Simulation Study: Trajectory Tracking Control Of Parafoil System Based On Deep Reinforcement Learning

Xuepu Zhang*

School of Big Data and Computer Science,Guizhou Normal University-550025,Guiyang, China

Corresponding author. E-mail: 1017782304@qq.com

Received: Oct. 03, 2024; Accepted: Dec. 21, 2024

The parafoil system is nonlinear and complex with a large time delay. This makes it challenging for traditional control methods to control the parafoil system effectively. However, the Markov property of reinforcement learning offers a new possibility for controlling the parafoil system. Therefore, this paper employs the deep reinforcement learning (DRL) method to train a neural network controller for controlling the parafoil system, based on a modified deterministic version of the distributional soft actor-critic with three refinements (DSAC-T) algorithm and it is named MC-DSAC-T. The controller of the parafoil system is denoted as a multilayer perceptron (MLP) and the objective function of the policy introduces cumulative discounted rewards of a single episode to improve the stability of the iterative update of the policy, a Monte Carlo (MC) thinking. In addition, wind disturbances are introduced during training to enhance the robustness of the neural network controller. First, a nine-degree-of-freedom (nine-DOF) dynamic model of the parafoil system is developed. Secondly, the network structure of the MC-DSAC-T algorithm and the process of updating the network using sampling data were introduced. Finally, the control effects of the neural network controller trained by the proposed method were compared with those of the proportion integration differentiation (PID) controller in a wind environment. While tracking 100 randomly selected trajectory segments, the results show that the neural network controller is superior to the PID controller in distance control accuracy, which proves that the neural network controller can control the parafoil system to perform the tracking task and verify the effectiveness of the proposed method.

Keywords: Parafoil System; Deep Reinforcement Learning; Trajectory Tracking; Neural Network Controller; DSAC-T

http://dx.doi.org/10.6180/jase.202512_28(12).0011

1. Introduction

The parafoil is a type of flexible-wing unmanned aerial vehicle (UAV) with excellent performance, developed based on parachute technology. The parafoil system (unpowered) primarily consists of a parafoil canopy, control lines, suspension lines, a controller, and a payload. The airborne flight and precise airdrop are achieved by manipulating the left and right control lines of the parafoil system, which results in aerodynamic changes. Because of the good glide performance, maneuverability, and strong carrying capacity of the parafoil, it has broad application prospects in aerospace, military, and civil fields [1, 2]. However, the parafoil system is a complex nonlinear system with strong coupling and significant time lags. This makes controlling it challenging when faced with uncertainties and disturbances such as initial state errors, aerodynamic errors, and wind disturbances. As a result, studies in parafoil system control have focused on developing controllers that can control the system to effectively track a predetermined target trajectory - also known as trajectory tracking control. Since real-world flight control experiments are expensive and difficult to conduct, many studies on the parafoil system trajectory tracking control rely on computer simula-

tion [3]. Studies have investigated dynamic models of the parafoil system with varying degrees of freedom, such as the three and four-DOF [4], six-DOF [5], eight-DOF [6], and nine-DOF [7–9]. The nine-DOF model takes into account the translation and rotation of the parafoil system in three-dimensional space, as well as the rotation of the parafoil relative to the payload. Control methods of the parafoil system may be divided into the following two classes: traditional control methods and intelligent control methods. The former uses manual adjustment control parameters of the controller, while the latter intelligent control method uses some optimization algorithms to find the optimal control parameters.

In terms of traditional control methods, some works have focused on improving the disturbance rejection and control accuracy of the parafoil system controllers. N. Slegers et al. [10] used model predictive control (MPC) to control the parafoil system of a six-DOF. They utilized the current state of the system and control inputs to predict future states and determine the current control action. J. Tao et al. [11]combined ESO and nonlinear state error feedback (NLSEF) control laws to design the horizontal and vertical trajectory tracking active disturbance rejection controllers of the powered parafoil system, respectively. It overcomes the influence of model uncertainty and external disturbance and has stronger robustness and wind resistance than the PID controller. Subsequently, J. Tao et al. [12] used computational fluid dynamics (CFD) technology to analyze the aerodynamic performance of the parafoil system in the wind environment and established an eight-DOF dynamic model of the parafoil system under wind conditions. At the same time, an active disturbance rejection control (ADRC) method is proposed for horizontal trajectory tracking control of the parafoil system. Y. Zheng et al. [13] introduced the reduced-order ESO for sideslip angle estimation, then designed a new trajectory tracking guidance law based on the estimated sideslip angle, and used two LADRC controllers to realize three-dimensional trajectory tracking of the parafoil system. To consider the sideslip angle and at the same time take into account the path tracking control accuracy, W. He et al. [14] propose an improved ADRC method based on surge-guided line-of-sight (SGLOS) guidance law and the multi-strategy marine predator algorithm (MSMPA) optimization, which improves the stability by pre-smoothing sideslip angles as well as abrupt yaw angle fluctuations. H. Sun et al. [15] proposed an improved PID control method combined with a linear ESO (LESO) to improve the responsiveness of the controller to system state changes and used an eight-DOF parafoil system for simulation, to obtain a better control effect than the traditional PID controller. To deal with the problems of nonlinearity, large inertia, and strong disturbances in the airdrop environment, J. Tao et al. [16] established a six-DOF dynamic model of the parafoil system in the wind environment and proposed an autonomous path-following control method for the parafoil system-based on generalized predictive control (GPC). Through online identification, the underlying controlled auto-regressive integrated moving average (CARIMA) model between the motor control input and the actual output is established. The GPC strategy is used to calculate the control quantity of the desired heading angle. Compared with the traditional PID control strategy, this method is better in dynamic performance and wind resistance. L. Zhao et al. [17] proposed a model-independent real-time trajectory control method, model-free adaptive control (MFAC) method, to solve the problem that the parafoil system dynamic model is not accurate enough, which makes the controller obtained by simulation training unable to be used in practice. Based on a six-DOF parafoil system, the stability and robustness of the method are demonstrated both theoretically and experimentally. The controller obtained by the proposed method has a smaller standard deviation and overshoot, as well as a smaller average distance error, than the control input values of the controllers obtained by the PID and ADRC methods. Z. Wei et al. [18] proposed a new guidance and control framework based on the dynamic model of the parafoil system. The advanced-step nonlinear model predictive control (as-NMPC) controller was introduced to track the six-DOF trajectory to compensate for wind and other disturbances. The dynamic model is updated by the moving horizon model correction method to compensate for the inaccuracy of the model.

