheta)\right)$$

$$-\left(\frac{\cos\mu}{\cos\beta}\right)\left(\dot{\vartheta} - \dot{\phi}\cos\chi - \left(\dot{\theta}+\Omega_{\mathrm{E}}\right)\cos\phi\sin\chi\right),$$

$$\dot{\beta} = p\sin\alpha - r\cos\alpha + \sin\mu\left(\dot{\vartheta} - \dot{\phi}\cos\chi + \left(\dot{\theta}+\Omega_{\mathrm{E}}\right)\cos\phi\sin\chi\right)$$

$$
\begin{aligned}
&+\cos\mu\left(\left(-\dot{\theta}+\Omega_{\mathrm{E}}\right)(\cos\phi\cos\chi\sin\vartheta-\sin\phi\cos\vartheta)+\dot{\chi}\cos\vartheta-\dot{\phi}\sin\chi\sin\vartheta\right),\\
\dot{\mu}&=-p\cos\alpha\cos\beta-q\sin\beta-r\sin\alpha\cos\beta+\dot{\alpha}\sin\beta\\
&\quad-\dot{\chi}\sin\vartheta-\dot{\phi}\sin\chi\cos\vartheta+\left(\dot{\theta}+\Omega_{\mathrm{E}}\right)(\sin\phi\sin\vartheta+\cos\phi\cos\chi\cos\vartheta),\\
\dot{p}&=\left(\frac{I_{=}M_x}{I_{xx}I_{zz}-I_{xz}^2}\right)+\left(\frac{I_{xz}M_z}{I_{xx}I_{zz}-I_{xz}^2}\right)+\left(\frac{\left(I_{xx}-I_{yy}+I_{zz}\right)I_{xz}}{I_{xx}I_z-I_{xz}^2}\right)pq+\left(\frac{\left(I_{yy}-I_{zz}\right)I_{zz}-I_{xz}^2}{I_{xx}I_{zz}-I_{xz}^2}\right)qr,\\
\dot{q}&=\left(\frac{M_y}{I_{yy}}\right)+\left(\frac{I_{xz}}{I_{yy}}\right)\left(r^2-p^2\right)+\left(\frac{I_{zz}-I_{xx}}{I_{yy}}\right)pr,\\
\dot{r}&=\left(\frac{I_{xz}M_x}{I_{xx}I_{zz}-I_{xz}^2}\right)+\left(\frac{I_{xx}M_z}{I_{xx}I_{zz}}\right)+\left(\frac{\left(I_{xx}-I_{yy}\right)I_{xx}-I_{xz}^2}{I_{xx}I_{zz}-I_{xz}^2}\right)pq+\left(\frac{\left(I_{yy}-I_{xx}-I_{zz}\right)I_{xz}}{I_{xx}I_{zz}-I_{xz}^2}\right)qr,
\end{aligned}
\tag{1}
$$

where α, β, and μ represent the angle of attack, sideslip, and bank angle, respectively; p, q, and r are the roll, pitch, and yaw rate, respectively. And M_x, M_y, and M_z denote the roll, pitch, and yaw control torques, respectively; $I_{ij}\,(i = x, y, z; j = x, y, z)$ is rotational inertia. ϕ, θ, χ, and ϑ are longitude, latitude, heading angle, and flight path angle, respectively; Ω_{E} is the Earth rotation angular velocity.

In actual control, vehicles can be regarded as an ideal rigid body. Considering that the rotation rate of the Earth is far less than that of vehicles, the rotation of the Earth is ignored. Besides, orbital motion is much slower than attitude motion, so the orbital motion terms of vehicles are described as $\dot{\phi} = \dot{\theta} = \dot{\vartheta} = \dot{\chi} = 0$. Finally, simplified dynamics can be obtained:

$$
\begin{aligned}
\dot{\alpha}&=-p\cos\alpha\tan\beta+q-r\sin\alpha\tan\beta,\\
\dot{\beta}&=p\sin\alpha-r\cos\alpha,\\
\dot{\mu}&=-p\cos\alpha\cos\beta-q\sin\beta-r\sin\alpha\cos\beta.
\end{aligned}
\tag{2}
$$

Above attitude kinematics equation (2) is abbreviated as

$$
\dot{\gamma}=\Gamma(\cdot)\omega, \tag{3}
$$

where $\gamma = [\alpha, \beta, \mu]^{\mathrm{T}} \in R^3$ and $\omega = [p, q, r]^{\mathrm{T}} \in R^3$. $\Gamma \in R^{3\times3}$ are defined as

$$
\Gamma=\begin{bmatrix}-\cos\alpha\tan\beta & 1 & -\sin\alpha\tan\beta\\ \sin\alpha & 0 & -\cos\alpha\\ -\cos\alpha\cos\beta & -\sin\beta & -\sin\alpha\cos\beta\end{bmatrix}. \tag{4}
$$

Similarly, rotational dynamics can be simplified as

$$
\dot{\omega}=-I^{-1}\Omega I\omega+I^{-1}M_c, \tag{5}
$$

where $I \in R^{3\times3}$ denotes inertial matrix; $M_c = [M_x, M_y, M_z]^{\mathrm{T}} \in R^3$ is a vector of control torques. $\Omega \in R^{3\times3}$ and $I \in R^{3\times3}$ are defined as

$$
\begin{aligned}
I&=\begin{bmatrix}I_{xx} & -I_{xy} & -I_{xz}\\ -I_{xy} & I_{yy} & -I_{yz}\\ -I_{xz} & -I_{yz} & I_{zz}\end{bmatrix},\\
\Omega&=\begin{bmatrix}0 & -r & q\\ r & 0 & -p\\ -q & p & 0\end{bmatrix}.
\end{aligned}
\tag{6}
$$

If there exist external disturbances, d_1 and d_2 are introduced into the vehicle system as follows:

$$
\begin{cases}\dot{\gamma}=\Gamma(\cdot)\omega+d_1,\\ \dot{\omega}=-I^{-1}\Omega I\omega+I^{-1}M_c+d_2,\end{cases} \tag{7}
$$

where $d_1 \in R^3$ and $d_2 \in R^3$ represent external disturbances.

Obviously, the attitude tracking control problem of the reentry vehicles can be described as

$$
\begin{aligned}
\lim_{t\longrightarrow\infty}\left\|\alpha-\alpha_d\right\|&=0,\\
\lim_{t\longrightarrow\infty}\left\|\beta-\beta_d\right\|&=0,\\
\lim_{t\longrightarrow\infty}\left\|\mu-\mu_d\right\|&=0.
\end{aligned}
\tag{8}
$$

3. Controller Design

In the previous section, the nominal attitude model of the reentry vehicle has been established by equations (3) and (5), which can be reorganized as equations (9a) and (9b). This section will devise a controller with an auxiliary according to this vehicle model:

$$
\dot{\gamma}=\Gamma(\cdot)\omega, \tag{9a}
$$

$$
\dot{\omega}=-I^{-1}\Omega I\omega+I^{-1}M_c. \tag{9b}
$$

It is well known that the attitude angles change more slowly than the angular rate. Therefore, according to the principle of time scale separation and cascade control,

equations (9a) and (9b) can be divided into attitude angle slow loop equation (9a) and angle rate fast loop equation (9b), also known as outer loop and inner loop, respectively. In this section, the ADP-based controller will be presented, and the overall control strategy is shown in Figure 1.

As shown in Figure 1, there are two control loops. The outer loop is an attitude control loop with two controllers. The controller 1 generates the main angular rate instruction ω_s according to the guidance instruction γ_d, and the ADP controller outputs the control instruction u_{ADP} according to the attitude angle error; both of which together yield the angular rate ω_c. Then, ω_c is a reference instruction for the inner angular rate loop so that the controller 2 of the inner loop generates the control torque M_c, which acts on the vehicle to output the actual attitude angles and complete the control task.

In this paper, the inner controller 1 and outer loop controller 2 are implemented based on conventional sliding mode control and serve as the main controllers. To increase the performance of the main controller of the outer loop, the ADP controller acts as an auxiliary and adopts an action-dependent structure such as ADHDP. Note that ADHDP belongs to the category of ADP, so it is called ADP in this paper. The output of the ADP serves as a supplementary reference signal for the inner loop. The focus of this paper is to discuss the auxiliary role of ADP structure. Of course, the main controllers can also choose other methods to design, but how to select the main controllers is not the focus of this paper. It should be pointed out that only the ADP auxiliary controller is introduced into the outer loop, mainly because the outer loop variable is the attitude angle and the inner loop variable is the angular rate, and the attitude angle changes slowly than the angular rate. Therefore, in each iteration, the iterative speed of the ADP is more easily matched with the update speed of the main controller 1. Perhaps we can similarly introduce the ADP auxiliary controller with the same structure into the inner loop, and its rationality and effectiveness will be researched and verified in future work.

In the following subsections: according to cascade control strategy, the outer loop controllers are first designed, including the main controller 1 and the ADP-based auxiliary controller. After the reference command signal ω_c is obtained by the outer loop controllers, the inner loop controller 2 is presented.

3.1. Outer Loop Controllers

3.1.1. Main Controller 1. The control objective of the outer loop is to operate the actual attitude angle γ to track γ_d within the desired accuracy. First, take the tracking error $e_\gamma = \gamma_d - \gamma \in R^3$. The sliding switching surface $S_\gamma \in R^3$ of the outer loop can be selected as

$$S_\gamma = \left[S_{\gamma 1}, S_{\gamma 2}, S_{\gamma 3}\right]^T = e_\gamma + \rho_\gamma \int_0^t e_\gamma \mathrm{d}\tau, \tag{10}$$

where $\rho_\gamma = \mathrm{diag}\left\{\rho_{\gamma 1}, \rho_{\gamma 2}, \rho_{\gamma 3}\right\} \in R^{3\times 3}$ and $\rho_{\gamma i} > 0$, $i = 1, 2, 3$ are the parameters to be designed [30]. Obviously, on the sliding surface $S_\gamma = 0$, the tracking error e_γ can be guaranteed to converge uniformly, that is,

$$S_\gamma = e_\gamma + \rho_\gamma \int_0^t e_\gamma \mathrm{d}\tau = 0. \tag{11}$$

In order to ensure the asymptotic convergence of the outer loop tracking error to the sliding surface, the virtual control law must be designed. First, take the derivative of S_γ as

$$\begin{aligned} \dot{S}_\gamma &= \dot{e}_\gamma + \rho_\gamma e_\gamma \\ &= \dot{\gamma}_d - \dot{\gamma} + \rho_\gamma e_\gamma \\ &= \dot{\gamma}_d - \Gamma\omega + \rho_\gamma e_\gamma. \end{aligned} \tag{12}$$

Take the following Lyapunov function:

$$L_1 = \left(\frac{1}{2}\right) S_\gamma^{\mathrm{T}} S_\gamma > 0, \tag{13}$$

and the derivative of L_1 is as

$$\dot{L}_1 = S_\gamma^{\mathrm{T}} \dot{S}_\gamma. \tag{14}$$

By Lyapunov stability, $\dot{L}_1 < 0$ has to be guaranteed. Therefore, the sliding mode approach law can be chosen as

$$\dot{S}_\gamma = -\tau_\gamma \mathrm{sign}\left(S_\gamma\right), \tag{15}$$

where designed parameter $\tau_\gamma > 0$ and $\mathrm{sign}(S_\gamma) = [\mathrm{sign}(S_{\gamma 1}), \mathrm{sign}(S_{\gamma 2}), \mathrm{sign}(S_{\gamma 3})]^{\mathrm{T}}$ denotes a sign function.

