SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

March 2025, Vol. 68, Iss. 3, 132201:1–132201:23
https://doi.org/10.1007/s11432-023-4217-8

A dynamic control decision approach for fixed-wing aircraft games via hybrid action reinforcement learning

Xing ZHUANG[1], Dongguang LI[1], Hanyu LI[1], Yue WANG[1*] & Jihong ZHU[2]

[1]*Science and Technology on Electromechanical Dynamic Control Laboratory,*
Beijing Institute of Technology, Beijing 100081, China
[2]*Department of Precision Instrument, Tsinghua University, Beijing 100084, China*

Received 1 August 2023/Revised 4 March 2024/Accepted 14 May 2024/Published online 10 February 2025

Abstract Autonomous decision-making is crucial for aircraft to achieve quick victories in diverse scenarios. Based on a 6-degree-of-freedom aircraft model, this paper proposes a decoupled guidance and control theory for autonomous aircraft maneuvering, distinguishing between close and long-range engagements. We introduce a method for heading attitude control to enhance stability during close-range interactions and a speed-based adaptive grid model for precise waypoint control in mid-to-long-range engagements. The paper transforms dynamic aircraft interactions into a Markov decision process and presents a hybrid discrete and continuous action reinforcement learning approach. This unified learning framework offers enhanced generalization and learning speed for dynamic aircraft adversarial processes. Experimental results indicate that in a symmetric environment, our approach rapidly achieves Nash equilibrium, securing over a 10% advantage. In unmanned aerial aircraft game control with higher maneuverability, the probability of gaining a situational advantage increases by more than 40%. Compared to similar methods, our approach demonstrates superior effectiveness in decision optimization and adversarial success probability. Furthermore, we validate the algorithm's robustness and adaptability in an asymmetric environment, showcasing its promising application potential in collaborative control of aircraft clusters.

Keywords intelligent air combat, unmanned aerial vehicle game, dynamic control, reinforcement learning

Citation Zhuang X, Li D G, Li H Y, et al. A dynamic control decision approach for fixed-wing aircraft games via hybrid action reinforcement learning. Sci China Inf Sci, 2025, 68(3): 132201, https://doi.org/10.1007/s11432-023-4217-8

1 Introduction

In recent years, autonomous drones have been gradually applied across various industries [1, 2]. Researching unmanned aerial vehicles (UAV) dynamic game problems enables effective resolution of their autonomous decision-making and control challenges in scenarios like interception, interference, and complex environmental threats. This allows UAVs to achieve specific game objectives, including disaster relief, counter-terrorism, and hazardous environment reconnaissance [3]. In dynamic game scenarios, UAV models can be categorized into fixed-wing and rotary-wing types. Fixed-wing UAVs have greater endurance and resilience in some complex environments, but the design of their autonomous decision-making systems presents greater challenges [4]. Due to the need to control multiple degrees of freedom and parameters such as pitch, roll, yaw, and speed [5], the dynamic game process of fixed-wing UAVs exhibits high dimensionality and strong coupling, requiring faster response times and more accurate attitude calculation between control strategies and actual flight states. Furthermore, given the complex and dynamic nature of current and future scenarios in aircraft adversarial games, research on the generalization and robustness of game decision control algorithms is crucial.

Currently, in the existing research and applications [6, 7], the dynamic game problem of UAVs can be transformed into bilateral extreme problems such as pursuit-evasion, attack-evasion, and reconnaissance-dispersal. These problems involve the autonomous decision-making and control of UAVs, which are further divided into trajectory planning, attitude control, overload control, and other aspects for investigation. For nonlinear control of vehicle attitude, Espinoza et al. [8, 9] modeled the nonlinear control

[*] Corresponding author (email: jackwy@bit.edu.cn)

system of fixed-wing UAV and employed differential methods to solve for the load factor and attitude angles. Wang et al. [10] proposed a composite adaptive fault-tolerant control strategy for quadcopter UAVs, building upon baseline sliding mode control, they integrated a neural adaptive control method, enhancing the robustness of the UAV's control strategy. Zheng et al. [11] addressed the optimal control problem of timely fault-tolerant attitude tracking for non-affine nonlinear faulty UAVs using the Lyapunov function. This type of research addresses the issue of stable attitude control of UAVs in complex environments. However, in game scenarios, where aircraft are confronted with complex gameplay behavior, the algorithmic performance of the solution is poor and unable to handle high-dynamic game decision-making problems. Therefore, in high-dynamic aircraft games, many scholars have transformed them into studies on dynamic path planning [12] and aircraft obstacle avoidance issues. Traditional approaches [13] such as A* [14] and Dijkstra [15], swarm intelligence algorithms [16] like particle swarm optimization (PSO), artificial bee colony (ABC) [17], and artificial fish swarm (AFS) [18, 19], as well as methods based on random sampling like probabilistic roadmap (PRM) and rapidly exploring random tree (RRT) [20], have been widely utilized. Specifically, Huang et al. [21] applied adaptive adjustment parameters, cylindrical vectors, and different evolution operators to the PSO algorithm (ACVDEPSO), efficiently generating higher-quality paths for UAVs in complex three-dimensional environments. Zhou et al. [22] proposed a biomimetic three-dimensional spatial path planning algorithm by simulating the basic mechanism of plant growth, this algorithm addresses the dynamic obstacle avoidance path planning problem for drones in unknown environmental maps. Diao et al. [23] introduced an artificial potential field-enhanced improved rapidly exploring random tree (APF-IRRT*) path planning algorithm, improving the convergence speed and path smoothness of drone trajectory optimization. These studies have made contributions to dynamic trajectory planning, but two main issues still exist. Firstly, the performance degradation is caused by the increase in search space complexity. For example, the same method may exhibit a significant performance difference when applied to two-dimensional and three-dimensional spaces. Secondly, general path planning methods are based on the 3-degree-of-freedom (3-DOF) point mass model of the aircraft, which introduces significant errors in practical applications. In the context of dynamic game scenarios for aircraft, relying solely on generated trajectories can lead to a loss of control over the aircraft's attitude. In the second aspect, the accuracy of the model in decision-making algorithms has been increasingly emphasized. Roberge et al. [24] utilized genetic algorithm (GA) and PSO to optimize the optimal trajectory of fixed-wing UAVs in complex three-dimensional environments, they composed a rational flight path using discrete line segments, arcs, and vertical spirals. Sandberg et al. [25] proposed several autonomous trajectory generation algorithms based on the 6-DOF model. Raigoza et al. [26] solved autonomous policy-making on fixed-wing UAV collision problems based on Sandberg's approach. However, the aforementioned methods still rely on path point control and do not consider the dynamic attitude control issues in UAV games. In our preliminary work [27, 28], we have studied overload control methods for fixed-wing UAVs and conducted research on autonomous control decision-making for dynamic interception weapons. However, our approach has not addressed the issue of autonomous decision-making for both sides in the dynamic game of UAVs. Specifically, research on autonomous decision-making for red-side aircraft typically overlooks the issue of decision intelligence for blue-side aircraft. In other words, our current focus is on studying the control decisions of red-side aircraft in the face of blue-side aircraft whose decision-making capabilities can evolve.