Manual adjustment of control parameters is often inefficient, and the obtained control parameters are usually not close to the optimal control parameters. Therefore, some researchers use intelligent algorithms to optimize the control parameters to get better control parameters. Y.P. Wang et al. [19] proposed a version of the particle swarm optimization (PSO) algorithm with adaptive adjustment of particle weight and used it to automatically tune PID parameters to improve the efficiency of PID parameter optimization. The results show that the adaptive PSO (APSO) method can solve the problem that PID parameters are difficult to determine, and have good dynamic and static performance. H. Jia et al. [20] automatically employed a single neuron to adjust Linear ADRC (LADRC) parameters. Compared with the LADRC trajectory tracking method, the single neuron modified LADRC trajectory tracking control method has a better tracking effect and stronger disturbance suppres-

sion ability. Additionally, other studies have utilized DRL to tune LADRC parameters, such as the DRL algorithms used, including deep deterministic policy gradient (DDPG) [21], twin delayed deep deterministic policy gradient (TD3) [22], and soft actor-critic (SAC) [23], demonstrating successful application in parafoil system control. An integrated guidance and control algorithm based on a proximal policy optimization (PPO) algorithm [24] in the DRL was recently proposed for the parafoil terminal landing problem. The controller trained by the algorithm can control the parafoil system to achieve a landing with an average landing error within a reasonable range and show smooth control output and real-time control ability.

To further improve the performance of the controller in controlling the parafoil system to track the target trajectory. The modified DSAC-T algorithm [25] is utilized in this study to train a neural network trajectory tracking controller of the parafoil system. Treating the control problem as a Markov decision process and consider training as identifying an optimal policy to maximize the expected value of the cumulative discounted reward. The primary contributions of this paper are summarized below.

1) A deep reinforcement learning approach was used to train a neural network trajectory tracking controller for the parafoil system;

2) The MC thinking was utilized to constrain the policy iterative updates, enhancing the stability of the training process;

3) The simulation results validate the effectiveness of the proposed method under wind disturbances.

2. Methods

2.1. Parafoil System Simulation Model

In this study, the dynamic model of a nine-DOF parafoil system is used. To facilitate the analysis of the motion characteristics of the parafoil system, four coordinate systems are used in the modeling: the inertial earth-constant coordinate system is denoted as follows $O_eX_eY_eZ_e$; the coordinate system at the joint c is parallel to the inertial earth-constant coordinate system, denoted as follows $O_cX_cY_cZ_c$; The parafoil body-fixed coordinate system is denoted as $O_pX_pY_pZ_p$; the payload body-fixed y coordinate system is denoted as $O_bX_bY_bZ_b$. Where O_p and O_b are the centers of gravity of the parafoil and the payload respectively and the subscripts e, p,b,c denote the earth, parafoil, the payload, and joint c, respectively. These coordinate systems are shown in Fig.1.

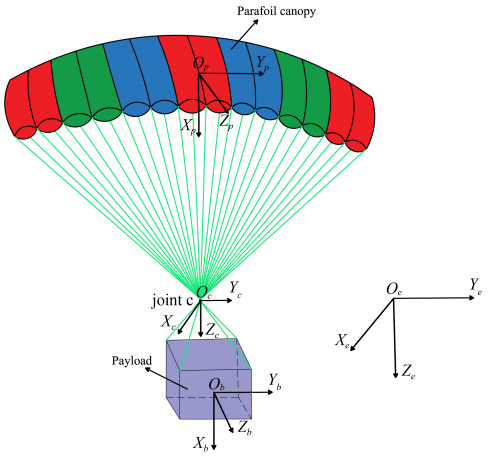


Fig. 1. Schematic diagram of parafoil system coordinates

2.2. Equations of Motion

2.2.1. *Kinematic Equations of Parafoil and Payload*

$$\begin{bmatrix}\dot{x}_e\\ \dot{y}_e\\ \dot{z}_e\end{bmatrix} = \begin{bmatrix}\dot{x}_c\\ \dot{y}_c\\ \dot{z}_c\end{bmatrix} = \begin{bmatrix}u_c\\ v_c\\ w_c\end{bmatrix} \tag{1}$$

$$\begin{bmatrix}\dot{\phi}_p\\ \dot{\theta}_p\\ \dot{\psi}_p\end{bmatrix} = \begin{bmatrix}1 & S_{\phi_p}t_{\theta_p} & C_{\phi_p}t_{\theta_p}\\ 0 & C_{\phi_p} & -S_{\phi_p}\\ 0 & S_{\phi_p}/C_{\theta_p} & C_{\phi_p}/C_{\theta_p}\end{bmatrix}\begin{bmatrix}p_p\\ q_p\\ r_p\end{bmatrix} \tag{2}$$

$$\begin{bmatrix}\dot{\phi}_b\\ \dot{\theta}_b\\ \dot{\psi}_b\end{bmatrix} = \begin{bmatrix}1 & S_{\phi_b}t_{\theta_b} & C_{\phi_b}t_{\theta_b}\\ 0 & C_{\phi_b} & -S_{\phi_b}\\ 0 & S_{\phi_b}/C_{\theta_b} & C_{\phi_b}/C_{\theta_b}\end{bmatrix}\begin{bmatrix}p_b\\ q_b\\ r_b\end{bmatrix} \tag{3}$$

where the trigonometric functions are in shorthand, $\sin\alpha \equiv S_\alpha, \cos\alpha \equiv C_\alpha, \tan\alpha \equiv t_\alpha$. $(u_c, v_c, w_c)^{\mathrm{T}}$ denotes the velocity of the parafoil system. $(\phi, \theta, \psi)^{\mathrm{T}}$ denotes three Euler orientation angles. $(p, q, r)^{\mathrm{T}}$ denotes three Euler angle rates.

2.2.2. *Dynamic Equations of Parafoil and Payload*

$$\begin{bmatrix} M_bT_{e-b} & -M_b\hat{R}_{cb} & 0 & -T_{e-b}\\ (M_p+M_F)\,T_{e-p} & 0 & -(M_p+M_F)\,\hat{R}_{cp} & T_{e-p}\\ 0 & I_b & 0 & \hat{R}_{cb}T_{e-b}\\ 0 & 0 & I_p+I_F & -\hat{R}_{cp}T_{e-p}\end{bmatrix} \times \begin{bmatrix}\dot{V}_c\\ \dot{\Omega}_b\\ \dot{\Omega}_p\\ F_c\end{bmatrix} = \begin{bmatrix}B1\\ B2\\ B3\\ B4\end{bmatrix} \tag{4}$$

$$\begin{aligned} B1 &= F_b^A + F_b^G - M_b\hat{\Omega}_b\hat{\Omega}_bR_{cb}\\ B2 &= F_p^A + F_p^G - (M_p+M_F)\hat{\Omega}_p\hat{\Omega}_pR_{cp}\\ B3 &= -\hat{\Omega}_bI_b\Omega_b\\ B4 &= \mathbf{M}_p^A - \hat{\Omega}_p(I_p+I_M)\Omega_p \end{aligned} \tag{5}$$