According to equations (12) and (15), there exists

$$\dot{\gamma}_d - \Gamma\omega + \rho_\gamma e_\gamma = -\tau_\gamma \mathrm{sign}\left(S_\gamma\right). \tag{16}$$

So, the virtual control law of the outer loop can be obtained as follows:

$$\omega_s = \Gamma^{-1}\left(\dot{\gamma}_d + \rho_\gamma e_\gamma + \tau_\gamma \mathrm{sign}\left(S_\gamma\right)\right). \tag{17}$$

In order to avoid or reduce the sliding mode chattering caused by the sign function in equation (17), a smooth continuous function can be adopted instead of the sign function. Because the saturation function is one of the most simple and effective ways, the virtual control law is redesigned as follows:

$$\omega_s = \Gamma^{-1}\left(\dot{\gamma}_d + \rho_\gamma e_\gamma + \tau_\gamma \mathrm{sat}\left(\frac{S_\gamma}{\xi_\gamma}\right)\right), \tag{18}$$

where $\mathrm{sat}(S_\gamma/\xi_\gamma) = [\mathrm{sat}(S_{\gamma 1}/\xi_\gamma), \mathrm{sat}(S_{\gamma 2}/\xi_\gamma), \mathrm{sat}(S_{\gamma 3}/\xi_\gamma)]^{\mathrm{T}}$ denotes a saturation function with width $\xi_\gamma > 0$ as follows:

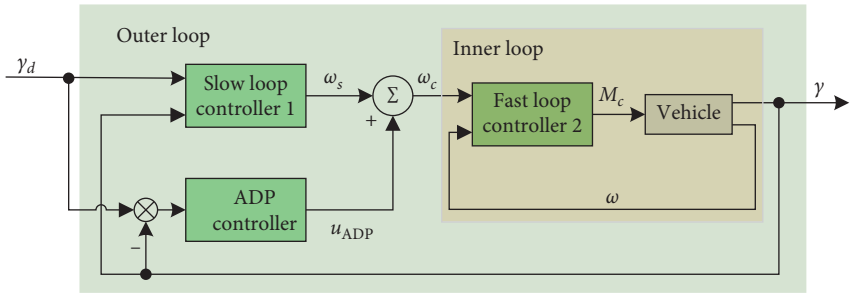

Figure 1: Dual-loop control structure based on ADP.

$$\operatorname*{sat}_{(i=1,2,3)}\left(\frac{S_{\gamma i}}{\xi_\gamma}\right)=\begin{cases}1, & \left(S_{\gamma i}>\xi_\gamma\right),\\ \left(\dfrac{S_{\gamma i}}{\xi_\gamma}\right), & \left(\left|S_{\gamma i}\right|\le\xi_\gamma\right),\\ -1, & \left(S_{\gamma i}<-\xi_\gamma\right).\end{cases}\tag{19}$$

Therefore, according to control law equation (18), the attitude angles can track the commands, and the error e_γ uniformly converges. Next, ω_s will be provided as the main reference signal to the inner loop.

3.1.2. ADP Auxiliary Controller. The idea of ADP is to take advantage of the function estimators to approximate the performance index functions and control strategies that meet the principle of optimality. By designing a critic-action structure, the critic network approximates the performance index J (the cost function) and J is defined as the forward accumulation of the utility function U with the discount factor λ [20, 21]:

$$J(k)=\sum_{i=k}^{\infty}\lambda^{i-k}U(i),\tag{20}$$

where U is usually defined as a quadratic. It can be seen that the cost function is also a quadratic convex function, with only a local minimum and at the same time a global minimum. The action network obtains the optimal control law u^* by minimizing J:

$$u^*(k)=\arg\min_{u(k)}\{U(k)+\lambda J(k+1)\}.\tag{21}$$

In this paper, only the auxiliary ADP controller is added to the outer loop to compensate for the attitude angle error generated by the main controller 1. ADP outputs u_{ADP} (u_{ADP} has the same dimension as ω_s), and the sum of u_{ADP} and ω_s inputs as a reference instruction to the inner loop. Obviously, the ADP controller is sensitive to the attitude angle error. It can be imagined that ADP will start to work when a certain error occurs; when the error meets the threshold requirements, the ADP does not need to work, which will balance the loss in accuracy and calculation speed. However, this does not seem to be the focus of this paper. It may be discussed in future research, such as the selection and optimization of the threshold.

In Figure 2, ADP adopts a network structure based on ADHDP, which includes an action network, a critic network, and attitude model (9a). The input of ADP is the attitude error, and the action network generates the control signal u_{ADP}. At the same time, the critic network approximates J. The specific design of each network is given below.

(1) Critic Network. In Figure 3, the critic network uses a single hidden-layer BP neural network with six input nodes, M hidden nodes, and one output node. The input contains the attitude angle error $\Delta\gamma$ and u_{ADP} generated by the action network. The output is the estimated $\widehat{J}$ of the cost function J. $\mathrm{Wc1}\in R^{M\times 6}$ is the weight matrix of the input layer to the hidden layer and $\mathrm{Wc1}_{ji}\,(i=1,\ldots,6;\ j=1,\ldots,M)$ represents the weight of the i-th input node to the j-th hidden node. $\mathrm{Wc2}\in R^{1\times M}$ is the weight matrix from the hidden-to-output layer, and $\mathrm{Wc2}_j$, $j=1,\ldots,M$ represents the connection weight of the j-th hidden node to the output. $\mathrm{Ch1}\in R^{M\times 1}$ and $\mathrm{Ch2}\in R^{M\times 1}$ are the input and output vectors of hidden nodes, respectively. The active functions of the hidden layer and the output layer are a bipolar sigmoid function and linear function, respectively. The attitude error is as follows:

$$\Delta\gamma=[\Delta\alpha,\Delta\beta,\Delta\mu]^{\mathrm{T}}=\left[\alpha_d-\alpha,\beta_d-\beta,\mu_d-\mu\right]^{\mathrm{T}}.\tag{22}$$

The input of the critic network is $\mathrm{INc}\in R^{6\times 1}$ as

$$\begin{aligned}\mathrm{INc}&=\left[\Delta\gamma;\Delta u_{\mathrm{ADP}}\right]\\&=\left[\Delta\alpha,\Delta\beta,\Delta\mu,u_{\mathrm{ADP}}(1),u_{\mathrm{ADP}}(2),u_{\mathrm{ADP}}(3)\right]^{\mathrm{T}}.\end{aligned}\tag{23}$$

The training of the critic network consists of two parts, one is the forward calculation, and the other is the error backpropagation of updating network weights. The forward process of step k is

$$\begin{aligned}\mathrm{Ch1}_j(k)&=\sum_{i=1}^{6}\mathrm{INc}_i(k)\cdot\mathrm{Wc1}_{ji}(k),\quad j=1,2,\ldots,M,\\ \mathrm{Ch2}_j(k)&=\frac{1-e^{-\mathrm{Ch1}_j(k)}}{1+e^{-\mathrm{Ch1}_j(k)}},\quad j=1,2,\ldots,M,\\ \widehat{J}(k)&=\sum_{j=1}^{M}\mathrm{Ch2}_j(k)\cdot\mathrm{Wc2}_j(k),\quad j=1,2,\ldots,M.\end{aligned}\tag{24}$$

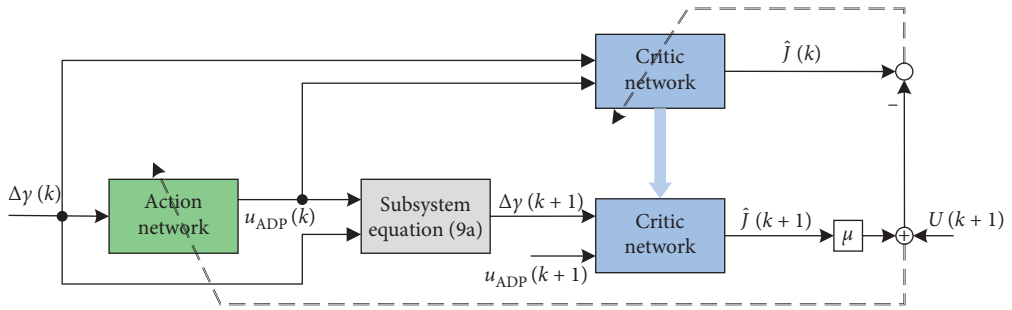

Figure 2: Structure of the ADP controller.

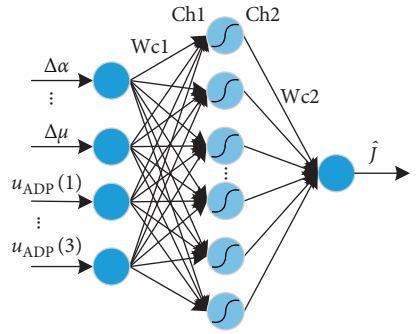

Figure 3: Structure of the critic network.

Equation (24) can be rewritten in matrix form as

$$\begin{aligned} \mathrm{Ch1}(k) &= \mathrm{Wc1} \cdot \mathrm{INc}, \\ \mathrm{Ch2}(k) &= \frac{(1-e)^{-\mathrm{Ch1}(k)}}{(1+e)^{-\mathrm{Ch1}(k)}}, \\ \hat{J}(k) &= \mathrm{Wc2}(k) \cdot \mathrm{Ch2}(k). \end{aligned} \tag{25}$$

Based on the Bellman optimality principle, the critic network approximates the cost function of the system. The actual $J(k)$ is defined as the cumulative return from the current state to the future:

$$J(k) = \sum_{i=k}^{\infty} \lambda^{i-k} U(i), \tag{26}$$

where $\lambda \in (0,1)$ is a discount factor or forgetting factor, indicating the influence of the future state on the current strategy. U is the utility function at each step, which is defined as a quadratic:

$$\begin{cases} U(k) = \Delta\gamma^T(k) \Lambda \Delta\gamma(k), \\ \Lambda = \mathrm{diag}\{\sigma, \sigma, \sigma\}, \quad \sigma > 0. \end{cases} \tag{27}$$

The following error E_c can be defined, and the critic network can approximate J by minimizing E_c:

$$\begin{cases} E_c = \left(\frac{1}{2}\right) e_c^2, \\ e_c = \hat{J}(k) - U(k+1) - \lambda \hat{J}(k+1). \end{cases} \tag{28}$$

Therefore, network weights can be updated through backpropagation of E_c.