In summary, studying autonomous decision control methods for both sides in one-on-one aircraft games aims to quickly reach a non-cooperative Nash equilibrium in adversarial scenarios. This research provides a unified learning framework for future aircraft adversarial games, offering decision control strategies for various confrontations. This equips aircraft with the ability to guide precisely in advantageous situations and maneuver to escape in disadvantaged situations. This paper proposes a decision model that combines discrete and continuous actions based on the actual 6-DOF model of the aircraft. It integrates deep reinforcement learning (DRL) methods to solve the coupled optimization problem of UAV's optimal trajectory planning and attitude control in both two-dimensional and three-dimensional spaces. This approach addresses the issues of autonomous decision-making and dynamic control for fixed-wing UAVs in non-cooperative games. The main contributions of this paper are as follows.

(1) By analyzing the relative positions and motion relationships in the dynamic game process of aircraft, this paper, based on 6-degree-of-freedom aircraft dynamic modeling, designed two control systems: waypoint control and heading control. It proposed the relationship between different control methods during the game process and the logic of control intervention. To address issues with waypoint control on rigid-body aircraft, heading attitude angles are combined with velocity vectors, establishing transfer

Table 1 Reinforcement learning applications in UAV autonomous decision-making and dynamic control.

Application	Problem description	Papers
UAV control system	Achieving stable attitude control (continuous or discrete control) [29–37]	Zhen et al., 2020; Huang et al., 2019; Bohn et al., 2023; Xie et al., 2023; Din et al., 2022; Zhang et al., 2022; Liu et al., 2022; Wei et al., 2022a; Wan et al., 2020
Path planning	Local or global trajectory control (discrete waypoints) [38–45]	Lee et al., 2019; Omoniwa et al., 2022; Xu et al., 2022b; Silvirianti and Shin, 2022; Huang et al., 2020; Byun and Nam, 2022; Puente-Castro et al., 2022; Hu et al., 2020
	Static or moving targets tracking (discrete waypoints) [46–54]	Ma et al., 2023; Li et al., 2020a; Bhagat et al., 2020; Akhloufi et al., 2019; Ajmera and Singh, 2020; Moon et al., 2021; Wang et al., 2019; Yin et al., 2019; Yu et al., 2023
	Take-off or landing control (discrete actions) [55–58]	Jiang et al., 2022; Xie et al., 2020; Mosali et al., 2022; Rodriguez-Ramos et al., 2019
Obstacle avoidance	Sensing and obstacle avoidance (continuous or discrete sensing) [59–61]	Hu et al., 2019; Ouahouah et al., 2022; Singla et al., 2021
	Dynamic or static obstacle avoidance (continuous or discrete path) [62–68]	Kim et al., 2020; Liu et al., 2019; Xu et al., 2022a; Li et al., 2020b; Zhao et al., 2017; Tu and Juang, 2023; Zhu et al., 2022

functions for stable control of aircraft mid-air attitudes and path planning.

(2) Based on a decoupled control system for heading and waypoint, this paper introduces integrated guidance control models for mid-guidance (waypoint control) and terminal guidance (heading control) in the dynamic game process of aircraft confrontation. To ensure effective coordination between the two guidance methods, the paper proposes a flight space-adaptive grid model based on aircraft heading attitude, body attitude, and velocity information. This model constrains the path search range during mid-to-long-range guidance control, enhances stability in waypoint control, and provides a favorable initial game posture for terminal guidance control in close-range engagements.

(3) We propose a dynamic game approach for 1v1 aerial combat, integrating a spatial scale-adaptive grid model. It introduces a hybrid action space design based on discrete waypoint control and continuous heading control for aircraft. Defining a Markov decision process for adversarial games, the paper constructs a reinforcement learning decision model tailored for dynamic aircraft games. This model provides a unified decision-learning framework, enhancing optimal path planning during aircraft games. It also ensures stable attitude control for tactical maneuvers like attack or interception in close-range combat. Additionally, the method offers a highly generalized decision model for autonomous control in dynamic aircraft games across different scenarios. Through the design of game objectives and reward functions, it enables offline learning for diverse adversarial tasks, further improving the efficiency and accuracy of UAVs' autonomous decision-making in unknown environments.

(4) At last, to address the issue of weak decision evolution capability in adversary aircraft during aerial combat, we propose a non-cooperative control decision framework based on reinforcement learning for dynamic decision-making in aircraft games. The framework focuses on both adversarial aircraft, enabling them to autonomously learn control decisions. This approach enhances the adversarial co-evolution of decision intelligence, significantly improving the learning efficiency and convergence speed of optimal decision-making for each aircraft.

2 Related work

DRL has played a crucial role in the autonomous decision-making of dynamic games for UAVs. It has been applied in various aspects, including UAV attitude control, obstacle avoidance, path planning, and game decision-making. In the study of such problems, DRL typically plays the role of an intelligent aircraft with environmental perception capabilities. It aims to maximize the rewards of a policy to achieve the optimal actions that lead to the desired objectives. In this process, DRL often adopts either end-to-end control (directly outputting trajectories) or two-stage control (attitude and overload controlling trajectories). Table 1 [29–68] summarizes the applications of reinforcement learning in UAV autonomous decision-making and dynamic control.

According to the classification results of RL applications by AlMahamid and Grolinger [69], the aforementioned problems are typically based on discrete actions and finite states. Currently, several algorithms

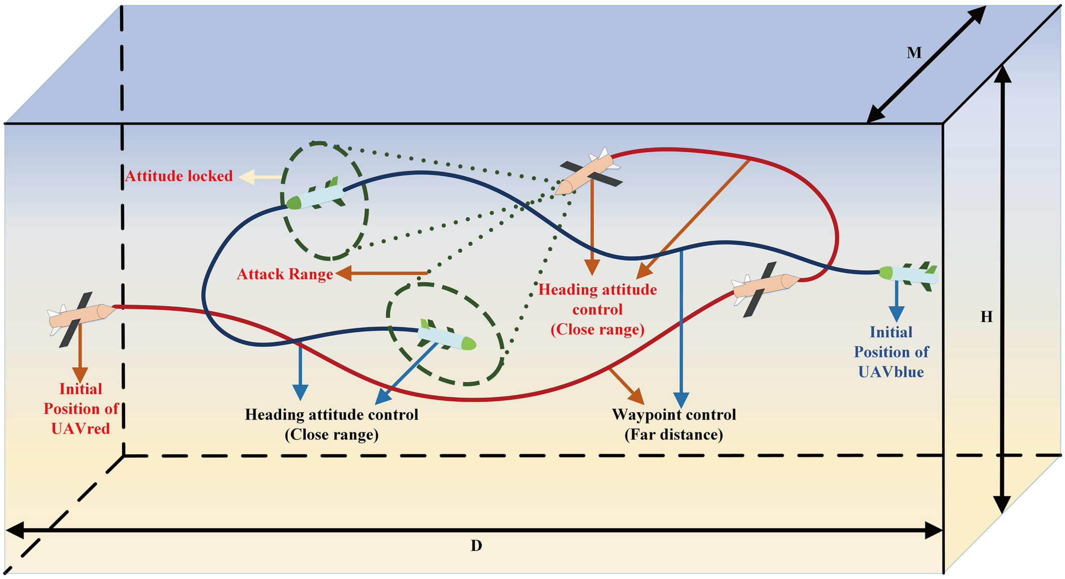
M
H
D
Attitude locked
Attack Range
Heading attitude control (Close range)
Initial Position of UAVblue
Initial Position of UAVred
Heading attitude control (Close range)
Waypoint control (Far distance)