$$T_{e-p} = \begin{bmatrix} C_{\theta_p}C_{\psi_p} & C_{\theta_p}S_{\psi_p} & -S_{\theta_p}\\ S_{\phi_p}S_{\theta_p}C_{\psi_p} - C_{\phi_p}S_{\psi_p} & S_{\phi_p}S_{\theta_p}S_{\psi_p} + C_{\phi_p}C_{\psi_p} & S_{\phi_p}C_{\theta_p}\\ C_{\phi_p}S_{\theta_p}C_{\psi_p} + S_{\phi_p}S_{\psi_p} & C_{\phi_p}S_{\theta_p}S_{\psi_p} - S_{\phi_p}C_{\psi_p} & C_{\phi_p}C_{\theta_p}\end{bmatrix} \tag{6}$$

$$T_{e-b} = \begin{bmatrix} C_{\theta_b} C_{\psi_b} & C_{\theta_b} S_{\psi_b} & -S_{\theta_b} \\ S_{\phi_b} S_{\theta_b} C_{\psi_b} - C_{\phi_b} S_{\psi_b} & S_{\phi_b} S_{\theta_b} S_{\psi_b} + C_{\phi_b} C_{\psi_b} & S_{\phi_b} C_{\theta_b} \\ C_{\phi_b} S_{\theta_b} C_{\psi_b} + S_{\phi_b} S_{\psi_b} & C_{\phi_b} S_{\theta_b} S_{\psi_b} - S_{\phi_b} C_{\psi_b} & C_{\phi_b} C_{\theta_b} \end{bmatrix} \tag{7}$$

Define the antisymmetric matrix form of the vector $\mathbf{a} = [a_x, a_y, a_z]^T$ as:

$$\hat{\mathbf{a}} = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix}$$

Where $\Omega_b = (p_b, q_b, r_b)^{\mathrm{T}}$, $\Omega_p = (p_p, q_p, r_p)^{\mathrm{T}}$, $V_c = (u_c, v_c, w_c)^{\mathrm{T}}$. M_b denotes the mass matrix of the payload. R_{cb} denotes the vector from the joint c to the payload center of gravity in the $O_c X_c Y_c Z_c$ coordinate system. T_{e-b} denotes the transformation matrix from the$O_e X_e Y_e Z_e$ coordinate system to the $O_b X_b Y_b Z_b$ coordinate system. I_b denotes the inertia matrix of the payload. F_b^A denotes the aerodynamic force vector of the payload. F_b^G denotes the gravity vector exerted by the payload. M_p denotes the mass matrix of the parafoil. M_F denotes the apparent mass matrix of the parafoil. R_{cp} denotes the vector from the joint c to the parafoil center of gravity in the $O_c X_c Y_c Z_c$ coordinate system. T_{e-p} denotes the conversion matrix from the $O_e X_e Y_e Z_e$ coordinate system to the $O_p X_p Y_p Z_p$ coordinate system. I_p denotes the inertia matrix of the parafoil. I_F denotes the apparent inertia matrix of the parafoil. F_p^A denotes the aerodynamic force vector exerted by the parafoil. F_p^G denotes the gravity vector exerted by the parafoil. $\mathbf{M}_p^A$ denotes the aerodynamic moment vector exerted by the parafoil. F_c denotes the internal joint force vector at the joint c. Detailed explanations are given in the references [7–9].

2.3. MC-DSAC-T Algorithm

Fig. 2 illustrates the core process of the algorithm updating the network, which primarily involves the agent interacting with the environment to collect data and using that data to optimize the network parameters. Where t denotes the time step, s_t denotes the state, a_t denotes the action selected according to the policy π_ϕ in the state s_t; Adopt action in the state to interact with the environment to transfer the state from s_t to s_{t+1}. $\pi_i(a|s)$ denotes the probability of obtaining an action by sampling in the state s, and the subscript i denotes the corresponding Actor network parameters, $i = (\phi, \bar{\phi})$. $Q_j(s,a)$ denotes the value estimation of the action taken on the state and the output of the mean part of the Critcic network, and the subscript j denotes the corresponding Critic network parameters, $j = (\theta_1, \bar{\theta}_1, \theta_2, \bar{\theta}_2)$. The $\mathcal{B}$ (Replay buffer) stores the set of sample data$(s_t, a_t, r_t, s_{t+1}, d)$ obtained at each time step, d indicating whether the sampling of a target trajectory is finished or not. The data stored in $\mathcal{B}$ is used to train the Actor

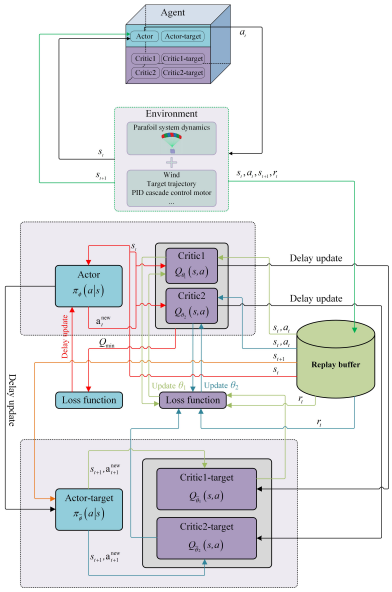


Fig. 2. Algorithm structure

(Policy) network and Critic network. The target networks employ delayed updates and soft updates to enhance training stability. The Gaussian distribution is used in the Critic network to dynamically adjust the gradient size according to the variance information when facing different reward scales, to avoid the training instability caused by the change of reward scale. The network with two pairs of critics is used to suppress the overestimation of the Q-value.

2.3.1. *Objective Function*

The DRL algorithm adopted in this paper is based on the deterministic policy version of the original DSAC-T algorithm. To enhance stability during training, incorporating the Monte Carlo thinking to constrain the update amplitude of the policy gradient, thereby reducing fluctuations in the policy iterative update process. The modified algorithm is denoted as MC-DSAC-T. The following sections will introduce the objective functions for the Actor and Critic networks in the MC-DSAC-T algorithm.