(2) Updating the Weights Wc2. Using the gradient descent method, let $\Delta\mathrm{Wc2}$ be the gradient, so

$$\mathrm{Wc2}(k+1) = \mathrm{Wc2}(k) + \Delta\mathrm{Wc2}(k), \tag{29}$$

where each component of $\Delta\mathrm{Wc2}$ is represented as

$$\begin{aligned} \Delta\mathrm{Wc2}_j(k) &= \zeta_c(k) \cdot \left(-\frac{\partial E_c(k)}{\partial \mathrm{Wc2}_j(k)} \right), \quad j = 1, \ldots, M \\ &= -\zeta_c(k) \frac{\partial E_c(k)}{\partial \hat{J}(k)} \frac{\partial \hat{J}(k)}{\partial \mathrm{Wc2}_j(k)}, \quad j = 1, \ldots, M \\ &= -\zeta_c(k) \cdot e_c(k) \cdot \mathrm{Ch2}_j(k), \quad j = 1, \ldots, M, \end{aligned} \tag{30}$$

where $\zeta_c(k) \in (0,1)$ is the learning rate. Equation (30) is combined and rewritten into a matrix form as

$$\Delta\mathrm{Wc2}(k) = -\zeta_c(k) \cdot e_c(k) \cdot \mathrm{Ch2}^{\mathrm{T}}(k). \tag{31}$$

(3) Updating the Weights Wc1. Similarly, let $\Delta\mathrm{Wc1}$ be the gradient, so

$$\begin{aligned}
\mathrm{Wc1}(k+1) &= \mathrm{Wc1}(k) + \Delta\mathrm{Wc1}(k),\\
\Delta\mathrm{Wc1}_{ji}(k) &= \zeta_c(k)\cdot\left(-\frac{\partial E_c(k)}{\partial \mathrm{Wc1}_{ji}(k)}\right), \quad \begin{matrix} i=1,\ldots,6 \\ j=1,\ldots,M \end{matrix}\\
&= -\zeta_c(k)\frac{\partial E_c(k)}{\partial \widehat{J}(k)}\frac{\partial \widehat{J}(k)}{\partial \mathrm{Ch2}_j(k)}\frac{\partial \mathrm{Ch2}_j(k)}{\partial \mathrm{Ch1}_j(k)}\frac{\partial \mathrm{Ch1}_j(k)}{\partial \mathrm{Wc1}_{ji}(k)}\\
&= -\zeta_c(k)\cdot e_c(k)\cdot \mathrm{Wc2}_j(k)\cdot\left(\frac{1}{2}\right)\left[1-\mathrm{Ch2}_j^2(k)\right]\cdot \mathrm{INc}_i(k).
\end{aligned} \tag{32}$$

Combine the above formula into a simplified matrix form as follows:

$$\begin{aligned}
\Delta\mathrm{Wc1}(k) = &-\left(\frac{1}{2}\right)\cdot\zeta_c(k)\cdot e_c(k)\\
&\cdot\left\{\mathrm{Wc2}^{\mathrm{T}}(k)\times[1-\mathrm{Ch2}(k)\times\mathrm{Ch2}(k)]\right\}\cdot\mathrm{INc}^{\mathrm{T}}(k),
\end{aligned} \tag{33}$$

where the symbol "×" represents the Hadamard product of two matrices, that is, bitwise multiplication; "·" represents the ordinary multiplication of matrices. These symbols appearing in the later parts of this paper possess the same meaning.

(4) Action Network. As shown in Figure 4, the action network adopts a single hidden-layer BP neural network with three input nodes, N hidden nodes, and three output nodes. The network's input is $\mathrm{INa} = \Delta\gamma \in R^{3\times 1}$, and output is $u_{\mathrm{ADP}} \in R^{3\times 1}$. Other parameters are defined similarly to the critic network. The active functions of the hidden and output layer are a bipolar sigmoid function and linear function, respectively.

The training of the action network also includes forward calculation and error backpropagation. Firstly, the forward process is briefly presented as

$$\begin{aligned}
\mathrm{Ah1}(k) &= \mathrm{Wa1}\cdot\mathrm{INa},\\
\mathrm{Ah2}(k) &= \frac{(1-e)^{-\mathrm{Ah1}(k)}}{(1+e)^{-\mathrm{Ah1}(k)}},\\
u_{\mathrm{ADP}}(k) &= \mathrm{Wa2}(k)\cdot\mathrm{Ah2}(k).
\end{aligned} \tag{34}$$

The action network generates an optimal control strategy by minimizing the system cost function J. This goal can be achieved by minimizing the defined error E_a:

$$\begin{cases}
E_a(k) = \left(\frac{1}{2}\right)e_a^2(k),\\
e_a(k) = \widehat{J}(k).
\end{cases} \tag{35}$$

(5) Updating the Weights Wa2. With the gradient descent method, the update process of Wa2 is

$$\begin{aligned}
\mathrm{Wa2}(k+1) &= \mathrm{Wa2}(k) + \Delta\mathrm{Wa2}(k),\\
\Delta\mathrm{Wa2}(k) &= \zeta_a(k)\cdot\left(-\frac{\partial E_a(k)}{\partial \mathrm{Wa2}(k)}\right),
\end{aligned} \tag{36}$$

where $\zeta_a(k)$ represents the learning rate. The connection weight from the j-th hidden node to the ith output node is denoted as $\mathrm{Wa2}_{ij}\,(i=1,2,3;\ j=1,\ldots,N)$, so

$$\mathrm{Wa2}_{ij}(k+1) = \mathrm{Wa2}_{ij}(k) + \Delta\mathrm{Wa2}_{ij}(k), \quad \begin{matrix} i=1,2,3 \\ j=1,\ldots,N \end{matrix}, \tag{37}$$

$$\begin{aligned}
\Delta\mathrm{Wa2}_{ij}(k) &= \zeta_a(k)\cdot\left(-\frac{\partial E_a(k)}{\partial e_a(k)}\frac{\partial e_a(k)}{\partial \widehat{J}(k)}\frac{\partial \widehat{J}(k)}{\partial \mathrm{Wa2}_{ij}(k)}\right)\\
&= \zeta_a(k)\cdot\left(-e_a(k)\frac{\partial \widehat{J}(k)}{\partial u_{\mathrm{ADP}i}(k)}\frac{\partial u_{\mathrm{ADP}i}(k)}{\partial \mathrm{Wa2}_{ij}(k)}\right)\\
&= -\zeta_a(k)\cdot e_a(k)\cdot\frac{\partial \widehat{J}(k)}{\partial u_{\mathrm{ADP}i}(k)}\cdot\mathrm{Ah2}_j(k).
\end{aligned} \tag{38}$$

The middle term $(\partial\widehat{J}(k)/\partial u_{\mathrm{ADP}i}(k))$ in equation (38) indicates that the path of the backpropagated signal passes through the critic network when training the action network [31]. Furthermore, by the output and input of the critic network, $(\partial\widehat{J}(k)/\partial u_{\mathrm{ADP}i}(k))$ can be obtained:

$$\frac{\partial \widehat{J}(k)}{\partial u_{\mathrm{ADP}}(k)} = \frac{\partial \widehat{J}(k)}{\partial \mathrm{Ch2}(k)}\frac{\partial \mathrm{Ch2}(k)}{\partial \mathrm{Ch1}(k)}\frac{\partial \mathrm{Ch1}(k)}{\partial u_{\mathrm{ADP}}(k)}. \tag{39}$$

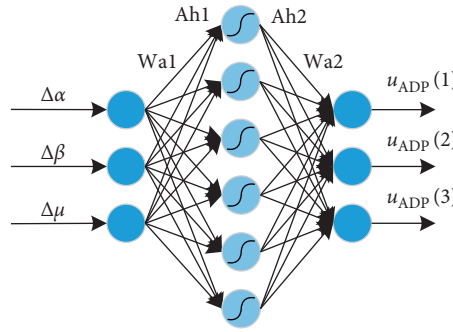

Figure 4: Structure of the action network.

So,

$$\frac{\partial \hat{J}(k)}{\partial u_{\mathrm{ADP}i}(k)} = \sum_{j=1}^{M}\left(\frac{\partial \hat{J}(k)}{\partial \mathrm{Ch2}_j(k)}\frac{\partial \mathrm{Ch2}_j(k)}{\partial \mathrm{Ch1}_j(k)}\frac{\partial \mathrm{Ch1}_j(k)}{\partial u_{\mathrm{ADP}i}(k)}\right), \quad i = 1, 2, 3$$

$$= \sum_{j=1}^{M}\left(\mathrm{Wc2}_j(k)\cdot\left(\frac{1}{2}\right)\left(1-\mathrm{Ch2}_j^2(k)\right)\cdot \mathrm{Wc1}_{(j,i+3)}(k)\right) \tag{40}$$

$$= \left(\frac{1}{2}\right)\cdot \mathrm{Wc1}_{(:,i+3)}{}^{\mathrm{T}}(k)\cdot\left(\mathrm{Wc2}^{\mathrm{T}}(k)\times\left(1-\mathrm{Ch2}(k)\times\mathrm{Ch2}(k)\right)\right),$$

where $Wc1_{(:,i+3)}$ represents the $(i+3)$-th column of Wc1. Equation (40) can be rewritten in matrix form:

$$\frac{\partial \hat{J}(k)}{\partial u_{\mathrm{ADP}}(k)} = \left(\frac{1}{2}\right)\cdot \mathrm{Wc1}_{u_{\mathrm{ADP}}}{}^{\mathrm{T}}(k) \cdot\left(\mathrm{Wc2}^{\mathrm{T}}(k)\times\left(1-\mathrm{Ch2}(k)\times\mathrm{Ch2}(k)\right)\right), \tag{41}$$

where $\mathrm{Wc1}_{u_{\mathrm{ADP}}} = \mathrm{Wc1}(:, 4{:}\ 6)$ represents columns 4 to 6 of Wc1, that is, the connection weights of the three input nodes corresponding to u_{ADP} and all hidden nodes in the critic network. From equations (37)–(41), ΔWa2 can be deduced as

$$\begin{aligned}\Delta \mathrm{Wa2}(k) &= \zeta_a(k)\cdot\left(-\frac{\partial E_a(k)}{\partial e_a(k)}\frac{\partial e_a(k)}{\partial \hat{J}(k)}\frac{\partial \hat{J}(k)}{\partial \mathrm{Wa2}(k)}\right)\\ &= -\zeta_a(k)\cdot e_a(k)\cdot\frac{\partial \hat{J}(k)}{\partial u_{\mathrm{ADP}}(k)}\cdot \mathrm{Ah2}^{\mathrm{T}}(k)\\ &= -\left(\frac{1}{2}\right)\cdot\zeta_a(k)\cdot e_a(k)\cdot\left\{\mathrm{Wc1}_{u_{\mathrm{ADP}}}{}^{\mathrm{T}}(k)\cdot\left(\mathrm{Wc2}^{\mathrm{T}}(k)\times\left(1-\mathrm{Ch2}(k)\times\mathrm{Ch2}(k)\right)\right)\right\}\cdot \mathrm{Ah2}^{\mathrm{T}}(k).\end{aligned} \tag{42}$$

(6) Updating the Weights Wa1. Similar to the Wa2, the update of Wa1 is

$$\mathrm{Wa1}(k+1) = \mathrm{Wa1}(k) + \Delta\mathrm{Wa1}(k), \tag{43}$$

$$\begin{aligned}\Delta\mathrm{Wa1}(k) &= \zeta_a(k)\cdot\left(-\frac{\partial E_a(k)}{\partial \mathrm{Wa1}(k)}\right)\\ &= \zeta_a(k)\cdot\left(-\frac{\partial E_a(k)}{\partial e_a(k)}\frac{\partial e_a(k)}{\partial \widehat{J}(k)}\frac{\partial \widehat{J}(k)}{\partial u_{\mathrm{ADP}}(k)}\frac{\partial u_{\mathrm{ADP}}(k)}{\partial \mathrm{Ah2}(k)}\frac{\partial \mathrm{Ah2}(k)}{\partial \mathrm{Ah1}(k)}\frac{\partial \mathrm{Ah1}(k)}{\partial \mathrm{Wa1}(k)}\right)\\ &= -\zeta_a(k)\cdot e_a(k)\cdot\left\{\left(\mathrm{Wa2}^{\mathrm{T}}(k)\cdot\frac{\partial \widehat{J}(k)}{\partial u_{\mathrm{ADP}}(k)}\right)\times\left(\frac{1}{2}\cdot\left(1-\mathrm{Ah2}(k)\times\mathrm{Ah2}(k)\right)\right)\right\}\cdot\mathrm{INa}^{\mathrm{T}}(k).\end{aligned} \tag{44}$$

Substituting equation (41) into equation (44), ΔWa1 can be easily obtained.

So far, the training process is completed. And the optimal control signal u_{ADP} output by the action network will be combined with ω_s output by outer loop main controller 1, that is

$$\omega_c = \omega_s + u_{\mathrm{ADP}}, \tag{45}$$

where the angular rate signal $\omega_c \in R^{3\times1}$ will be input as the reference command of the inner loop controller 2, and the control torque M_c output by controller 2 will operate the vehicle to complete the attitude control task.