Figure 1 (Color online) Dynamic gaming scenario of UAVs.

have shown promising results in deterministic action control, such as DDPG (deep deterministic policy gradient), TD3 (twin delayed deep deterministic policy gradient), and SAC (soft actor-critic), while others like TRPO (trust region policy optimization), PPO (proximal policy optimization algorithms), and A3C (asynchronous advantage actor-critic) are suitable for stochastic action control. These methods are commonly derived from the actor-critic (AC) framework. However, TD3, SAC, and similar algorithms have high demands on hyperparameter tuning and rely on hardware performance. TRPO exhibits weaker learning capabilities than PPO in comparable environments, and A3C consumes significant computational resources due to its asynchronous processing approach. DDPG, on the other hand, cannot handle stochastic discrete actions effectively. The paper addresses the dynamic game problem in UAVs and aims to tackle the high-dimensional state space while addressing the decision-making problem involving both discrete and continuous actions. In essence, it applies DRL methods to the research of dynamic target tracking and obstacle avoidance, providing a unified decision control framework for the attitude stability and trajectory optimization of UAVs.

3 Problem analysis and theoretical modeling

3.1 Game scenario design and description

This paper designs the game scenario for fixed-wing UAVs as depicted in Figure 1, and makes the following assumptions.

Assumption 1. The game scenario consists of two fixed-wing UAVs engaged in a 1v1 game. The UAVs have the same structure and can accurately identify each other. It is assumed that there is a satellite providing real-time detection of UAV positions, and this information is transmitted to both players in the game, disregarding communication delays and position errors.

Assumption 2. The game scenario is defined as a three-dimensional airspace with specific boundaries. The depth of the adversarial airspace in the longitudinal direction is $D = 8$ km, its width is $M = 8$ km, and the flight altitude is restricted to within $H = 4$ km.

Assumption 3. The UAV confrontation is conducted in the form of a non-cooperative game. The UAVs have radar line-of-sight angle α and a maximum attack distance d.attack. The objectives of both UAVs in the game are pursuit and evasion.

Assumption 4. Both drones are fully controlled by strategy models. The strategy algorithm outputs the next waypoint coordinates for the drones, which is defined as the waypoint control mode. The strategy algorithm continuously provides the drones with the speed tilt angle, speed yaw angle, and rate of speed change, defined as the heading attitude control.

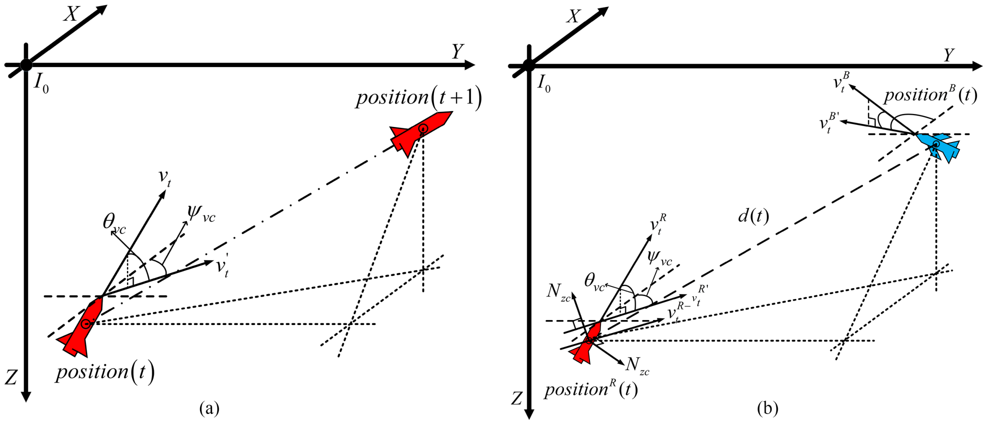

Figure 2 (Color online) Theoretical model of the autonomous control system of UAVs. (a) Waypoint control; (b) heading attitude control.

According to Figure 1, the red and blue lines represent the trajectories of the two teams' UAVs. The game process presents two types of adversarial scenarios based on distance: far and close distances.

(1) In the far-distance adversarial scenario ("distance between red and blue" > "d.attack or threshold"), the UAV primarily uses waypoint control as their main control strategy. The objective is to approach the opponent's UAV, occupy advantageous altitudes, or position themselves behind the enemy to observe and predict the opponent's actions.

(2) In the close-range adversarial scenario ("distance between red and blue" < "d.attack"), the game transitions into more intense and direct interactions, similar to a dogfight. Both UAVs possess high maneuverability, and their control strategies focus on rapid and agile heading attitude control to achieve more stable and advantageous tactical positions. The game objective is for the UAVs to attempt to outmaneuver or evade incoming attacks from each other while also trying to strike the opponent.

3.2 UAV control system theoretical analysis and modeling

Based on Figure 1, the dynamic game of fixed-wing UAVs can be decomposed into two function optimization processes. (1) Dynamic tracking of moving target points based on the rigid-body model. (2) Attitude stability and optimization of advantageous positions for the rigid-body model during close-range relative motion. In this paper, we design the mathematical model for trajectory optimization and attitude control of UAVs based on the above two points, as shown in Figure 2.

Based on the north-east-down (NED) coordinate system, OX points to the local true north, OY points to the local true east, OZ is oriented vertically downward, and the theoretical model for waypoint control is established in Figure 2(a). In the waypoint model, position(t) represents the position of the UAV at time t, v_t represents the current velocity vector, v_t' represents the projection of the velocity vector on the XOY plane, ϑ_{vc} represents the pitch angle of the velocity vector, and ψ_{vc} represents the yaw angle of the velocity vector. Figure 2(b) presents the theoretical model for UAV heading and attitude control. In the heading attitude model, position$^R(t)$ and position$^B(t)$ represent the current positions of the aircraft from both opposing sides, and v_t^R and v_t^B represent the velocities of the red and blue aircraft, respectively. v_t^{R-} represents the parallel projection of $v_t^{R\prime}$, which is used to explain the perpendicular relationship between the load factor and the velocity vector. The load factor commands in the y and z directions, denoted as N_{yc} and N_{zc}, respectively, are both perpendicular to the velocity vector v. $d(t)$ represents the relative line of sight distance between the two UAVs in the game.