2.3.2. *Objective Function of Critic*

a) The minimized objective function

$$J_{\mathcal{Z}}(\theta) = \mathop{\mathbb{E}}_{(s,a)\sim\mathcal{B}}[D_{\mathrm{KL}}(\mathcal{T}_{\mathcal{D}}^{\pi_{\bar{\phi}}} \mathcal{Z}_{\bar{\theta}}(\cdot|s,a), \mathcal{Z}_{\theta}(\cdot|s,a))] \tag{8}$$

A dynamic adjustment parameter ω is introduced to update the gradient:

$$J_{\mathcal{Z}}^{scale}(\theta) = \omega \mathop{\mathbb{E}}_{(s,a)\sim\mathcal{B}}[D_{\mathrm{KL}}(\mathcal{T}_{\mathcal{D}}^{\pi_{\bar{\phi}}} \mathcal{Z}_{\bar{\theta}}(\cdot|s,a), \mathcal{Z}_{\theta}(\cdot|s,a))] \tag{9}$$

Where D_{KL} denotes KL divergence, which is used to measure the similarity between two distributions, and its

value is inversely proportional to the similarity degree. $\mathcal{Z}_\theta(\cdot|s,a) = \mathcal{N}(Q_\theta(s,a),\sigma_\theta(s,a)^2)$ is Gaussian distribution function, where $Q_\theta(s,a)$ and $\sigma_\theta(s,a)$ is the mean and standard deviation of the network output of the Critic; $\mathcal{T}_D^{\pi_{\bar{\phi}}}\mathcal{Z}_{\bar{\theta}}(\cdot|s,a) = r + \gamma * \mathcal{Z}_{\bar{\theta}}(\cdot|s,a)$, where r is the reward value and γ $(0 \leq \gamma \leq 1)$ is the discount factor; After introducing the gradient scaling factor ω, $J_{\mathcal{Z}}(\theta)$ is denoted as $J_{\mathcal{Z}}^{\text{scale}}(\theta)$.

b) The specific form of the objective function

$$J_{\mathcal{Z}}^{scale}(\theta_i) \approx (\omega_i + \epsilon_\omega)\mathbb{E}[-\frac{\left(y_q^{min} - Q_{\theta_i}(s,a)\right)}{\sigma_{\theta_i}(s,a)^2 + \epsilon}Q_{\theta_i}(s,a) - \frac{\left(C(y_z^{min},b_i) - Q_{\theta_i}(s,a)\right)^2 - \sigma_{\theta_i}(s,a)^2}{\sigma_{\theta_i}(s,a)^3 + \epsilon}\sigma_{\theta_i}(s,a)] \tag{10}$$

The relevant variables involved in the equation 10 are calculated in the following way.

$$\begin{aligned} &\bar{i} := \arg\min_{i=1,2} Q_{\bar{\theta}_i}(s^{'},a^{'})|a^{'} \sim \pi_{\overline{\phi}}(\cdot|s^{'}) \\ &y_q^{min} = r + \gamma Q_{\bar{\theta}_{\bar{i}}}(s^{'},a^{'}) \\ &y_z^{min} = r + \gamma Z\left(s^{'},a^{'}\right)\Big|_{Z(s^{'},a^{'})\sim\mathcal{Z}_{\bar{\theta}_{\bar{i}}}(s^{'},a^{'})} \\ &C(y_z,b) := clip(y_z, Q_\theta(s,a) - b, Q_\theta(s,a) + b) \end{aligned} \tag{11}$$

Where ω_i is the square of the mean smooth update value of the standard deviation of the network output of the main Critic (see equation 12 for the update, taking $\xi = 3$, $0 < \tau \leq 1$), b_i combined with the $Q_{\theta_i}(s,a)$ output of the mean part of the main Critic network, the Gaussian distribution sampling value of the Critic network is limited to $[Q_{\theta_i}(s,a) - b_i, Q_{\theta_i}(s,a) + b_i]$, and its update is shown in the equation (13). ϵ_ω and ϵ are two small positive numbers, ϵ_ω is to prevent the gradient from disappearing and ϵ is to prevent the gradient from exploding.

$$\omega_i \leftarrow \tau \mathop{\mathbb{E}}_{(s,a)\sim B}[\sigma_{\theta_i}(s,a)^2] + (1-\tau)\omega_i \tag{12}$$

$$b_i \leftarrow \tau\xi \mathop{\mathbb{E}}_{(s,a)\sim B}[\sigma_{\theta_i}(s,a)] + (1-\tau)b_i \tag{13}$$

2.3.3. *Objective Function of Actor*

The minimized objective function

$$J_\pi(\phi) = \mathbb{E}_{s\sim\mathcal{B},a\sim\pi_\phi}[G_t * \min\left(Q_{\theta_i}(s,a)\right)] \tag{14}$$

Where, G_t denotes the cumulative discounted reward obtained from the time step t to the end of the sampling at time step N, by using the current policy π_ϕ to sample action and interact with the environment.

See Algorithm 1. for the pseudocode of the MC-DSAC-T algorithm.

Algorithm 1. MC-DSAC-T

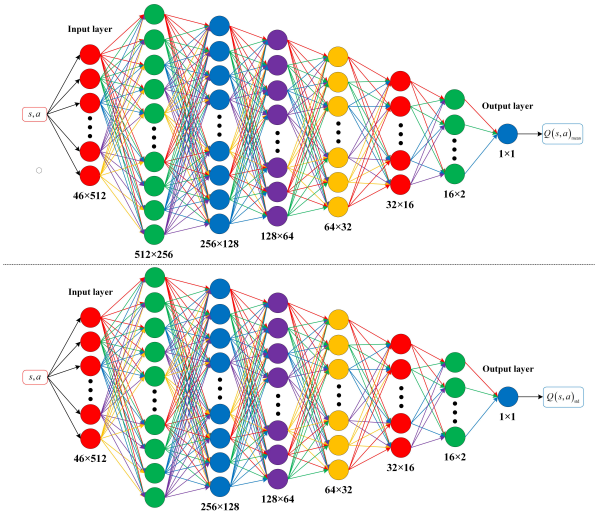
Input:θ_1, θ_2, ϕ, $\beta_{\mathcal{Z}}$, β_π,τ, M, $\mathcal{B}$
Output: θ_1, θ_2, ϕ
Initialize target networks: $\overline{\theta}_1 \leftarrow \theta_1, \overline{\theta}_2 \leftarrow \theta_2, \overline{\phi} \leftarrow \phi$
for each iteration **do**
  for each trajectory in M **do**
    for each sampling step **do**
      Calculate action $a_t \sim \pi(a_t|s_t)$
      Get reward r_t and new state s_{t+1}
      Store samples $(s_t, a_t, r_t, s_{t+1}, d)$ in the $\mathcal{B}$
  The network parameters are updated using the data stored in $\mathcal{B}$:
    Critic networks: $\theta \leftarrow \theta - \beta_{\mathcal{Z}}\nabla_\theta J_{\mathcal{Z}}^{\text{scale}}(\theta)$
    The delay update:
      Actor networks: $\phi \leftarrow \phi + \beta_\pi \nabla_\phi J_\pi(\phi)$
      Target networks: $\overline{\theta} \leftarrow \tau\theta + (1-\tau)\overline{\theta}$, $\overline{\phi} \leftarrow \tau\phi + (1-\tau)\overline{\phi}$
  Clear $\mathcal{B}$

2.3.4. *Neural Network Structure*

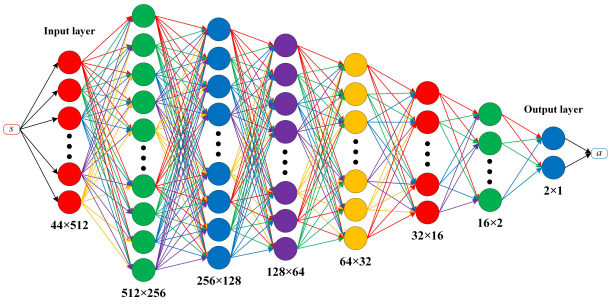
The Actor network and Critic network of the MC-DSAC-T algorithm use MLP and the hidden layer activation function is all Tanh function. The output layer activation function of the former and latter is the Sigmoid function and is not used, respectively. The Critic network includes the mean and standard deviation networks, which have the same network structure. Only the input and output layers are different between the two networks. A 44-dimensional state vector and a two-dimensional action vector are inputs to the input layer of Critic network, which output values as the mean and standard deviation of the Q-value function. In contrast, the Actor network has only a 44-dimensional state vector as input, and its output value is a two-dimensional action vector used to compute the amount of left and right control of the motor. The neural network structure is shown in Fig.3.