3.2. Inner Loop Controller. To ensure that the actual angular rate ω can stably track the expected reference angular rate ω_c, similar to controller 1, the sliding variable is selected for inner loop controller 2 as follows:

$$S_\omega = \left[S_{\omega1}, S_{\omega2}, S_{\omega3}\right]^T = e_\omega + \rho_\omega\int_0^t e_\omega \mathrm{d}\tau, \tag{46}$$

where $e_\omega = \omega_c - \omega \in R^{3\times1}$ and $\rho_\omega = \mathrm{diag}\{\rho_{\omega1}, \rho_{\omega2}, \rho_{\omega3}\} \in R^{3\times3}$ with $\rho_{\omega i} > 0$, $i = 1, 2, 3$. In order to ensure the inner loop tracking error e_ω asymptotically converges to the sliding surface $S_\omega = 0$, the actual control law M has to be designed.

The derivative of S_ω is

$$\begin{aligned}\dot{S}_\omega &= \dot{e}_\omega + \rho_\omega e_\omega\\ &= \dot{\omega}_c + I^{-1}\Omega I\omega - I^{-1}M_c + \rho_\omega e_\omega.\end{aligned} \tag{47}$$

Take the following Lyapunov function L_2:

$$\begin{aligned}L_2 &= \left(\frac{1}{2}\right)S_\omega^{\mathrm{T}}S_\omega > 0,\\ \dot{L}_2 &= S_\omega^{\mathrm{T}}\dot{S}_\omega.\end{aligned} \tag{48}$$

By Lyapunov stability, $\dot{L}_2 < 0$ has to be guaranteed. Therefore, the dynamics $\dot{S}_\omega$ can be chosen as

$$\dot{S}_\omega = -\tau_\omega \mathrm{sign}\left(S_\omega\right), \tag{49}$$

where designed parameter $\tau_\omega > 0$ and $\mathrm{sign}(S_\omega) = [\mathrm{sign}(S_{\omega1}), \mathrm{sign}(S_{\omega2}), \mathrm{sign}(S_{\omega3})]^{\mathrm{T}}$ denotes a sign function.

According to equations (47) and (49), there exists

$$\dot{\omega}_c + I^{-1}\Omega I\omega - I^{-1}M_c + \rho_\omega e_\omega = -\tau_\omega \mathrm{sign}\left(S_\omega\right). \tag{50}$$

So, the actual control law of the inner loop can be obtained as follows:

$$M_c = I\dot{\omega}_c + \Omega I\omega + I\rho_\omega e_\omega + \tau_\omega I\cdot\mathrm{sign}\left(S_\omega\right). \tag{51}$$

Similarly, a continuous saturation function is chosen to replace the sign function to reduce the chattering. Therefore, the actual control law is rewritten as follows:

$$M_c = I\dot{\omega}_c + \Omega I\omega + I\rho_\omega e_\omega + \tau_\omega I\cdot\mathrm{sat}\left(\frac{S_\omega}{\xi_\omega}\right). \tag{52}$$

where $\mathrm{sat}(S_\omega/\xi_\omega) = [\mathrm{sat}(S_{\omega1}/\xi_\omega), \mathrm{sat}(S_{\omega2}/\xi_\omega), \mathrm{sat}(S_{\omega3}/\xi_\omega)]^{\mathrm{T}}$ denotes a saturation function with width $\xi_\omega > 0$.

Therefore, for actual control law as equation (52), $\dot{L}_2 < 0$ holds. That is, the actual attitude angular rate ω converges asymptotically to the expected angular rate ω_c.

4. Implementation Issues

In Section 3, the design of ADP auxiliary controller is completed, but the parameter selection and training speed of ADP cannot be ignored in practical application. So, in this section, some issues are discussed about implementation of ADP structure, including parameter selection for networks and skills related to speed up training.

4.1. Network Parameters and Their Convergence. It is clear that the critic network with a single hidden layer and randomly initialized weights can approximate J with arbitrarily small errors, that is, $\lim_{k\longrightarrow\infty}\|\widehat{J}(k) - J(k)\| = 0$. Similarly, the action network with randomly initialized weights can minimize the cost function $\widehat{J}$ and its output can approximate to the optimal control law u_{ADP}^*, that is,

$$u_{\mathrm{ADP}}^* = \arg\min_{u_{\mathrm{ADP}}^*}\|\widehat{J}\|. \tag{53}$$

In other words, both the critic network and action network evolve towards the optimal direction to achieve their goals. Furthermore, considering equations (25) and (34), it is because of the adjustment of network weights Wc1, Wc2, Wa1, and Wa2 that the output of the networks reaches the desired optimal value. That is, when the optimal control strategy u_{ADP}^* is obtained, the network weights will also reach the optimal weights as follows [32]:

$$\begin{cases} \mathrm{Wc}^* = \arg\min_{\mathrm{Wc}} \|\widehat{J}(k) - U(k+1) - \lambda\widehat{J}(k+1)\|, \\ \mathrm{Wa}^* = \arg\min_{\mathrm{Wa}} \|\widehat{J}(k)\|, \end{cases} \tag{54}$$

where Wc^* and Wa^* represent the optimal weights of the critic and action network, respectively.

Lemma 1. *In critic and action network, the weights Wc and Wa are finally uniformly stable and approach the optimal weights Wc^* and Wa^*.*

Proof. It is well known that the weights of the input to hidden layer are similar to the weights of the hidden to output layer. In order to facilitate the elaboration, this paper only presents the uniform stability proof about Wc2 and Wa2, which are the weights of the hidden to output layer. Let the optimal weights corresponding to Wc2 and Wa2 be $\mathrm{Wc2}^*$ and $\mathrm{Wa2}^*$, respectively, and they are bounded. $\|\mathrm{Wc2}^*\| \le \kappa_c$, $\|\mathrm{Wa2}^*\| \le \kappa_a$, and κ_c, κ_a are positive constants.

Equation (28) can be rewritten as

$$\begin{aligned} E_c(k) &= \left(\frac{1}{2}\right)e_c^2(k), \\ e_c(k) &= \lambda\widehat{J}(k) - (\widehat{J}(k-1) - U(k)) \\ &= \lambda \cdot \mathrm{Wc2}(k) \cdot \mathrm{Ch2}(k) - (\lambda \cdot \mathrm{Wc2}(k-1) \\ &\quad \cdot \mathrm{Ch2}(k-1) - U(k)). \end{aligned} \tag{55}$$

From equations (29) to (31), the update of Wc2 can be rewritten as follows:

$$\begin{aligned} \mathrm{Wc2}(k+1) &= \mathrm{Wc2}(k) + \Delta\mathrm{Wc2}(k) \\ &= \mathrm{Wc2}(k) + \left(-\zeta_c(k) \cdot \frac{\partial E_c(k)}{\partial e_c(k)} \frac{\partial e_c(k)}{\partial \widehat{J}(k)} \frac{\partial \widehat{J}(k)}{\mathrm{Wc2}(k)}\right) \\ &= \mathrm{Wc2}(k) - \lambda \cdot \zeta_c(k) \cdot (\lambda \cdot \mathrm{Wc2}(k) \\ &\quad \cdot \mathrm{Ch2}(k) - ((\lambda \cdot \mathrm{Wc2}(k-1)) \cdot \mathrm{Ch2}(k-1) \\ &\quad - U(k))) \cdot \mathrm{Ch2}^{\mathrm{T}}(k). \end{aligned} \tag{56}$$

Similarly, the update of Wa2 is

$$\begin{aligned} \mathrm{Wa2}(k+1) &= \mathrm{Wa2}(k) + \Delta\mathrm{Wa2}(k), \\ &= \mathrm{Wa2}(k) + \zeta_a(k) \cdot \left(-\frac{\partial E_a(k)}{\partial \mathrm{Wa2}(k)}\right) \\ &= \mathrm{Wa2}(k) - \left(\frac{1}{2}\right) \cdot \zeta_a(k) \cdot e_a(k) \cdot \left\{\mathrm{Wc1}_{u_{\mathrm{ADP}}}{}^{\mathrm{T}}(k) \cdot \left(\mathrm{Wc2}^{\mathrm{T}}(k) \times (1 - \mathrm{Ch2}(k) \times \mathrm{Ch2}(k))\right)\right\} \cdot \mathrm{Ah2}^{\mathrm{T}}(k) \\ &= \mathrm{Wa2}(k) - \left(\frac{1}{2}\right) \cdot \zeta_a(k) \cdot \widehat{J}(k) \cdot \left(\Theta(k) \cdot \mathrm{Wc2}^{\mathrm{T}}(k) \cdot \mathrm{Ah2}^{\mathrm{T}}(k)\right) \\ &= \mathrm{Wa2}(k) - \left(\frac{1}{2}\right) \cdot \zeta_a(k) \cdot (\mathrm{Wc2}(k) \cdot \mathrm{Ch2}(k)) \cdot \left(\Theta(k) \cdot \mathrm{Wc2}^{\mathrm{T}}(k) \cdot \mathrm{Ah2}^{\mathrm{T}}(k)\right), \end{aligned} \tag{57}$$

where $\Theta(k) = \mathrm{Wc1}_{u_{\mathrm{ADP}}}{}^{\mathrm{T}}(k) \times [1 - \mathrm{Ch2}(k) \times \mathrm{Ch2}(k) \quad 1 - \mathrm{Ch2}(k) \times \mathrm{Ch2}(k) \quad 1 - \mathrm{Ch2}(k) \times \mathrm{Ch2}(k)]^{\mathrm{T}}$.

First, the Lyapunov method is adopted to analyse the convergence of Wc2:

$$V_c(k) = \frac{1}{\zeta_c(k)} \mathrm{tr}\left(\widetilde{\mathrm{W}}\mathrm{c2}(k)\widetilde{\mathrm{W}}\mathrm{c2}^{\mathrm{T}}(k)\right), \tag{58}$$

where $\widetilde{\mathrm{W}}\mathrm{c2}(k) = \mathrm{Wc2}(k) - \mathrm{Wc2}^*(k)$ is the error between actual and optimal weights. Then, the first-order difference of V_c is expressed as

$$\begin{aligned} \Delta V_c(k) = \frac{1}{\zeta_c(k)} \mathrm{tr}\Big(&\widetilde{\mathrm{W}}\mathrm{c2}(k+1)\widetilde{\mathrm{W}}\mathrm{c2}^{\mathrm{T}}(k+1) \\ &- \widetilde{\mathrm{W}}\mathrm{c2}(k)\widetilde{\mathrm{W}}\mathrm{c2}^{\mathrm{T}}(k)\Big). \end{aligned} \tag{59}$$

According to equation (56), (60) can be obtained:

$$\begin{aligned}\widetilde{\mathrm{W}}\mathrm{c2}(k+1) &= \mathrm{Wc2}(k+1) - \mathrm{Wc2}^{*} \\ &= \mathrm{Wc2}(k) - \lambda \cdot \zeta_c(k) \cdot [\lambda \cdot \mathrm{Wc2}(k) \cdot \mathrm{Ch2}(k) - (\lambda \cdot \mathrm{Wc2}(k-1) \cdot \mathrm{Ch2}(k-1) - U(k))] \cdot \mathrm{Ch2}^{\mathrm{T}}(k) - \mathrm{Wc2}^{*} \\ &= \widetilde{\mathrm{W}}\mathrm{c2}(k) - \lambda \cdot \zeta_c(k) \cdot [\lambda \cdot (\widetilde{\mathrm{W}}\mathrm{c2}(k) + \mathrm{Wc2}^{*}) \cdot \mathrm{Ch2}(k) - (\lambda \cdot \mathrm{Wc2}(k-1) \cdot \mathrm{Ch2}(k-1) - U(k))] \cdot \mathrm{Ch2}^{\mathrm{T}}(k) \\ &= \widetilde{\mathrm{W}}\mathrm{c2}(k) \cdot \left(I - \lambda^2 \cdot \zeta_c(k) \cdot \mathrm{Ch2}(k)\mathrm{Ch2}^{\mathrm{T}}(k)\right) - \lambda \cdot \zeta_c(k) \cdot [\lambda \cdot \mathrm{Wc2}^{*} \cdot \mathrm{Ch2}(k) - (\lambda \cdot \mathrm{Wc2}(k-1) \\ &\quad \cdot \mathrm{Ch2}(k-1) - U(k))] \cdot \mathrm{Ch2}^{\mathrm{T}}(k).\end{aligned} \tag{60}$$