Based on the decomposition of UAV control decisions shown in Figure 2, this paper proposes a design of an autonomous decision-making control system for UAVs based on a hybrid of heading attitude and trajectory. Specifically, the policy model generates heading attitude control commands or trajectory control commands based on the current environmental state. The output of the policy algorithm is transformed into load factor and roll angle information for the UAV through an inner-loop feedback controller. Under the 6-DOF model, this achieves decision control for both aircraft in adversarial states. The corresponding (1) aircraft modeling, (2) waypoint control system, and (3) heading attitude control system are designed as follows.

(1) Aircraft modeling. Define the body coordinate system as $OX_bY_bZ_b$, where OX_b points in the direction of the aircraft's nose, OZ_b lies within the longitudinal symmetry plane of the aircraft, pointing downward and perpendicular to OX_b, OY_b is perpendicular to OX_b, and OZ_b pointing to the right. Define the trajectory coordinate system as $OX_tY_tZ_t$, where OX_t points in the direction of velocity, OY_t lies in the horizontal plane, pointing to the right and perpendicular to OX_t, OZ_t is perpendicular to OX_t, and OY_t pointing downward. The translational motion kinematic equation for the aircraft is given by

$$\begin{pmatrix}\dot{x}\\ \dot{y}\\ \dot{z}\end{pmatrix}=\begin{pmatrix}c_\theta c_\psi & s_\gamma s_\theta c_\psi - c_\theta s_\psi & c_\gamma s_\theta c_\psi + s_\theta s_\psi\\ c_\theta s_\psi & s_\gamma s_\theta c_\psi + c_\theta c_\psi & c_\gamma s_\theta s_\psi - s_\theta c_\psi\\ -s_\theta & s_\gamma c_\theta & c_\gamma c_\theta\end{pmatrix}\begin{pmatrix}v_x\\ v_y\\ v_z\end{pmatrix}. \tag{1}$$

Define $c_x = \cos x$, $s_x = \sin x$. x, y, z represent the position coordinates of the drone in the NED coordinate system. v_x, v_y, v_z represent the components of the velocity vector $\boldsymbol{v}$ of the drone in the NED coordinate system along its three axes. γ, θ, ψ represent the roll angle, pitch angle, and yaw angle, respectively, of the drone's body relative to the NED coordinate system.

The rotational kinematic equation for the drone is given by

$$\begin{pmatrix}\dot{\gamma}\\ \dot{\theta}\\ \dot{\psi}\end{pmatrix}=\begin{pmatrix}1 & \sin\gamma\tan\theta & \cos\gamma\tan\theta\\ 0 & \cos\gamma & -\sin\gamma\\ 0 & \sin\gamma\sin\gamma\sin\theta & \cos\gamma\sin\theta\end{pmatrix}\begin{pmatrix}\omega_x\\ \omega_y\\ \omega_z\end{pmatrix}, \tag{2}$$

where $\omega_x, \omega_y, \omega_z$ are the components of the body angular velocity ω along the three axes of the body coordinate system.

The translational dynamic equation for the drone is given by

$$\begin{pmatrix}\dot{v}_x\\ \dot{v}_y\\ \dot{v}_z\end{pmatrix}=\begin{pmatrix}\omega_z v_y - \omega_y v_z\\ \omega_x v_z - \omega_z v_x\\ \omega_y v_x - \omega_x v_y\end{pmatrix}+\frac{1}{m}\begin{pmatrix}f_x\\ f_y\\ f_z\end{pmatrix}, \tag{3}$$

where m is the mass of the aircraft. f_x, f_y, f_z are the components of the total external force $\boldsymbol{F}$ acting on the drone along the three axes of the body coordinate system.

Define

$$\begin{cases}\Gamma_1 = \dfrac{J_{xz}(J_x - J_x + J_x)}{\Gamma},\\ \Gamma_2 = \dfrac{J_x(J_x - J_x) + J_{xz}^2}{\Gamma},\\ \Gamma_3 = \dfrac{J_x}{\Gamma},\\ \Gamma_4 = \dfrac{J_{xz}}{\Gamma},\\ \Gamma_5 = \dfrac{J_z - J_x}{J_y},\\ \Gamma_6 = \dfrac{J_{xz}}{J_y},\\ \Gamma_7 = \dfrac{(J_x - J_y)J_z + J_{xz}^2}{\Gamma},\\ \Gamma_8 = \dfrac{J_z}{\Gamma},\end{cases} \tag{4}$$

where $\Gamma \triangleq J_xJ_z - J_{xz}^2$. J_x, J_y, J_z represent the rotational inertias, and J_{xz} is the moment of inertia (due to the symmetry of the aircraft, J_{xy} and J_{yz} are nearly zero and thus ignored in the formula). Therefore, the rotational dynamics equations for the UAV are given by

$$\begin{pmatrix}\dot{\omega}_x\\ \dot{\omega}_y\\ \dot{\omega}_z\end{pmatrix}=\begin{pmatrix}\Gamma_1\omega_x\omega_y - \Gamma_2\omega_y\omega_z\\ \Gamma_s\omega_x\omega_z - \Gamma_6(\omega_x^2 - \omega_z^2)\\ \Gamma_\gamma\omega_x\omega_y - \Gamma_1\omega_y\omega_z\end{pmatrix}+\begin{pmatrix}\Gamma_3M_x + \Gamma_4M_z\\ \dfrac{1}{J_y}M_y\\ \Gamma_4M_x + \Gamma_8M_z\end{pmatrix}, \tag{5}$$

where M_x, M_y, M_z represent the components of the torque $\boldsymbol{M}$ acting on the drone along the three axes of the body coordinate system. The total external force $\boldsymbol{F}$ and total external torque $\boldsymbol{M}$ acting on the drone are functions of the drone's velocity $\boldsymbol{v}$, angular velocity ω, attitude angles γ, θ, ψ, elevator deflection angle δ_e, rudder deflection angle δ_r, and aileron deflection angle δ_a, defined as

$$\begin{cases}\boldsymbol{F} = \boldsymbol{F}(\boldsymbol{v}, \boldsymbol{\omega}, \gamma, \theta, \psi, \delta_e, \delta_r, \delta_a),\\ \boldsymbol{M} = \boldsymbol{M}(\boldsymbol{v}, \boldsymbol{\omega}, \gamma, \theta, \psi, \delta_e, \delta_r, \delta_a).\end{cases} \tag{6}$$

In (6), only three control surface deflection angles $\delta_e, \delta_r, \delta_a$ are adjustable. Therefore, the drone is controlled by modifying these three control surface deflection angles.