2.4. Reinforcement Learning

In reinforcement learning, for an agent to learn, it first needs to input the current state s_t of the parafoil system into the main Actor network; then, the action a_t output by the network is used to interact with the environment, and the corresponding reward r_t is obtained. Finally, the agent can use information such as states, actions, and rewards to learn. Therefore, the state space, action space, and reward function directly influence the final learning outcome, and the definitions of these three components will be introduced below.



(a) Critic network



(b) Actor network

Fig. 3. Neural networks structure

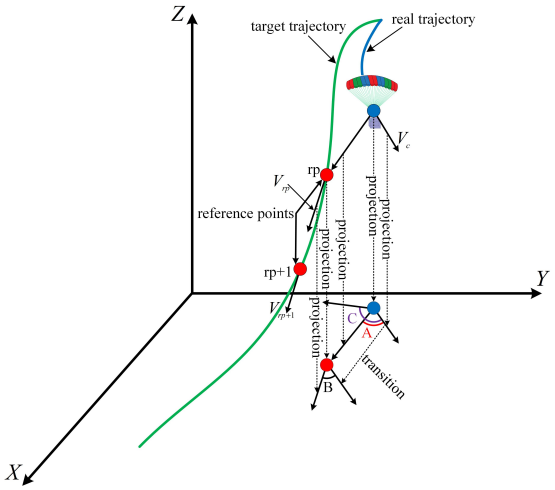


Fig. 4. Schematic of some observation states

2.4.1. *State Space*

Input states determine the controller output, so multiple observation states are employed to construct the state representations to enhance the robustness of the controller. To obtain effective observation values, the short-range and long-range distances are subdivided according to the height difference between the point on the target trajectory and the actual position of the parafoil system. The current altitude of the parafoil system is the reference altitude, and the short-range and long-range distances are taken 50 meters and 200 meters down from the reference altitude, respectively. The short-range and long-range distances take six and seven target points as reference points to calculate the corresponding observation values, and the target points can be obtained according to the interpolation function of the target trajectory height. The observation states in the short-range distance representations include horizontal velocity angle error, heading angle error, glide ratio error, and horizontal distance error between the parafoil system and the target points. The observation states in the long-range distance representations include heading angle error between the parafoil system and the target points, and the horizontal curvature of the trajectory segment contained between two neighboring target points. As well as the current wind velocity (three-dimensional), the Euler angles of the parafoil (three-dimensional), and the actual motor control value (two-dimensional). Each observation state in the short-range and long-range distances has six observation values. Some observation states above are shown in Fig. 4.

In Fig. 4, the green trajectory is the target trajectory tracked by the parafoil system, and the blue color denotes the trajectory generated by the parafoil system during the actual flight. The red and blue points are the points on the target trajectory and the actual position of the parafoil system, respectively. The symbol A denotes the heading angle error, and B and C respectively denote the horizontal velocity angle error between the parafoil system with the target velocity and wind velocity. The curvature can be computed by combining the change in horizontal angle between vectors V_{rp+1} and V_{rp} with the horizontal trajectory length between points rp and rp+1.

2.4.2. *Action Space*

During the flight, the motor dynamically controls the parafoil system by precisely adjusting the left and right control lines. To align with the control rules of the motor, the action space is defined as a two-dimensional continuous vector a=[a_{left}, a_{right}], which corresponds to the motor control output. This vector a is transformed into the control output of the motor via a mapping formula:

$$a_i \times 1.25 - 0.125$$

The value range for each component of the vector and motor control output is [0, 1].

2.4.3. Reward Function

The reward function is crucial in the iterative update of policy parameters. The policy directly influences the control output of the motor, and consequently, the reward function directly impacts the control effect. In this study, the horizontal distance error (m) and horizontal velocity angle error (deg) between the actual flight path and target path at each time step were considered to make the reward function effectively guide the update of the policy network parameters.The details are as follows:

$$distance\_error = \frac{1}{N}\sum_{i=1}^{N}\sqrt{\left(x_i' - x_i\right)^2 + \left(y_i' - y_i\right)^2} \quad (15)$$

$$direction\_error = \frac{1}{N}\sum_{i=1}^{N}|angle(x_{i+1} - x_i, y_{i+1} - y_i) - real\_angle\_velocity| \quad (16)$$

$$reward = -\left(distance\_error * direction\_error\right) \quad (17)$$

Where the simulation time step N for each time step is set to 100, and the distance error and direction error are calculated in the unit of simulation time step with intensive reward. At the same altitude, $(x_i^{'}, y_i^{'})$ denotes the horizontal coordinate corresponding to the actual flight trajectory of the parafoil system and (x_i, y_i) denotes the horizontal coordinates corresponding to the target trajectory. (x_{i+1}, y_{i+1}) denotes the horizontal coordinate adjacent to (x_i, y_i). In particular, because adjacent coordinate subtraction is used to calculate the velocity on the target trajectory, $N+1$ points should be selected, and the height Z_{N+1} of the $N+1$ point is obtained by the approximate operation of the corresponding height of the $N-1$ and N simulation time steps of the parafoil system: $Z_{N+1} = 2 * Z_N - Z_{N-1}$.