In addition, denote the approximation error between actual and optimal output as

$$\delta_c(k) = (\mathrm{Wc2}(k) - \mathrm{Wc2}^{*}) \cdot \mathrm{Ch2}(k) = \widetilde{\mathrm{W}}\mathrm{c2}(k) \cdot \mathrm{Ch2}(k). \tag{61}$$

Substituting equations (60) and (61) into equation (59), $\Delta V_c(k)$ can be deduced:

$$\begin{aligned}\Delta V_c(k) &= -\lambda^2 \|\delta_c(k)\|^2 - \lambda^2 \left(1 - \lambda^2 \cdot \zeta_c(k) \|\mathrm{Ch2}(k)\|^2\right) \\ &\quad \cdot \left\|\delta_c(k) + \mathrm{Wc2}^{*} \cdot \mathrm{Ch2}(k) + \left(\frac{1}{\lambda}\right) U(k) - \left(\frac{1}{\lambda}\right) \mathrm{Wc2}(k-1) \cdot \mathrm{Ch2}(k-1)\right\|^2 \\ &\quad + \|\lambda \cdot \mathrm{Wc2}^{*} \cdot \mathrm{Ch2}(k) + U(k) - \mathrm{Wc2}(k-1) \cdot \mathrm{Ch2}(k-1)\|^2.\end{aligned} \tag{62}$$

Furthermore, applying the Cauchy–Schwarz inequality [33], it can be deduced as

$$\begin{aligned}\Delta V_c(k) \leq & -\lambda^2 \|\delta_c(k)\|^2 \\ & - \lambda^2 \left(1 - \lambda^2 \cdot \zeta_c(k) \|\mathrm{Ch2}(k)\|^2\right) \left\|\delta_c(k) + \mathrm{Wc2}^{*} \cdot \mathrm{Ch2}(k) + \frac{1}{\lambda} U(k) - \frac{1}{\lambda} \mathrm{Wc2}(k-1) \cdot \mathrm{Ch2}(k-1)\right\|^2 \\ & + 2\left\|\lambda \cdot \mathrm{Wc2}^{*} \cdot \mathrm{Ch2}(k) + U(k) - \frac{1}{2} \mathrm{Wc2}(k-1) \cdot \mathrm{Ch2}(k-1) - \frac{1}{2} \mathrm{Wc2}^{*} \cdot \mathrm{Ch2}(k-1)\right\|^2 + \frac{1}{2} \|\delta_c(k-1)\|^2.\end{aligned} \tag{63}$$

Similarly, set $V_a(k) = (1/\psi\zeta_a(k))\mathrm{tr}(\widetilde{\mathrm{W}}\mathrm{a2}(k)\widetilde{\mathrm{W}}\mathrm{a2}^{\mathrm{T}}(k))$, $(\psi > 0)$.

Denote the approximation error of the action network between the actual and optimal output as $\delta_a(k) = (\mathrm{Wa2}(k) - \mathrm{Wa2}^{*}) \cdot \mathrm{Ah2}(k) = \widetilde{\mathrm{W}}\mathrm{a2}(k) \cdot \mathrm{Ah2}(k)$. Referring to ΔV_c, ΔV_a satisfies

$$\begin{aligned}\Delta V_a(k) \leq \left(\frac{1}{\psi}\right) \Big\{ & -\left(\|\Theta(k) \cdot \mathrm{Wc2}^{\mathrm{T}}(k)\|^2 - \zeta_a(k) \cdot \|\Theta(k) \cdot \mathrm{Wc2}^{\mathrm{T}}(k)\|^2 \|\mathrm{Ah2}(k)\|^2\right) \cdot \|\mathrm{Wc2}(k) \cdot \mathrm{Ch2}(k)\|^2 \\ & + \|\mathrm{Wc2}(k) \cdot \mathrm{Ch2}(k)\|^2 \|\delta_a(k)\|^2 + 4\|\mathrm{Wc2}^{*} \cdot \mathrm{Ch2}(k)\|^2 + 4\|\delta_c(k)\|^2\Big\}.\end{aligned} \tag{64}$$

Furthermore, set $V_\delta(k) = (1/2)\|\delta_c(k-1)\|^2$, and then

$$\Delta V_\delta(k) = \left(\frac{1}{2}\right)\left(\|\delta_c(k)\|^2 - \|\delta_c(k-1)\|^2\right). \tag{65}$$

From the above derivation, we can finally take the total Lyapunov function $V(k)$ as

$$\begin{aligned} V(k) &= V_c(k) + V_a(k) + V_\delta(k) \\ &\le -\left(\lambda^2 - \frac{1}{2} - \frac{4}{\psi}\right)\left\|\delta_c(k)\right\|^2 \\ &\quad - \lambda^2\left(1 - \lambda^2 \cdot \zeta_c(k)\|\mathrm{Ch2}(k)\|^2\right) \cdot \left\|\delta_c(k) + \mathrm{Wc2}^* \cdot \mathrm{Ch2}(k) + \frac{1}{\lambda}U(k) - \frac{1}{\lambda}\mathrm{Wc2}(k-1) \cdot \mathrm{Ch2}(k-1)\right\|^2 \\ &\quad - \frac{1}{\psi}\left(\left\|\Theta(k) \cdot \mathrm{Wc2}^{\mathrm{T}}(k)\right\|^2 - \zeta_a(k) \cdot \left\|\Theta(k) \cdot \mathrm{Wc2}^{\mathrm{T}}(k)\right\|^2 \|\mathrm{Ah2}(k)\|^2\right) \cdot \|\mathrm{Wc2}(k) \cdot \mathrm{Ch2}(k)\|^2 \\ &\quad + 2\left\|\lambda \cdot \mathrm{Wc2}^*(k) \cdot \mathrm{Ch2}(k) + U(k) - \frac{1}{2}\mathrm{Wc2}(k-1) \cdot \mathrm{Ch2}(k-1) - \frac{1}{2}\mathrm{Wc2}^* \cdot \mathrm{Ch2}(k-1)\right\|^2 \\ &\quad + \frac{1}{\psi}\|\mathrm{Wc2}(k) \cdot \mathrm{Ch2}(k)\|^2\left\|\delta_a(k)\right\|^2 + \frac{4}{\psi}\left\|\mathrm{Wc2}^*(k) \cdot \mathrm{Ch2}(k)\right\|^2. \end{aligned} \tag{66}$$

Selecting some parameters as equation (67), then equation (68) holds:

$$\begin{aligned} \frac{1}{\sqrt{2}} &< \lambda < 1, \\ \zeta_c(k) &< \frac{1}{\lambda^2\|\mathrm{Ch2}(k)\|^2}, \\ \zeta_a(k) &< \frac{1}{\|\mathrm{Ah2}(k)\|^2}, \\ \psi &> \frac{4}{\lambda^2 - (1/2)}, \end{aligned} \tag{67}$$

$$\begin{aligned} \Delta V(k) \le & -\left(\lambda^2 - \frac{1}{2} - \frac{4}{\psi}\right)\left\|\delta_c(k)\right\|^2 - \lambda^2\left(1 - \lambda^2 \cdot \zeta_c(k)\|\mathrm{Ch2}(k)\|^2\right) \\ & \cdot \left\|\delta_c(k) + \mathrm{Wc2}^* \cdot \mathrm{Ch2}(k) + \frac{1}{\lambda}U(k) - \frac{1}{\lambda}\mathrm{Wc2}(k-1) \cdot \mathrm{Ch2}(k-1)\right\|^2 \\ & - \frac{1}{\psi}\left(\left\|\Theta(k) \cdot \mathrm{Wc2}^{\mathrm{T}}(k)\right\|^2 - \zeta_a(k) \cdot \left\|\Theta(k) \cdot \mathrm{Wc2}^{\mathrm{T}}(k)\right\|^2 \|\mathrm{Ah2}(k)\|^2\right) \cdot \|\mathrm{Wc2}(k) \cdot \mathrm{Ch2}(k)\|^2 + D^2, \end{aligned} \tag{68}$$

where D^2 represents

$$\begin{aligned} D^2 = & 2\left\|\lambda \cdot \mathrm{Wc2}^*(k) \cdot \mathrm{Ch2}(k) + U(k) - \frac{1}{2}\mathrm{Wc2}(k-1) \cdot \mathrm{Ch2}(k-1) - \frac{1}{2}\mathrm{Wc2}^* \cdot \mathrm{Ch2}(k-1)\right\|^2 \\ & + \frac{1}{\psi}\|\mathrm{Wc2}(k) \cdot \mathrm{Ch2}(k)\|^2\left\|\delta_a(k)\right\|^2 + \frac{4}{\psi}\left\|\mathrm{Wc2}^*(k) \cdot \mathrm{Ch2}(k)\right\|^2. \end{aligned} \tag{69}$$

Furthermore, applying the Cauchy–Schwarz inequality, we get

$$\begin{aligned} D^2 &\le 8\Big(\lambda^2\left\|\mathrm{Wc2}^*(k)\cdot \mathrm{Ch2}(k)\right\|^2 + U^2(k) + \frac{1}{4}\|\mathrm{Wc2}(k-1)\cdot \mathrm{Ch2}(k-1)\|^2 + \frac{1}{4}\left\|\mathrm{Wc2}^*\cdot \mathrm{Ch2}(k-1)\right\|\Big), \\ &+ \frac{2}{\psi}\left\|\Theta(k)\cdot \mathrm{Wc2}^T(k)\right\|^2 \cdot \left(\|\mathrm{Wa2}(k)\cdot \mathrm{Ah2}(k)\|^2 + \left\|\mathrm{Wa2}^*\cdot \mathrm{Ah2}(k)\right\|^2\right) + \frac{4}{\psi}\left\|\mathrm{Wc2}^*(k)\cdot \mathrm{Ch2}(k)\right\|^2, \\ &\le \left(8\lambda^2 + 4 + \frac{4}{\psi}\right)\cdot \mathrm{Wc2}_{\max}^2\cdot \mathrm{Ch2}_{\max}^2 + \frac{4}{\psi}\cdot \mathrm{Wc2}_{\max}^2\cdot \Theta_{\max}^2 \mathrm{Wa2}_{\max}^2\cdot \mathrm{Ah2}_{\max}^2 + 8U_{\max}^2 = D_{\max}^2, \end{aligned} \tag{70}$$

where the subscript "max" represents the upper bound of the corresponding parameters' 2-norm, such as $\|\mathrm{Wc2}\| \le \mathrm{Wc2}_{\max}$.

Therefore, for any

$$\left\|\zeta_c(k)\right\| > \left(\frac{D_{\max}}{\sqrt{\lambda - (1/2) - (4/\psi)}}\right), \tag{71}$$

$\Delta V(k) \le 0$ holds. This indicates that the actual weights will converge to the optimal weights. In other words, the weight error δ_c and δ_a are uniformly bounded. This also results in a stable ADP system and an optimal output.