(2) Waypoint control logic. Based on the environmental information, the control decision model outputs discrete waypoints. The UAV model calculates the heading control command, speed bank angle command, and speed heading angle command based on the current position of the UAV, position$(t) = (x, y, z)$, and the desired waypoint, position$(t+1) = (x_{\mathrm{des}}, y_{\mathrm{des}}, z_{\mathrm{des}})$. The calculation methods for the speed bank angle command and speed heading angle command are given by

$$\begin{cases}\vartheta_{\mathrm{vc}} = \arctan \dfrac{-(z_{\mathrm{des}} - z)}{\sqrt{(x_{\mathrm{des}} - x)^2 + (y_{\mathrm{des}} - y)^2}},\\ \psi_{\mathrm{vc}} = \mathrm{atan2}\left((y_{\mathrm{des}} - y), (x_{\mathrm{des}} - x)\right),\end{cases} \tag{7}$$

where the function atan2$(\cdot)$ is defined as given in

$$\mathrm{atan2}(y, x) = \begin{cases}\arctan\left(\dfrac{y}{x}\right), & x > 0,\\ \arctan\left(\dfrac{y}{x}\right) + \pi, & y \geqslant 0, x < 0,\\ \arctan\left(\dfrac{y}{x}\right) - \pi, & y < 0, x < 0,\\ +\dfrac{\pi}{2}, & y > 0, x = 0,\\ -\dfrac{\pi}{2}, & y < 0, x = 0.\end{cases} \tag{8}$$

(3) Heading attitude control logic. The control decision model outputs continuous command control for heading attitude $(\psi_{\mathrm{vc}}, \vartheta_{\mathrm{vc}})$ and velocity parameters. The UAV model calculates the overload commands, denoted as n_{yc} and n_{zc} in the trajectory coordinate system based on the heading control parameters, as given in

$$\begin{cases}n_{yc} = \left[K_{\psi P}(\psi_{\mathrm{vc}} - \psi_v) + K_{\psi I} \displaystyle\int_0^t (\psi_{\mathrm{vc}} - \psi_v)\right] \cos \vartheta_v,\\ n_{zc} = K_{\theta P}(\vartheta_{\mathrm{vc}} - \vartheta_v) + K_{\theta I} \displaystyle\int_0^t (\vartheta_{\mathrm{vc}} - \vartheta_v),\end{cases} \tag{9}$$

where ϑ_v represents the actual velocity tilt angle, ψ_v represents the actual velocity yaw angle, t represents the current time, and $K_{\psi P}$, $K_{\psi I}$, $K_{\theta P}$ and $K_{\theta I}$ are adjustable parameters. The pitch channel overload command n_c and the roll angle command γ_c are calculated as

$$\begin{cases}n_c = \mathrm{sign}(n_{zc})\sqrt{n_{yc}^2 + n_{zc}^2},\\ \gamma_c = \arctan \dfrac{n_{yc}}{-n_{zc}},\end{cases} \tag{10}$$

where the function sign$(\cdot)$ represents the sign function, which is defined as

$$\mathrm{sign}(x) = \begin{cases}1, & x > 0,\\ 0, & x = 0,\\ -1, & x < 0.\end{cases} \tag{11}$$

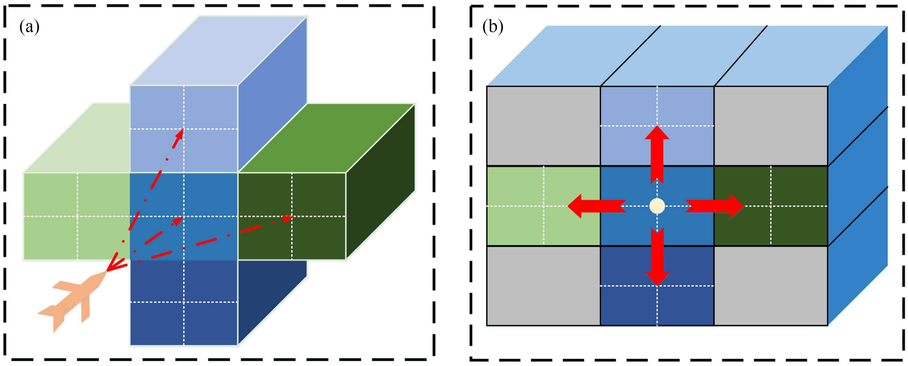

Figure 3 (Color online) Illustration of 3D spatial mesh discretization and action selection. (a) Spatial discretization modeling; (b) discrete action selection.

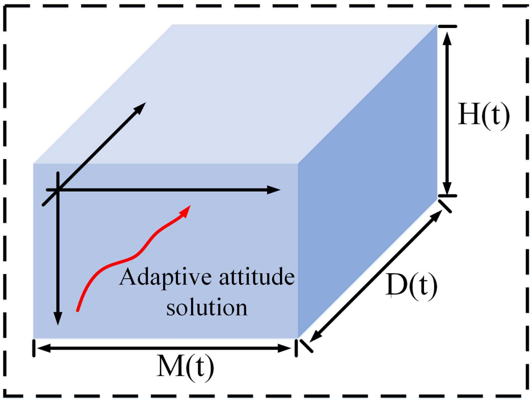

Figure 4 (Color online) Illustration of the size of the grid space and the internal continuous action.

4 Dynamic game control method based on hybrid action reinforcement learning

4.1 Adaptive discretization method for variable-scale space

In the research on unmanned platform games, conventional approaches for generating UAV trajectories often rely on heuristic algorithms to determine waypoints. However, these methods suffer from inefficiency due to the large search space involved. UGVs (unmanned ground vehicles) often employ grid-based search methods for trajectory exploration. The limitation of such approaches is that they require pre-computation of the grid cells by performing point cloud calculations and constructing the grid structure, in many UAV game scenarios, these preconditions do not exist. To enhance the efficiency of trajectory control strategies for fixed-wing UAVs without compromising search accuracy, this paper proposes a variable-scale trajectory region grid-based method. The principle is based on constructing a spatial grid of variable scales along the feasible exploration path for the UAV's next step, considering its current state. A random point is selected from the grid as the target waypoint for the UAV in the next time step. This method avoids the need for global spatial grid discretization and enables the calculation of trajectory points based on the distance between grid cells. As a result, it reduces the computational time and spatial resources required for calculating the trajectory points in the control algorithm.

As shown in Figure 3, the UAV constructs a spatial grid for the next time step based on its velocity and position information at the current time. The size of the grid is illustrated in Figure 4, where $M(t) = v_y^{t-1} \times t$, $D(t) = v_x^{t-1} \times t$, $H(t) = v_z^{t-1} \times t$, and v_x, v_y, v_z represent the velocity of the UAV.

The adaptive-scale grid is then mapped onto a two-dimensional plane, as shown in Figure 5. At time t, the UAV has a velocity of v^t, and the spatial grid size at this time is denoted as $\mathrm{Space}(t)$. Assuming the UAV is located in the yellow grid in Figure 5(a), the possible positions at time $t+1$ are represented by the red grid. If the grid scale remains unchanged, the possible positions at time $t+2$ would be indicated by the blue grid in Figure 5(a). However, when utilizing the variable-scale grid method, as shown in

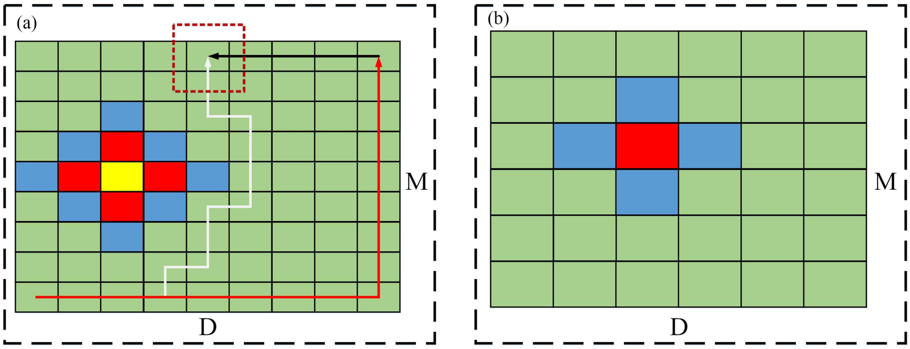

Figure 5 (Color online) 2D spatial mapping representation of the adaptive grid. (a) Discrete actions in the 2D plane; (b) adaptive variation of the grid in the 2D plane.