2.5. PID

PID is a classical control method that consists of four main components: the proportion coefficient K_p, the integration coefficient K_i, the differentiation coefficient K_d, and the error function Error. Among them, the Error has an important influence on the control effect, usually, after the Error is determined, the three coefficients of K_p, K_i, and K_d can be found in a set of effective control parameters through the finite trial-and-error method. The error of the PID controller in this study is described below, and the values of the coefficients of K_p, K_i, and K_d found by the trial-and-error method are given in the experimental section below. The computational formula for the error function in this study is as follows:

$$Error = V_e * D_e * (1 + W_e) \quad (18)$$

Table 1. Wind Field Parameters

Wind Type	Speed	Direction
Mean Wind	$Uniform_{[0,\ 3]}$	$Uniform_{[-180,\ 180]}$
Gust Wind	$Uniform_{[0,\ 1.5]}$	$Random_{[-180,\ 180]}$
Random Wind	$Uniform_{[0,\ 0.5]}$	$Random_{[-180,\ 180]}$

Table 2. Wind Field Parameters

Wind Type	Level
Mean Wind	Speed*(altitude/70)$^{0.25}$
Gust Wind	Speed*random(0,1)$_{float}$
Random Wind	normal(0, Speed / 3, altitude)

where V_e denotes the angle error between the actual velocity of the parafoil system and the target velocity. D_e denotes the horizontal distance error between the actual position of the parafoil system and the target position. W_e denotes the horizontal wind speed multiplied by the angle difference between the horizontal wind direction and the heading angle of the parafoil system.

3. Results and discussion

3.1. Experiment Settings

Experiments are conducted on CPU, using the Ubuntu 24.04 operating system, based on Python 3.11 and Pytorch 2.1.0 deep learning framework.

3.1.1. Dataset and Environment

The training data set used in the experiment is 1000 trajectories generated by introducing random action sequences (unilateral pull-down) to interact with the simulation environment under the superposition of three wind fields of the average wind, gust wind, and random wind (50 trajectories were generated separately for the test data set). Table1 shows how to obtain the wind speed and direction for the three winds. The wind level is obtained based on the wind speed and height obtained by sampling the uniform distribution, and the calculation method is shown in Table 1. The wind level is calculated from the wind speed and the current altitude: decomposing the wind level into three axes according to the wind direction to acquire the wind vector.

3.1.2. Parameter Setting

This section describes the setting of the experimentally relevant parameters and the relevant details during the training process. The relevant parameters of the parafoil system are shown in Table 3. In addition, the relevant parameters involved in the training process of the neural

Table 3. Physical parameters of the parafoil system

Physical parameters	Value
Wing Span	7.5m
Wing Area	$28m^2$
Geometric Chord Length	3.75m
Leading Edge Line Length	7.5m
Canopy Thickness	0.675m
Trailing Edge Line Length	8.0m
Suspension Line Length	1.0m
Parafoil Mass	5kg
Payload Mass	135kg

Table 4. Hyperparameters of the MC-DSAC-T algorithm

Hyperparameter	Value
Optimizer	Adam (default)
Actor learning rate	1e-3
Critic learning rate	1e-3
Learning rate decay	0.95
Discount factor (γ)	0.99
Soft update factor (τ)	0.01
Delayed update	1
Batch size M	100
Torch random seed	1009
Random number seed	0
Random wind seed	int(0,65536)$_{random\_number\_generator}$

networks controller are shown in Table 4. The Actor network and Critic networks are described in detail above. The parameter settings in Table 3 and Table 4 were used for simulation experiments; during training, trajectories were randomly selected from the training data set, segments of the selected trajectory were randomly intercepted as the target trajectory, and the wind field was consistent with the wind field when generating the target trajectory. An iterative update of the model requires M target trajectory data. When sampling each of the M target trajectories, the following two principles are followed:

1) If the termination condition is satisfied and the total number of sampled target trajectories (including this one) does not reach the batch size M, then replace the trajectory and continue sampling;

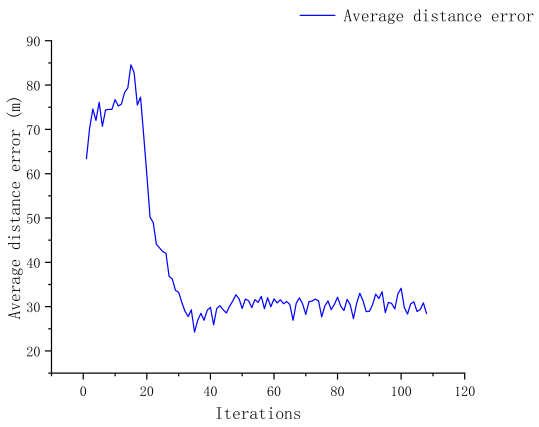
2) If the termination condition is satisfied and the number of sampled target trajectories reaches the batch size M, the sampling is terminated, and the sampled data of this batch is used for model training.

The termination condition for sampling each target trajectory is as follows.

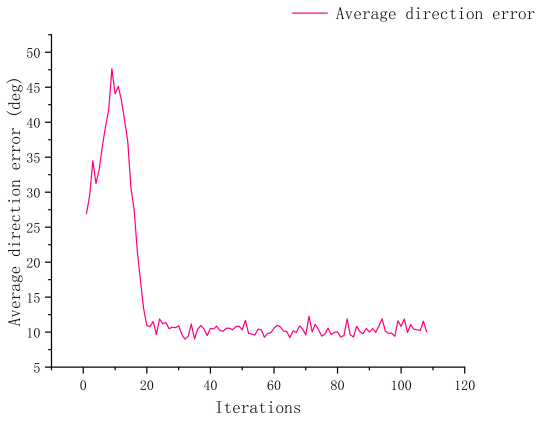
1) The time step of tracking the target trajectory reaches the set maximum value of 100;

2) The horizontal distance error with the target trajectory at the same height is more than 200 meters.

3.2. Training Results and Discussion



(a) Distance error



(b) Direction error

Fig. 5. Train curve

Fig. 5 illustrates the training process of the MC-DSAC-T algorithm. Since the initialization of the weights and bias values in both the Actor network and the Critic network are random, the control output from the Actor network at the beginning of the training is not effective in controlling the parafoil system to track the target trajectory. In addition, the initial Q-value estimation of the Critic network is not sufficient to properly guide the update of the Actor network. As a result, the distance and direction errors increase rather than decrease at the beginning of training. However, as the number of training iterations increases, the Critic network gradually converges to accurate Q-value. This convergence allows the Actor network to learn more efficient control parameters, thus reducing the distance and direction errors over time.