Furthermore, note that the components of Ch2 and Ah2 are limited to [−1, 1] due to the activation functions of the hidden nodes, that are

$$\begin{aligned} -1 \le \mathrm{Ch2}_i \le 1, \quad i = 1, \ldots, M, \\ -1 \le \mathrm{Ah2}_j \le 1, \quad j = 1, \ldots, N. \end{aligned} \tag{72}$$

So, there exist

$$\begin{aligned} \|\mathrm{Ch2}(k)\|^2 &= \sum_{i=1}^{M}\left[\mathrm{Ch2}_i(k)\right]^2 \le M, \\ \|\mathrm{Ah2}(k)\|^2 &= \sum_{j=1}^{N}\left[\mathrm{Ah2}_j(k)\right]^2 \le N. \end{aligned} \tag{73}$$

According to equation (67), some networks' parameters should satisfy

$$\begin{aligned} \frac{1}{\sqrt{2}} &< \lambda < 1, \\ \zeta_c(k) &< \frac{1}{\lambda^2 M}, \\ \zeta_a(k) &< \frac{1}{N}, \\ \psi &> \frac{4}{\lambda^2 - (1/2)}. \end{aligned} \tag{74}$$

Equation (74) provides a simple and intuitive guidance to select networks' structure and learning rate, while maintaining the stability of weights and ADP structure.

4.2. Improvement in Implementation. In the previous literature, when it comes to the training of feedforward networks, all weights usually need to be adjusted, so there are serious dependencies between different layers. Moreover, the algorithm based on gradient descent is widely applied to the learning of various feedforward neural networks. However, it is obvious that the learning method based on gradient descent is usually very slow and time-consuming because of improper learning steps, or it is easy to be overtrained and falls into local minima.

In order to make the training process as time-saving as possible and better meet the time matching between online training and practical applications, we can consider two ideas: one is based on Igelnik and Pao's theory [34], that is, for a single hidden-layer forward neural network, if the weights of input to hidden layer are randomly initialized and kept constant, as long as the number of hidden nodes is sufficient, the approximation error of the network can be arbitrarily small. The second is based on the extreme learning machine (ELM) proposed by Huang et al. [35, 36]. For a single hidden layer forward neural network, the weights of the input to hidden layer are initialized randomly and kept constant, and then the hidden nodes are arbitrarily selected. The weights of hidden to output layer are directly determined analytically by the Moore–Penrose inverse, without necessary to derive and calculate partial derivatives layer by layer such as the gradient descent method. The speed of extreme learning methods has been proven to be tens or even thousands of times that of ordinary gradient descent methods, and it can effectively reduce complexity and avoid local minima [37].

To facilitate implementation, this paper will adopt the first idea to improve the performance; that is, the weights Wc1 and Wa1 are randomly initialized in a finite interval and kept constant, and only the weights Wc2 and Wa2 are adjusted by the gradient descent algorithm, resulting in effectively avoiding excessive time consumption. As for the thinking based on extreme learning machine, it is only given here without in-depth discussion due to the limited space of this paper and the lack of theoretical guidance in the application of vehicles. We may make further analysis and give more rigorous theories to support the application in practical vehicle control in future research.

5. Simulations

In this section, the control strategy with ADP derived above is implemented to vehicle attitude control, and the

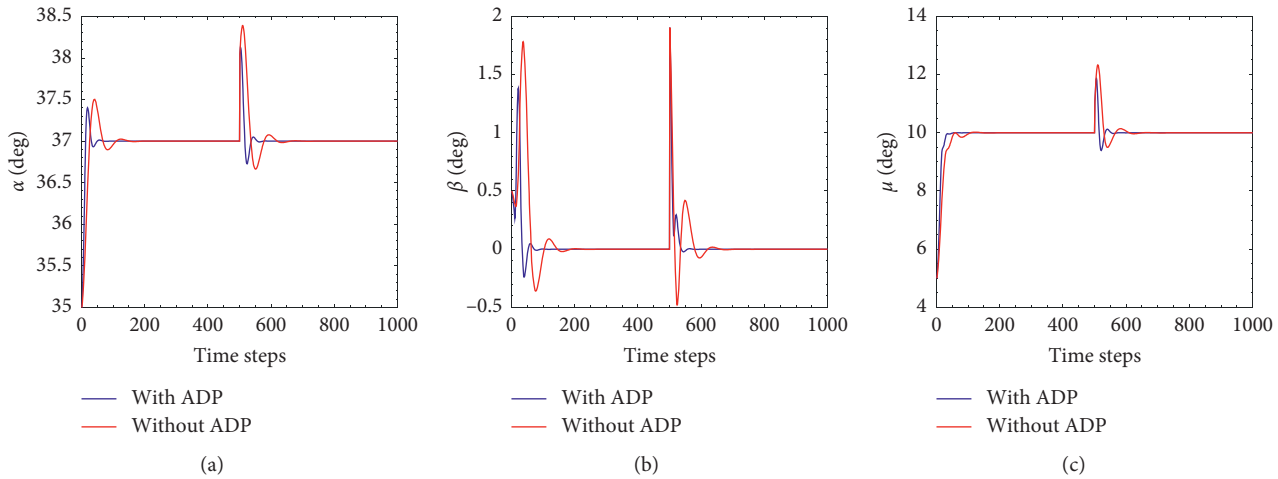

Figure 5: Tracking result of the three attitude angles. (a) The angle of attack α. (b) The sideslip angle β. (c) The bank angle μ.

effectiveness of the designed strategy is verified by comparing with the conventional controller without ADP.

According to a vehicle model in laboratory, the inertia matrix I is taken as

$$I = \begin{bmatrix} 135 & -20 & -1 \\ -20 & 1060 & -1 \\ -1 & -1 & 975 \end{bmatrix}. \quad (75)$$

The common parameters are taken as follows: $\rho_\gamma = \text{diag}\{1, 1, 1\}$, $\rho_w = \text{diag}\{0.5, 0.5, 0.5\}$; the width $\xi_\gamma = \xi_w = 2$ in the saturation function; $\tau_\gamma = \tau_w = 0.2$. The number of hidden nodes is $M = N = 8$. According to equation (74), the discount factor takes $\lambda = 0.9$ with learning rate $\zeta_c(k) < 0.155$, $\zeta_a(k) < 0.125$. Take $\Lambda = \text{diag}\{1, 1, 1\}$, and all weights are randomly initialized in $[-0.2, 0.2]$.

Set the initial flight state of the vehicle as $\gamma_0 = [35, 0.5, 5]^T$ deg and $w_0 = [0, 0, 0]^T$ (rad/s). The desired attitude instruction is $\gamma_d = [37, 0, 10]^T$ deg, and the simulation step size is 0.02 s. To verify the performance of the controller, pulsed disturbances $d_1 = d_2 = [10, 20, 10]$ will be added at 10 s.

Figure 5 presents the tracking results of the three attitude angles. As can be seen from these figures, the controller with ADP is more responsive than the controller without ADP. For example, the controller with ADP can accurately track instructions within 100 steps and cause less overshoot, while the controller without ADP requires about 200 steps. When external disturbances are added at 10 s, the controller with ADP also responds more quickly and with less overshoot. Through these, it can be seen that ADP improves the performance of the system.

The controller with ADP shows faster performance and less overshoot, which benefits from the ADP structure's auxiliary behaviour to the outer loop. Through the training process that meets the expected threshold, the ADP structure generates the auxiliary optimal control signal to compensate for the deficiency of the outer loop main controller 1 in eliminating attitude error. Figures 6–11 show the training process of the ADP network. Specifically,

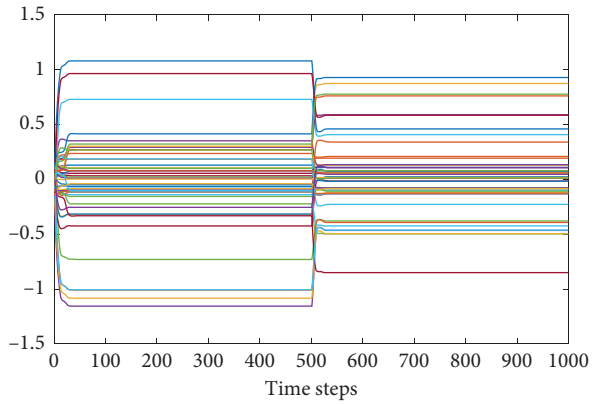

Figure 6: Wc1.

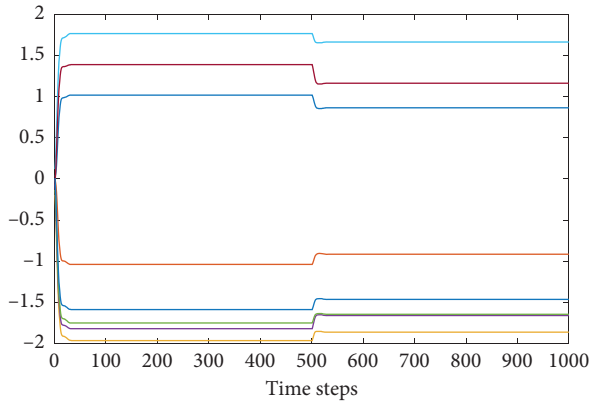

Figure 7: Wc2.

Figures 6–9 show the dynamic adjustment of network weights. Figures 10 and 11 are the estimated value of the cost function output by the critic network and the optimal control signal output by the action network. Compared to the previous Figures 5, 6–9 show the rapid adjustment of the network weights at the beginning stage to achieve the

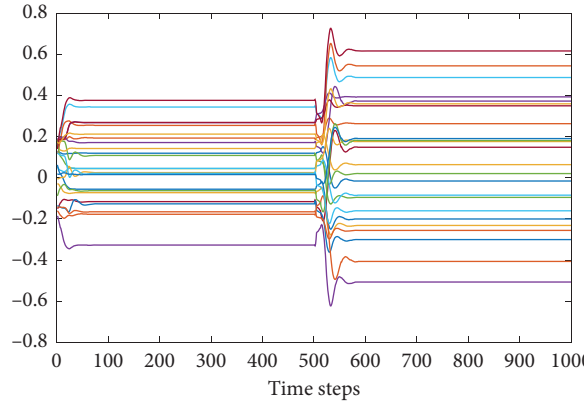

Figure 8: Wa1.

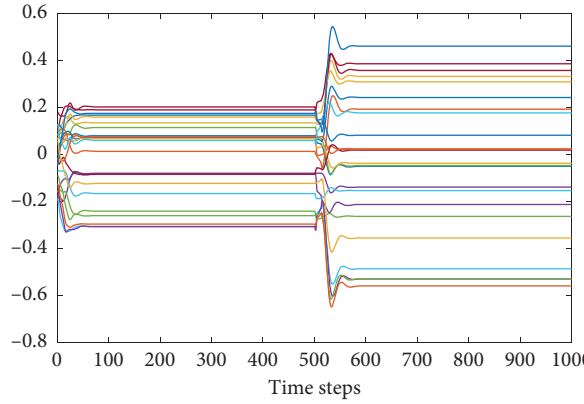

Figure 9: Wa2.

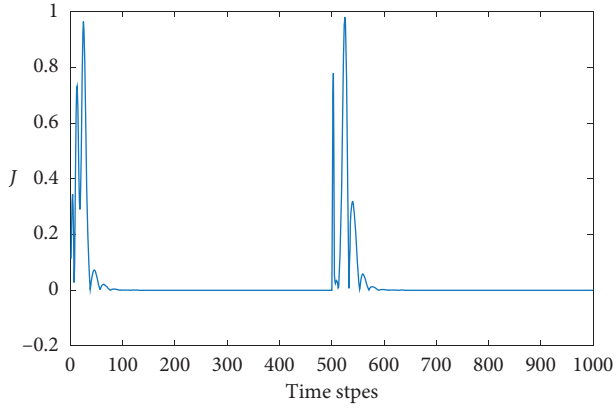

Figure 10: Estimated cost function J.

purpose of tracking instructions. As the system output gradually keeps up with the instructions, the weights also converge to the optimal weights (W^* as demonstrated in Section 4) and remain stable. When the external disturbances are added at 10 s, the network weights are adjusted again and tend to other optimal weights. It shows that ADP produces auxiliary output to play a certain role at the beginning and when disturbance appears.