Figure 5(b), at time $t+1$, the grid form changes due to the altered velocity of the UAV. Consequently, the range of the grid at time $t+2$ becomes the blue area in Figure 5(b).

4.2 Design of discrete and continuous action spaces based on variable-scale space

Based on the grid-based design method and the definition of Markov processes, this paper defines the discrete action output as the selection of the next grid cell in the three-dimensional space. The selection of each grid cell follows a policy denoted as $\pi(a|s)$, as illustrated in Figure 3(b). Due to the constraints on the UAV's speed inclination and yaw angle, the discrete action selection is limited to nine grid intervals in the direction of the UAV's velocity. The center grid is labeled as a_5, the right side as a_1, the top side as a_2, the left side as a_3, and the bottom side as a_4. The gray area has a selection probability of 0. The policy $\pi(a|s)$ is computed using the SoftMax function, and the discrete action space is defined as

$$\text{action}_{\text{discrete}}[] = \pi(a|s) = [a_1, a_2, a_3, a_4, a_5]. \tag{12}$$

The trajectory control policy model computes the next trajectory point after determining the grid space through the algorithm. Taking the current time t as an example for analysis, the size range of the current space is $\text{Space}(t) = (v_x^{t-1}, v_y^{t-1}, v_z^{t-1})$. Assuming the current position is $\text{position}(t) = (x_t, y_t, z_t)$, the next time step is $\text{position}(t+1) = (x_{t+1}, y_{t+1}, z_{t+1})$. To ensure the randomness of discretely selecting points in the grid space, the next trajectory point follows a multi-modal Gaussian distribution centered around the grid's central point $\text{grid}(t) = (x'_t, y'_t, z'_t)$. The coordinates of the grid center points between two consecutive policies are related according to (13)–(15).

$$x'_{t+1} = x'_t + \frac{1}{2}v_x^t + \frac{1}{2}v_x^{t+1}, \quad a = a_i,\ i \in [1,5], \tag{13}$$

$$y'_{t+1} = \begin{cases} y'_t, & a = a_2, a_4, a_5, \\ y'_t + \frac{1}{2}v_y^t + \frac{1}{2}v_y^{t+1}, & a = a_1, \\ y'_t + \frac{1}{2}(-v_y^t) + \frac{1}{2}(-v_y^{t+1}), & a = a_3, \end{cases} \tag{14}$$

$$z'_{t+1} = \begin{cases} z'_t, & a = a_1, a_3, a_5, \\ z'_t + \frac{1}{2}v_z^t + \frac{1}{2}v_z^{t+1}, & a = a_2, \\ z'_t + \frac{1}{2}(-v_z^t) + \frac{1}{2}(-v_z^{t+1}), & a = a_4, \end{cases} \tag{15}$$

where $v^{t+1} = v^t + \Delta v$.

This discrete action selection process allows the UAV to navigate through the variable-scale space effectively while maintaining trajectory accuracy. Building upon this foundation, the control system designed in this paper incorporates continuous actions to facilitate fine-grained adjustments during the flight process. The continuous action space includes parameters such as tilt angle and yaw angle deviation of the velocity. The continuous action space, represented by $\text{action}_{\text{continuous}}[]$ in (16), is utilized to perform

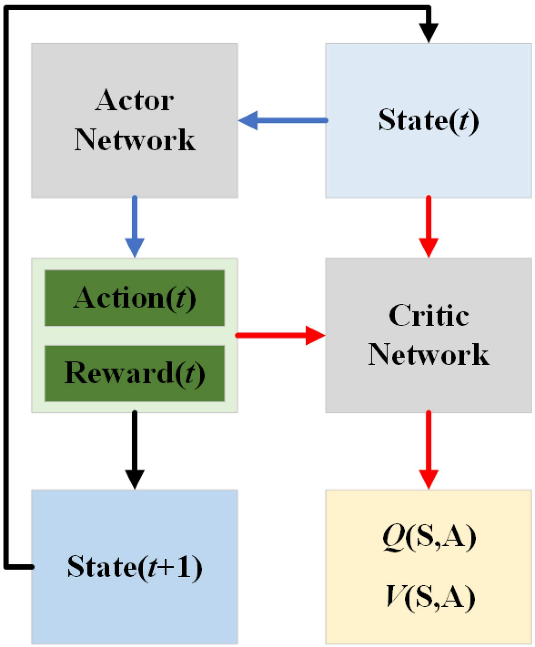

Figure 6 (Color online) AC Framework.

more precise attitude adjustments for the UAV, enabling it to maneuver within the opponent's attack line of sight or evade the opponent's lock-on.

$$\text{action}_{\text{continuous}}[] = \mu(s) = (\vartheta_{\text{vc}}, \psi_{\text{vc}}). \tag{16}$$

4.3 Dynamic control method based on hybrid action PPO

4.3.1 *Dynamic control framework modeling based on AC framework*

In this paper, we propose an improved PPO method based on hybrid action spaces to address the challenges of combining stochastic discrete actions and deterministic continuous actions in dynamic control of UAV. The basic algorithm framework is illustrated in Figure 6.

Figure 6 can be referred to as the QAC method, where the process indicated by the red arrows is also known as the value function approximation process, which is used to evaluate the current policy. The process indicated by the blue arrows is referred to as the policy gradient (PG) process, which is used to update the current policy. The action value $Q(\boldsymbol{S}, \boldsymbol{A})$ is represented as

$$Q_\pi(\boldsymbol{S}, \boldsymbol{A}) \triangleq q_{\alpha\in\pi\|\mu}(\boldsymbol{S} = s_t, \boldsymbol{A} = a_t, \boldsymbol{R} = r_{t+1}), \tag{17}$$

where $q_\pi(s,a) = \mathbb{E}_{\tau\sim P(s'|s,a)}[G_t|S_t = s, A_t = a]$. The discounted return G_t is calculated based on the reward function $\boldsymbol{R}$ in the Markov decision process $\text{MDP}(\boldsymbol{S}, \boldsymbol{A}, \boldsymbol{R}) = [s_t, a_t, r_{t+1}, s_{t+1}, \ldots]$, as shown in

$$\begin{aligned} G_t &= R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots \\ &= \sum_{i=0}^{N} \gamma^i R_{t+i+1}. \end{aligned} \tag{18}$$

The reward function $\boldsymbol{R}$ is the sum of individual rewards. According to the AC framework, the dynamic control method for fixed-wing UAV is defined in Algorithm 1.