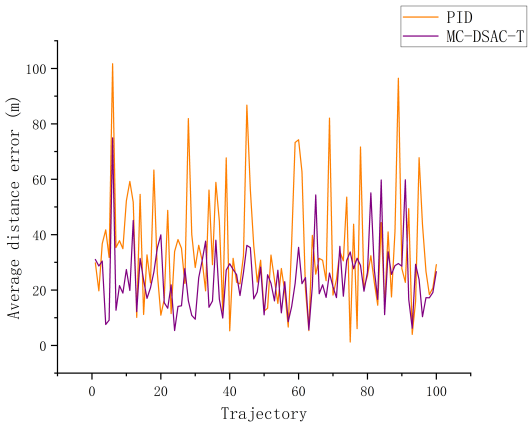
Table 5. Mean Values of Performance Evaluation Indices for Two Controllers

Performance Index	PID	MC-DSAC-T
Average distance error	34.58m	24.21m
Average direction error	5.59deg	8.78deg
Average energy consumption	0.05	0.35

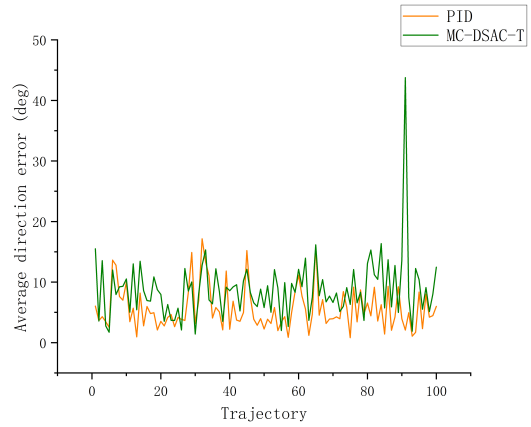
3.3. Test Results and Discussion

Fig. 6 illustrates the test results of controlling the parafoil system to track 100 target trajectories using the PID controller(K_p= 4.5, K_i= 0.05,K_d= 0.1) and the neural network controller trained by the MC-DSAC-T method. The average distance error between the actual flight trajectory of the parafoil system and the target trajectory, the average velocity angle error, and the average motor energy consumption in the tracking control phase are considered the controller performance evaluation indexes. The two controllers control the parafoil system to track each target trajectory, Figs. 6a, 6b, and 6c show the average distance error, velocity angle error, and energy consumption of tracking each target trajectory, respectively. Table 5 shows the mean values of the three performance evaluation index objects of the two controllers controlling the parafoil system to track 100 target trajectories. Fig. 7 shows the tracking control effect of two controllers controlling the parafoil system to track the same target trajectory(green). Combining the data in Fig.7, Fig.6, and Table5, it can be concluded that the PID controller can control the velocity angle of the parafoil system closer to the velocity angle of the target trajectory, and consume less energy at the same time. However, the neural network controller can control the parafoil system closer to the target trajectory, and the control accuracy of the distance is better than that of the PID controller.

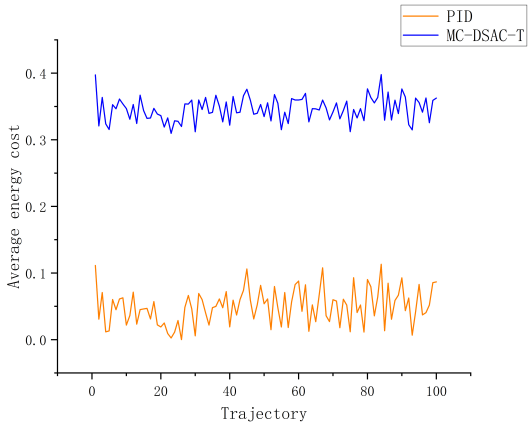
The neural network controller, trained by the MC-DSAC-T algorithm, uses the bilateral pull-down to control the parafoil system, while the PID controller uses the unilateral pull-down. The bilateral pull-down can control the parafoil system to realize the operation of deceleration and rapid altitude reduction, which is difficult to achieve by unilateral pull-down. However, when controlling turning and other control operations, the extra energy after the left and right pull-down of the bilateral pull-down cancel each other is equivalent to using only the unilateral pull-down, so the energy consumption of the bilateral pull-down is higher than that of the unilateral pull-down. The error function of the PID controller and the reward function of reinforcement learning both consider the distance error and direction error. Combining the test results of two controllers, the neural network controller is superior to the PID controller



(a) Distance error

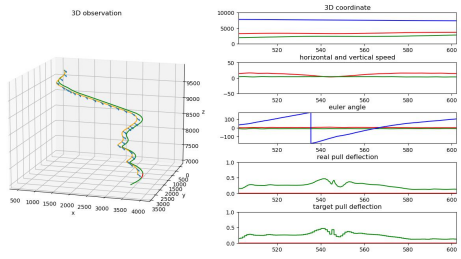


(b) Direction error

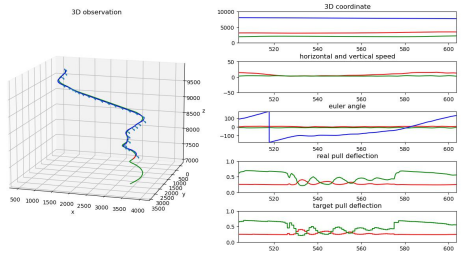


(c) Energy consumption

Fig. 6. Test results



(a) PID



(b) MC-DSAC-T

Fig. 7. Control effect comparison

in the overall control accuracy of direction and distance.

4. Conclusion

In this study, the single-episode reward information was introduced into the policy objective function of the deterministic DSAC-T algorithm to get the MC-DSAC-T algorithm. Since the single-episode reward information is obtained through real sampling, it is unbiased and can effectively reduce the bias in the temporal difference method used by the DSAC-T algorithm, thereby enhancing the stability of the policy iterative update.

Aiming the trajectory tracking control problem of the parafoil system, the MC-DSAC-T algorithm was employed to train a neural network controller. Experimental results demonstrate that under wind conditions, the trained neural network controller could effectively control the parafoil system to track target trajectories with accuracy. This study may provide some help for research in reinforcement learning algorithms and the control of the parafoil system.

However, the study utilized a nine-degree-of-freedom (9-DOF) dynamic model of the parafoil system, which differs from the real-world conditions. Future work can explore using higher-DOF dynamic models for the simulation studies and semi-physical simulation experiments. Terrain threats can also be added simultaneously to realize the application of the parafoil system in complex terrains.

References

[1] H. Zhu, Q. Sun, X. Liu, J. Liu, H. Sun, W. Wu, P. Tan, and Z. Chen, (2021) *"Fluid–structure interaction-based aerodynamic modeling for flight dynamics simulation of parafoil system"* **Nonlinear Dynamics *104***(4): 3445–3466. DOI: 10.1007/s11071-021-06486-0.

[2] Y. Li, M. Zhao, M. Yao, Q. Chen, R. Guo, T. Sun, T. Jiang, and Z. Zhao, (2020) *"6-DOF Modeling and 3D Trajectory Tracking Control of a Powered Parafoil System"* **IEEE Access *8***: 151087–151105. DOI: 10.1109/ACCESS.2020.3016669.

[3] Q. Sun, L. Yu, Y. Zheng, J. Tao, H. Sun, M. Sun, M. Dehmer, and Z. Chen, (2022) *"Trajectory tracking control of powered parafoil system based on sliding mode control in a complex environment"* **Aerospace Science and Technology *122***: 107406. DOI: https://doi.org/10.1016/j.ast.2022.107406.

[4] T. Jann. "Aerodynamic model identification and GNC design for the parafoil-load system ALEX". In: *16th AIAA Aerodynamic Decelerator Systems Technology Conference and Seminar*. American Institute of Aeronautics and Astronautics (AIAA), 2001. DOI: 10.2514/6.2001-2015. eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2001-2015.