According to the thinking and analysis in Section 4.2, when implementing this control strategy, it can be considered that randomly initializing the weights of the input to hidden layer (Wc1 and Wa1) and keeping them constant. During the training, only adjusting the weights Wc2 and Wa2 can not only achieve the same optimal control goal but also greatly reduce the time consumption. Figures 12 and 13 show the corresponding weight changes. Simultaneously, Figure 14 shows the comparison of time consumption in 12 group simulations. It can be further concluded that the average time consumption of maintaining the weights of the input-to-hidden layer (Wc1 and Wa1) and only adjusting the weights of the hidden-to-output layer (Wc2 and Wa2) is 31.9% lower than that of adjusting all weights. Although the sample in Figure 14 is limited, combined with the analysis in Section 4.2 and neural network theory, the effectiveness of this idea in reducing time consumption and improving efficiency is significant.

Furthermore, Figures 15–19 show the tracking control results of time-varying attitude commands, using the controller with ADP. The pulsed disturbances $d = [5, 1, 1]^{\mathrm{T}}$ and $d = [10, 2, 2]^{\mathrm{T}}$ are introduced at 10 s and 20 s, respectively, as shown by the yellow arrow in the figures. From Figures 15–17, it can be seen that the controller with ADP auxiliary structure can make the actual attitude angles accurately track the commands. Figures 18–19 show the weights of action network

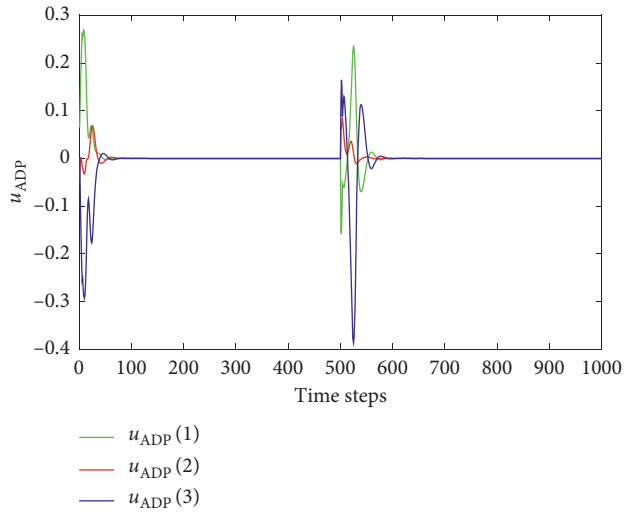

Figure 11: Optimal control signal u_{ADP}.

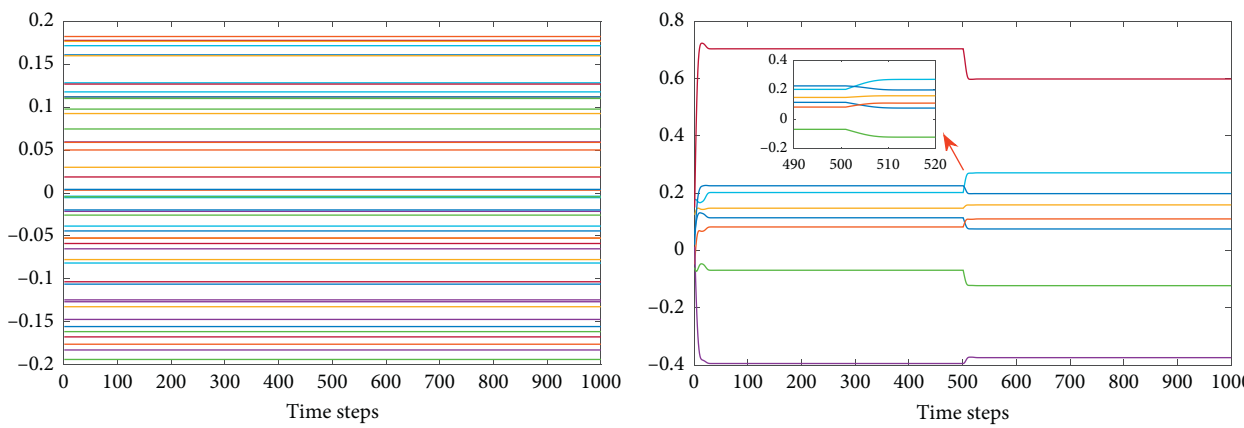

Figure 12: Keep (a) Wc1 and update (b) Wc2.

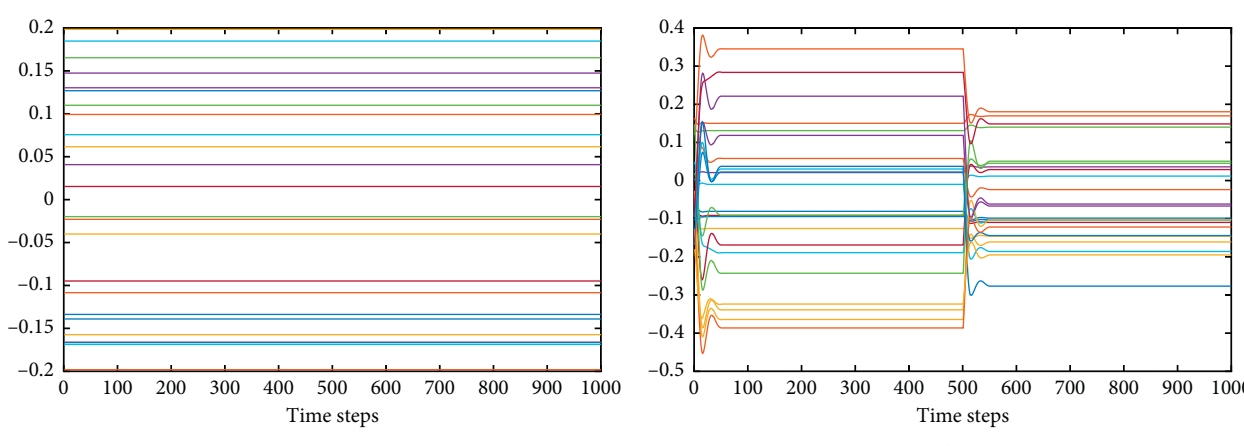

Figure 13: Keep (a) Wa1 and update (b) Wa2.

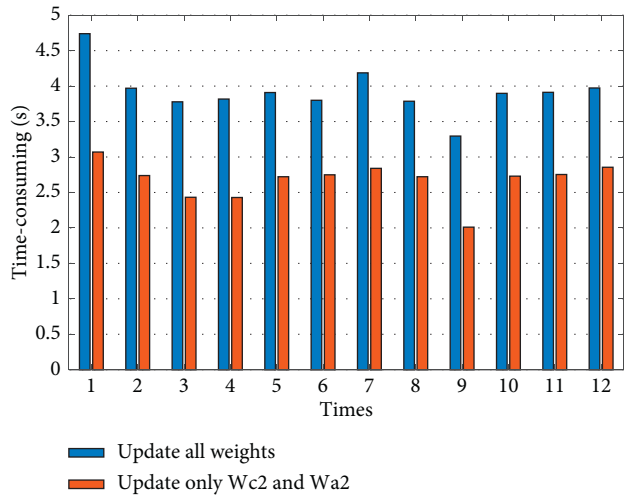

Figure 14: Comparison of time consumption in 12 group simulations.

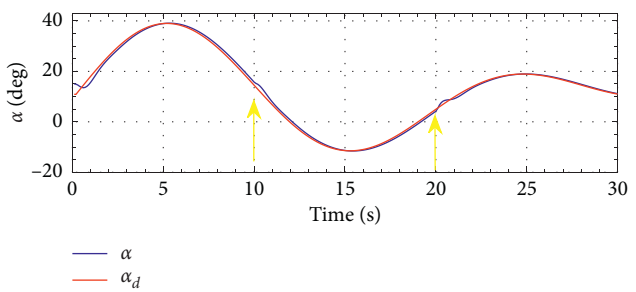

Figure 15: Tracking result of the angle of attack α.

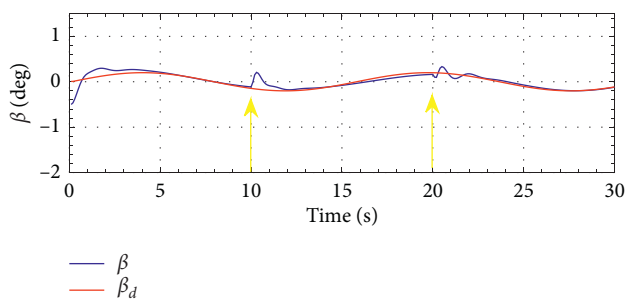

Figure 16: Tracking result of the sideslip angle β.

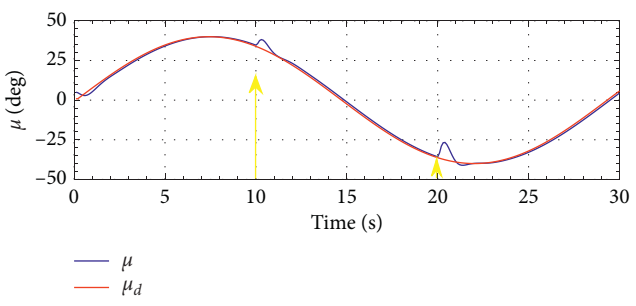

Figure 17: Tracking result of the bank angle μ.

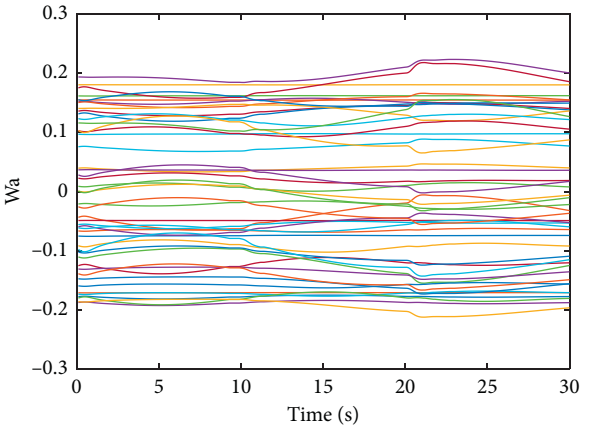

Figure 18: Wa1 of action network.

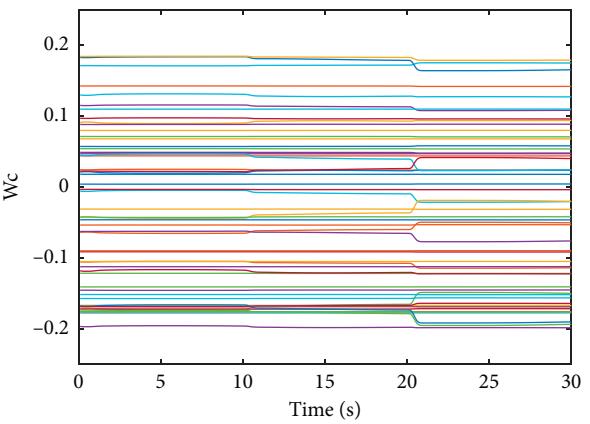

Figure 19: Wc1 of critic network.