4.3.2 *Reinforcement learning algorithm design based on hybrid action space*

In this paper, we propose a hybrid action space algorithm (hybrid-action-PPO) for dynamic control of fixed-wing UAV. The algorithm follows the AC framework and utilizes the PPO algorithm as the base. It integrates the advantages of both discrete and continuous actions to achieve flexible and precise control of the UAV. On the basis of the AC framework, the PPO algorithm improves the form of the objective

Algorithm 1 Actor-critic reinforcement learning.

```
Define: Episode_max = $N$, Step_max = $M$;
Initialize: critic network parameters $w$, actor network parameters $\theta$;
for each episode do
    Reset the environment and obtain the initial state $s_t$;
    for each time step $t$ do
        Select an action $a_t$ from $\pi(a_t|s_t,\theta_t)$, get $r_{t+1}$, enter $s_{t+1}$, and then choose $a_{t+1}$;
        Update the critic network by minimizing the mean squared TD error:
        $w_{t+1} = w_t + \alpha_w[r_{t+1} + \gamma q\,(s_{t+1}, a_{t+1}, w_t) - q(s_t, a_t, w_t)]\nabla_w q\,(s_t, a_t, w_t)$;
        Update the actor network using the sampled policy gradient:
        $\theta_{t+1} = \theta_t + \alpha_\theta \nabla_\theta \text{In}\pi\,(a_t|s_t,\theta_t)\,q\,(s_t, a_t, w_{t+1})$;
    end for
end for
```

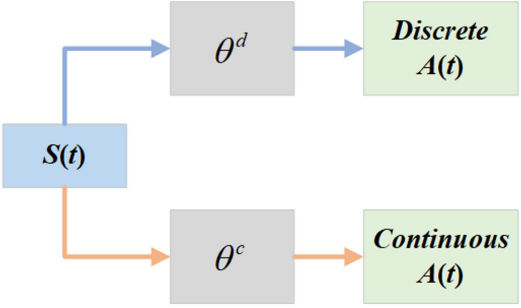

Figure 7 (Color online) Neural network structure of the hybrid action.

function and the gradient descent update process. The objective function of the PPO method is defined as follows:

$$\begin{cases} \mathcal{L}^{\text{clip}}_{\pi_{\theta_k}}(\pi_{\theta_k}) = \mathbb{E}_{\tau\sim\pi\theta}\left[\sum_{t=0}^{T}[\min J(\theta)]\right], \\ J(\theta) = \left(\rho_t(\pi_\theta,\pi_{\theta_k})A_t^{\pi_{\theta_k}}, \text{clip}(\rho_t(\pi_\theta,\pi_{\theta_k}),1-\epsilon,1+\epsilon)A_t^{\pi_{\theta_k}}\right), \end{cases} \tag{19}$$

where $L^{\text{clip}}_{\pi_{\theta_k}}(\pi_{\theta_k})$ represents the PPO objective function with respect to the policy parameters θ, $A_t^{\pi_{\theta_k}}$ is the advantage function, to evaluate the appropriateness of selecting a specific action a in a given state.

$$\begin{cases} A_t^{\pi_{\theta_k}}(S=s_t, A=a_t) = Q_\pi(S,A) - V_\pi(S), \\ V_\pi(S) = E_\pi[G_t|S=s_t, A=a_t] = \sum_{a_t}\pi(A=a_t|S=s_t)Q_\pi(S,A). \end{cases} \tag{20}$$

$\rho_t(\pi_\theta,\pi_{\theta_k})$ denotes the probability ratio between the new and old policies, defined as

$$\rho_t(\pi_\theta,\pi_{\theta_k}) = \frac{\pi_\theta}{\pi_{\theta_k}}. \tag{21}$$

The variable ϵ is a hyperparameter used to control the magnitude of clipping during function updates.

$\text{clip}(\rho_t(\theta), 1-\epsilon, 1+\epsilon)$ is a mathematical function that restricts the value x to be within the interval $[a,b]$.

$$\text{clip}(x,a,b) = \begin{cases} x\cdots a < x < b, \\ a\cdots x \leqslant a, \\ b\cdots x \geqslant b. \end{cases} \tag{22}$$

In our hybrid-action PPO method, the improved action function is defined as

$$\begin{cases} a^d = \pi(a|s,\theta^d), \\ a^c = \mu(s,\theta^c). \end{cases} \tag{23}$$

The neural network design for the corresponding action function, where a^c represents the deterministic continuous action and a^d represents the stochastic discrete action, is shown in Figure 7.

Additionally, in order to explore the distribution space of continuous actions more effectively, this paper replaces the truncated Gaussian distribution with a Beta distribution on the output of the actor network. In order to improve the sample reuse rate of importance sampling, the form of $\rho_t(\theta)$ is modified as follows in (24), where θ_{old} represents the parameters from the past K adversarial sample training iterations.

$$\rho_t(\theta) \triangleq \left[\omega_1 \frac{\pi_\theta}{\pi_{\theta_{\text{old}}}} + \omega_2 \frac{\mu_\theta}{\mu_{\theta_{\text{old}}}}\right]. \tag{24}$$

The computation of action value Q incorporates a weighted sum of discrete action probabilities $\pi(s)$ and continuous action $\mu(s)$ to calculate the $\text{TD} - \text{Error} = \delta_t$, as shown in

$$\begin{gathered} \delta_t = r_{t+1} + \omega_1 \delta_t^1 + \omega_2 \delta_t^2, \\ \begin{cases} \delta_t^1 = \gamma_1 q(s_{t+1}, \pi(a_{t+1}|s_{t+1}, \theta^d), w_t) - q(s_t, \pi(a_t|s_t, \theta^d), w_t), \\ \delta_t^2 = \gamma_2 q(s_{t+1}, \mu(s_{t+1}, \theta^c), w_t) - q(s_t, \mu(s_t, \theta^c), w_t). \end{cases} \end{gathered} \tag{25}$$

As a result, the parameter update for the actor network is transformed into

$$\begin{cases} \theta_{t+1}^d = \theta_t^d + \alpha_\theta \nabla_{\theta\,\mathrm{In}} \pi(a_t|s_t, \theta_t^d) q(s_t, a_t, w_{t+1}), \\ \theta_{t+1}^c = \theta_t^c + \alpha_\theta \nabla_\theta \mu(s_t, \theta_t^c) \big(\nabla_a q(s_t, a_t, w_{t+1})\big)\big|_{a=\mu(s_t)}, \end{cases} \tag{26}$$

where the weights ω_1 or ω_2 are obtained from a binary classifier, and the input to the classifier is the environmental observation. The relationship of the weights is described in

$$\omega_1 + \omega_2 = 1, \omega_1 = 1 \text{ or } 0. \tag{27}$$

4.3.3 *Game-theoretic decision model for the UAV via the hybrid PPO method*

On the basis of the hybrid action strategy gradient, this paper designs a dynamic control decision model for the UAV in dynamic game scenarios. Following the definition of reinforcement learning, and the UAV game scenario, the dynamic control decision model for UAV in this paper consists of three components: the state space, action space, and reward function.