[5] Z. Zhang, Z. Zhao, and Y. Fu, (2019) *"Dynamics analysis and simulation of six DOF parafoil system"* **Cluster Computing *22***(5): 12669–12680. DOI: 10.1007/s10586-018-1720-3.

[6] N. J. Slegers, (2010) *"Effects of Canopy-Payload Relative Motion on Control of Autonomous Parafoils"* **Journal of Guidance, Control, and Dynamics *33***(1): 116–125. DOI: 10.2514/1.44564. eprint: https://doi.org/10.2514/1.44564.

[7] E. Mooij, Q. Wijnands, and B. Schat. "9 DoF parafoil/payload simulator development and validation". In: *AIAA Modeling and Simulation Technologies Conference and Exhibit*. 2003, 5459. DOI: 10.2514/6.2003-5459. eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2003-5459.

[8] O. Prakash and N. Ananthkrishnan. "Modeling and simulation of 9-DOF parafoil-payload system flight dynamics". In: *AIAA Atmospheric Flight Mechanics Conference and Exhibit*. 2006, 6130. DOI: 10.2514/6.2006-6130. eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2006-6130.

[9] Y. Gang, (2015) *"Nine-degree of Freedom Modeling and Flight Dynamic Analysis of Parafoil Aerial Delivery System"* **Procedia Engineering *99***: 866–872. DOI: 10.1016/j.proeng.2014.12.614.

[10] N. Slegers and M. Costello, (2005) *"Model Predictive Control of A Parafoil and Payload System"* **Journal of Guidance, Control, and Dynamics *28***(4): 816–821. DOI: 10.2514/1.12251. eprint: https://doi.org/10.2514/1.12251.

[11] J. Tao, Q. Sun, P. Tan, Z. Chen, and Y. He, (2016) *"Active disturbance rejection control (ADRC)-based autonomous homing control of powered parafoils"* **Nonlinear Dynamics *86***(3): 1461–1476. DOI: 10.1007/s11071-016-2972-1.

[12] J. Tao, Q. Sun, H. Sun, Z. Chen, M. Dehmer, and M. Sun, (2017) *"Dynamic Modeling and Trajectory Tracking Control of Parafoil System in Wind Environments"* **IEEE/ASME Transactions on Mechatronics *22***(6): 2736–2745. DOI: 10.1109/TMECH.2017.2766882.

[13] Y. Zheng, J. Tao, Q. Sun, H. Sun, Z. Chen, M. Sun, and G. Xie, (2023) *"Sideslip angle estimation based active disturbance rejection 3D trajectory tracking control for powered parafoil system and hardware-in-the-loop simulation verification"* **Aerospace Science and Technology *141***: 108497. DOI: https://doi.org/10.1016/j.ast.2023.108497.

[14] W. He, Y. Zheng, J. Wen, J. Tao, and Q. Sun, (2024) *"Path Following Control of Parafoil System Based on SG-LOS and Improved ADRC Tuned by MSMPA"* **IEEE Transactions on Circuits and Systems II: Express Briefs *71***(8): 3895–3899. DOI: 10.1109/TCSII.2024.3377010.

[15] H. Sun, Q. Sun, J. Tao, S. Luo, and Z. Chen. "A flight control system for parafoils based on improved PID control approach". In: *2017 36th Chinese Control Conference (CCC)*. 2017, 1168–1173. DOI: 10.23919/ChiCC.2017.8027506.

[16] J. Tao, M. Dehmer, G. Xie, and Q. Zhou, (2019) *"A Generalized Predictive Control-Based Path Following Method for Parafoil Systems in Wind Environments"* **IEEE Access *7***: 42586–42595. DOI: 10.1109/ACCESS.2019.2905632.

[17] L. Zhao, W. He, F. Lv, and W. Xiaoguang, (2020) *"Trajectory Tracking Control for Parafoil Systems Based on the Model-Free Adaptive Control Method"* **IEEE Access *8***: 152620–152636. DOI: 10.1109/ACCESS.2020.3017539.

[18] Y. P. Wang and H. R. Dong. "The Application of Adaptive PSO in PID Parameter Optimization of Unmanned Powered Parafoil". In: *Modern Technologies in Materials, Mechanics and Intelligent Systems*. ***1049***. Advanced Materials Research. Trans Tech Publications Ltd, 2014, 1094–1097. DOI: 10.4028/www.scientific.net/AMR.1049-1050.1094.

[19] H. Jia, Q. Sun, and Z. Chen. "Application of Single Neuron LADRC in Trajectory Tracking Control of Parafoil System". In: *Proceedings of 2018 Chinese Intelligent Systems Conference*. Ed. by Y. Jia, J. Du, and W. Zhang. Singapore: Springer Singapore, 2019, 33–42. DOI: 10.1007/978-981-13-2288-4_4.

[20] Y. Zheng, J. Tao, Q. Sun, X. Zeng, H. Sun, M. Sun, and Z. Chen, (2023) *"DDPG-based active disturbance rejection 3D path-following control for powered parafoil under wind disturbances"* **Nonlinear Dynamics *111***(12): 11205–11221. DOI: 10.1007/s11071-023-08444-4.

[21] Y. Zheng, J. Tao, Q. Sun, H. Sun, M. Sun, and Z. Chen. "Path following control for powered parafoil system based on TD3-LADRC". In: *2023 42nd Chinese Control Conference (CCC)*. 2023, 2364–2369. DOI: 10.23919/CCC58697.2023.10240655.

[22] Y. Zheng, J. Tao, Q. Sun, J. Yang, H. Sun, M. Sun, and Z. Chen. "Intelligent Trajectory Tracking Control of Unmanned Parafoil System Based on SAC Optimized LADRC". In: *Neural Information Processing*. Ed. by B. Luo, L. Cheng, Z.-G. Wu, H. Li, and C. Li. Singapore: Springer Nature Singapore, 2024, 97–108. DOI: 10.1007/978-981-99-8073-4_8.

[23] Z. Wei and Z. Shao. "Precision landing of autonomous parafoil system via deep reinforcement learning". In: *2024 IEEE Aerospace Conference*. IEEE. 2024, 1–10. DOI: 10.1109/AERO58975.2024.10521056.

[24] Z. Wei, Y. Gao, Z. Shao, and C. Wang, (2024) *"Dynamic-model-based closed-loop guidance and control for heavy parafoil system precision landing"* **Aerospace Science and Technology *146***: 108964. DOI: 10.1016/j.ast.2024.108964.

[25] J. Duan, W. Wang, L. Xiao, J. Gao, and S. E. Li, (2023) *"DSAC-T: Distributional Soft Actor-Critic with Three Refinements"* **arXiv preprint arXiv:2310.05858**: DOI: 10.48550/arXiv.2310.05858. arXiv: 2310.05858 [cs.LG].