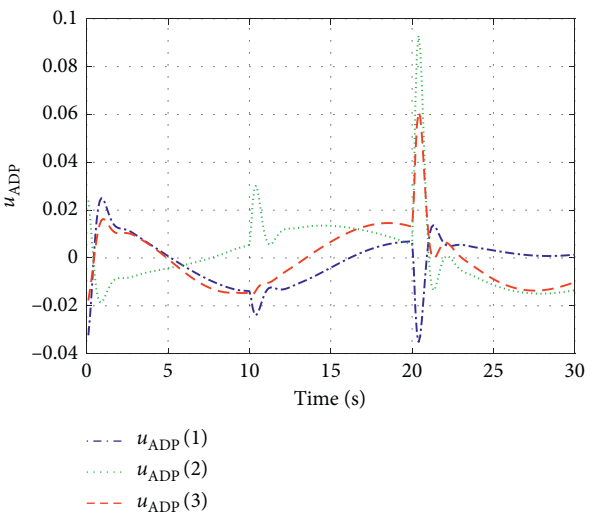

Figure 20: Optimal control signal u_{ADP}.

and critic network in ADP. The weights of the action network are dynamically adjusted to output the optimal auxiliary control signal u_{ADP} in real time, as shown in Figure 20. Figure 21 shows the control torque acting on the vehicle. From

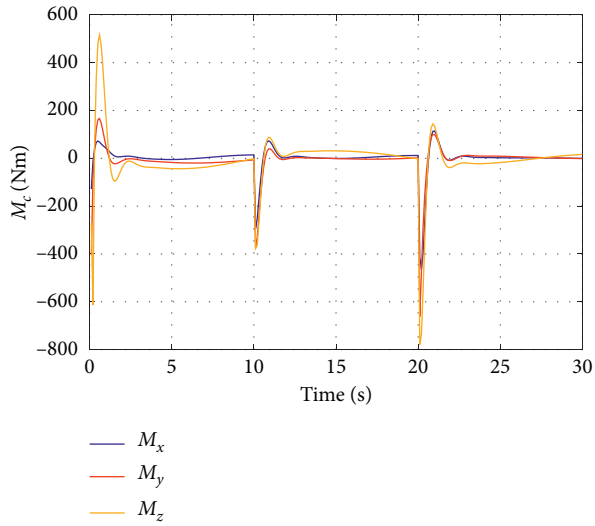

Figure 21: Control inputs M_c.

these, it can be seen that the controller with ADP auxiliary structure has good dynamic stability performance.

6. Conclusions

Combining the hottest reinforcement learning at present, this paper presents an ADP-based attitude control methodology for reentry vehicles, applying the ADP to the three-channel attitude control. First, a nonlinear model of the three-channel attitude system is established, and it is divided into inner and outer loops according to the principle of time scale separation. Both the inner and outer loops utilize a conventional sliding mode controller as the main controller, and an auxiliary ADP framework is introduced to the outer loop. When facing the vehicle's nonlinearity and sudden disturbances in particular, the main controller is easy to be weak due to its lack of sufficient adaptability. At this time, the auxiliary role of ADP will be fully exerted. Because ADP uses the critic network and action network, ADP structure has good learning ability. It generates the optimal auxiliary signal immediately after learning the tracking error to compensate for the deficiency of the main controller and improves the adaptability and response speed of the entire control system. For implementation, this paper discusses selection strategies of the ADP parameter and some tips for speeding up training. And the stability is proved by the Lyapunov method. Finally, simulation results of step and time-varying commands demonstrate the effectiveness of the designed scheme for the nonlinear attitude system.

In the future work, we will focus on some switching or event-triggered strategies for this structure with dual controllers. Imagining that if the ADP auxiliary structure is event-triggered rather than time-triggered, it will greatly reduce consumption of ADP's time and system resources, to improve efficiency.

Data Availability

Some data used in this article are confidential, but other public data can be obtained by contacting li_xu@hust.edu.cn.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported partially by the National Natural Science Foundation of China under grant nos. 61873319, 61903146, and 61803162.

References

[1] Z. Gao and J. Fu, "Robust LPV modeling and control of aircraft flying through wind disturbance," *Chinese Journal of Aeronautics*, vol. 32, no. 7, pp. 1588–1602, 2019.

[2] G. Wu, X. Meng, and F. Wang, "Improved nonlinear dynamic inversion control for a flexible air-breathing hypersonic vehicle," *Aerospace Science and Technology*, vol. 78, pp. 734–743, 2018.

[3] I. Ali, G. Radice, and J. Kim, "Backstepping control design with actuator torque bound for spacecraft attitude maneuver," *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 1, pp. 254–259, 2010.

[4] Q. Hu, J. Cao, and Y. Zhang, "Robust backstepping sliding mode attitude tracking and vibration damping of flexible spacecraft with actuator dynamics," *Journal of Aerospace Engineering*, vol. 22, no. 2, pp. 139–152, 2009.

[5] Z. Zhen, P. Zhu, J. Jiang, and G. Tao, "Research progress of adaptive control for hypersonic vehicle in near space," *Journal of Astronautics*, vol. 39, no. 4, pp. 355–367, 2018.

[6] A. H. J. de Ruiter, "Observer-based adaptive spacecraft attitude control with guaranteed performance bounds," *IEEE Transactions on Automatic Control*, vol. 61, no. 10, pp. 3146–3151, 2016.

[7] Y. Liu, Z. Pu, and J. Yi, "Observer-based robust adaptive T2 fuzzy tracking control for flexible air-breathing hypersonic vehicles," *IET Control Theory & Applications*, vol. 12, no. 8, pp. 1036–1045, 2018.

[8] C. Wang, M. Liang, and Y. Chai, "Adaptive neural network control of a class of fractional order uncertain nonlinear MIMO systems with input constraints," *Complexity*, vol. 2019, Article ID 5643298, 17 pages, 2019.

[9] S. G. Khan, G. Herrmann, F. L. Lewis, T. Pipe, and C. Melhuish, "Reinforcement learning and optimal adaptive control: an overview and implementation examples," *Annual Reviews in Control*, vol. 36, no. 1, pp. 42–59, 2012.

[10] T. Mannucci, E.-J. van Kampen, C. de Visser, and Q. Chu, "Safe exploration algorithms for reinforcement learning controllers," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 4, pp. 1069–1081, 2018.

[11] J. Shin, T. A. Badgwell, K.-H. Liu, and J. H. Lee, "Reinforcement learning—overview of recent progress and implications for process control," *Computers & Chemical Engineering*, vol. 127, pp. 282–294, 2019.

[12] A. Heydari, "Theoretical and numerical analysis of approximate dynamic programming with approximation errors," *Journal of Guidance, Control, and Dynamics*, vol. 39, no. 2, pp. 301–311, 2016.

[13] C. Yang, Y. Xu, L. Zhou, and Y. Sun, "Model-free composite control of flexible manipulators based on adaptive dynamic programming," *Complexity*, vol. 2018, Article ID 9720309, 9 pages, 2018.

[14] Y. Sokolov, R. Kozma, L. D. Werbos, and P. J. Werbos, "Complete stability analysis of a heuristic approximate dynamic programming control design," *Automatica*, vol. 59, pp. 9–18, 2015.

[15] Z. Ni, H. He, X. Zhong, and D. V. Prokhorov, "Model-free dual heuristic dynamic programming," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 8, pp. 1834–1839, 2015.

[16] C. Mu, Y. Zhang, Y. Yu, and C. Sun, "An overview on robust control of aviation and aerospace aircraft based on adaptive dynamic programming," *Aerospace Control and Application*, vol. 45, no. 4, pp. 71–79, 2019.

[17] X. Luo, Y. Chen, J. Si, and F. Liu, "Longitudinal control of hypersonic vehicles based on direct heuristic dynamic programming using ANFIS," in *Proceedings of the 2014 International Joint Conference on Neural Networks (IJCNN)*, pp. 3685–3692, Beijing, China, July 2014.

[18] J. Zhu, X. Ge, and M. Wang, "Approximate dynamic programming for attitude control of three-axis satellite," *Journal of Beijing Information Science and Technology University*, vol. 33, no. 1, pp. 27–32, 2018.

[19] Y.-H. Cheng, B. Jiang, H. Li, and X.-d. Han, "On-orbit reconfiguration using adaptive dynamic programming for multi-mission-constrained spacecraft attitude control system," *International Journal of Control, Automation and Systems*, vol. 17, no. 4, pp. 822–835, 2019.

[20] C. Mu, Z. Ni, C. Sun, and H. He, "Data-driven tracking control with adaptive dynamic programming for a class of continuous-time nonlinear systems," *IEEE Transactions on Cybernetics*, vol. 47, no. 6, pp. 1460–1470, 2017.

[21] C. Mu, Z. Ni, C. Sun, and H. He, "Air-breathing hypersonic vehicle tracking control based on adaptive dynamic programming," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 3, pp. 584–598, 2017.

[22] C. Dong, C. Liu, Q. Wang, and L. Gong, "Switched adaptive active disturbance rejection control of variable structure near space vehicles based on adaptive dynamic programming," *Chinese Journal of Aeronautics*, vol. 32, no. 7, pp. 1684–1694, 2019.

[23] Q. Xie, F. Tan, B. Luo, and X. Guan, "Optimal control for vertical take-off and landing aircraft non-linear system by online kernel-based dual heuristic programming learning," *IET Control Theory & Applications*, vol. 9, no. 6, pp. 981–987, 2015.

[24] Y. Zhou, E.-J. van Kampen, and Q. P. Chu, "Incremental model based online dual heuristic programming for nonlinear adaptive control," *Control Engineering Practice*, vol. 73, pp. 13–25, 2018.

[25] Y. Zhou, E.-J. van Kampen, and Q. Chu, "Incremental approximate dynamic programming for nonlinear adaptive tracking control with partial observability," *Journal of Guidance, Control, and Dynamics*, vol. 41, no. 12, pp. 2554–2567, 2018.

[26] Y. Zhou, E.-J. van Kampen, and Q. Chu, "Nonlinear adaptive flight control using incremental approximate dynamic programming and output feedback," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 2, pp. 493–500, 2017.

[27] B. Sun and E.-J. van Kampen, "Incremental model-based global dual heuristic programming with explicit analytical calculations applied to flight control," *Engineering Applications of Artificial Intelligence*, vol. 89, Article ID 103425, 2020.

[28] B. Sun and E.-J. van Kampen, "Incremental model-based global dual heuristic programming for flight control," *IFAC-PapersOnLine*, vol. 52, no. 29, pp. 7–12, 2019.

[29] J. J. Recasens, Q. P. Chu, and J. A. Mulder, "Robust model predictive control of a feedback linearized system for a lifting-body re-entry vehicle," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit (Guidance, Navigation, and Control and Co-Located Conferences*, pp. 1–33, American Institute of Aeronautics and Astronautics, San Francisco, CA, USA, August 2005.

[30] R. Zhai, R. Qi, and J. Zhang, "Compound fault-tolerant attitude control for hypersonic vehicle with reaction control systems in reentry phase," *ISA Transactions*, vol. 90, pp. 123–137, 2019.

[31] E.-J. van Kampen, Q. P. Chu, and J. A. Mulder, "Continuous adaptive critic flight control aided with approximated plant dynamics," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, August 2006.

[32] J. Si and Y. T. Wang, "Online learning control by association and reinforcement," *IEEE Transactions on Neural Networks*, vol. 12, no. 2, pp. 264–276, 2001.

[33] S. Puntanen, G. P. H. Styan, and J. Isotalo, "The Cauchy–Schwarz inequality," in *Matrix Tricks for Linear Statistical Models*, pp. 415–426, Springer, Berlin, Germany, 2011.

[34] B. Igelnik and Y. H. Pao, "Stochastic choice of basis functions in adaptive function approximation and the functional-link net," *IEEE Transactions on Neural Networks*, vol. 6, no. 6, pp. 1320–1329, 1995.

[35] G. Huang, G.-B. Huang, S. Song, and K. You, "Trends in extreme learning machines: a review," *Neural Networks*, vol. 61, pp. 32–48, 2015.

[36] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks," in *Proceedings of the 2004 IEEE International Joint Conference on Neural Networks (IJCNN)*, vol. 1–4, pp. 985–990, Budapest, Hungary, July 2004.

[37] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1–3, pp. 489–501, 2006.