(1) The environmental state observation(t) of the UAV is defined as

$$\text{observation}\,(t) = [\text{position}\,(t)\,, v\,(t)\,, \phi\,(t)\,, \varphi\,(t)\,, \text{distance}\,(t)]\,, \tag{28}$$

where position$(t) = (x_t, y_t, z_t)$ is consistent with the previous context and represents the spatial position in the NED coordinate system. $v(t) = (v_x^t, v_y^t, v_z^t)$ represents the velocity information in the NED coordinate system. $\phi(t) = (\phi_{\text{pitch}}^t, \phi_{\text{yaw}}^t, \phi_{\text{roll}}^t)$ refers to the UAV's body attitude information in spatial space, and the attitude information constrains the range of actions that the decision-making model can output. $\varphi(t) = (\psi_{\text{vc}}^t, \vartheta_{\text{vc}}^t)$ represents the heading attitude information. In the discrete action network, it is automatically computed using (1). In the continuous action network, it is obtained through the policy algorithm using the variable $\mu(s, \theta)$. distance(t) represents the Euclidean distance between UAVs.

(2) The action space for dynamic control of the UAV is defined to be consistent with (23). The decision-making process is influenced by the state space observation(t). When the relative distance(t) is large, the decision-making model outputs random discrete actions $a = \pi(a|s) = (a_i)$. When the relative distance(t) is small, the decision-making model outputs deterministic continuous actions $a = \mu(s) = (\psi, \vartheta, v)$. To ensure the stability of the UAV, the values of continuous actions should not change dramatically. Therefore, for each UAV, the output of the decision-making model represents a small incremental change. After being added to the current heading attitude angle, the actions are controlled based on (9) in terms of the absolute values of the attitude angles. Thus, the continuous output of the decision-making model is represented as a tuple $a = (\Delta\psi, \Delta\vartheta, \Delta v)$.

(3) Considering the game objectives of both sides, the reward functions for the two players are defined in Algorithm 2. The reward functions for both players satisfy the requirements of zero-sum game and guide the UAV to achieve a Nash equilibrium in both continuous and discrete action control. Reward.red and Reward.blue represent the reward value for the two players in the game. Since the rewards are zero-sum, this paper only shows the calculation of rewards for red's attack posture.

The determination of the termination state for a complete adversarial round is depicted in Figure 8. In Figure 8, d represents the line-of-sight distance between the UAVs, d.attack represents the maximum attack distance, and φ_β^R and φ_β^B represent the angle of view of the game constituted by the UAV and

Algorithm 2 UAV action reward function.

```
Initialize: Reward.red = Reward.blue = 0, $d(t)$ = distance($t$), $d$.attack = $\varepsilon$, blue.health = $K$.
if $d(t) > 50$ and blue.health$(t) > 0$ then
    if $d(t) < d(t-1)$ then
        Reward.red $+= ((d(t-1) - d(t))/d(t-1))$;
        Reward.blue $+= ((d(t-1) - d(t))/d(t-1))$;
    end if
    if $d(t) < \varepsilon$ then
        if $\varphi_\beta^R < \varphi_\alpha^R$ and $\varphi_\beta^B > 90^\circ$ then
            Reward.red $+= 100$;
            Reward.blue $-= 100$;
            blue.health$(t) = (K-1) * p$;
        end if
    end if
end if
```

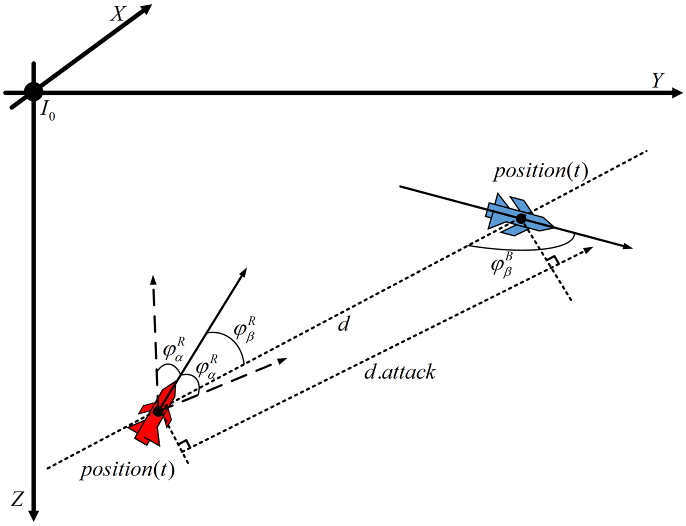

Figure 8 (Color online) Attitude determination of UAV close-range gaming.

Table 2 Relationship between situational information and relative line of sight angle in the close-range combat.

Angle of view (°)	$\varphi_\beta^R < \varphi_\alpha^R$	$\varphi_\alpha^R < \varphi_\beta^R < 90^\circ$	$90^\circ < \varphi_\beta^R < 180^\circ$
$\varphi_\beta^B < \varphi_\alpha^B$	$(\text{red} = 0, \text{blue} = 0)$	$(\text{red} = 0, \text{blue} = 1)$	$(\text{red} = -1, \text{blue} = 2)$
$\varphi_\alpha^B < \varphi_\beta^B < 90^\circ$	$(\text{red} = 1, \text{blue} = 0)$	$(\text{red} = 1, \text{blue} = 1)$	$(\text{red} = -1, \text{blue} = 1)$
$90^\circ < \varphi_\beta^B < 180^\circ$	$(\text{red} = 2, \text{blue} = -1)$	$(\text{red} = 1, \text{blue} = -1)$	$(\text{red} = 0, \text{blue} = 0)$

the relative line of sight (the angle between the line connecting the center of mass of the aircraft and the positive direction of the aircraft's body axis), respectively. Superscript R for red, B for blue, φ_α^R stands for maximum attack sight angle, when $\varphi_\beta^R < \varphi_\alpha^R$ and $\varphi_\beta^B > 90^\circ$, the red UAV can lock onto the tail of the blue UAV, causing blue to lose blue.heath with a certain probability P. Based on Figure 8, we can define the situation in the aircraft game process as absolute advantage (indicated by 2), advantageous position (indicated by 1), disadvantageous position (indicated by 0), and absolute disadvantage (indicated by −1), as shown in Table 2. In Table 2, the vector (red, blue) represents the situational information for the red and blue UAV, where red corresponds to the red side, and blue corresponds to the blue side.

Based on the design of various parameter spaces in this section, we present the training of the fixed-wing UAV game decision model as shown in Figure 9.

Figure 9 can be divided into three main modules. (1) Environment update module: the UAV model updates the state information in the game space based on the control parameters, such as waypoints or heading attitudes. (2) Hybrid-action-RL network: based on the state space information of the UAV, this module generates a grid-based space with adaptive velocity and generates discrete control actions for waypoint navigation. In the case of close-range game states, it directly generates continuous control actions for heading attitudes. (3) Reward and attitude calculation module: this module calculates the reward values and the $Q(\boldsymbol{S}, \boldsymbol{A})$ value based on the reward function and the UAV's actions. It evaluates and updates the policy function accordingly. Additionally, it changes the state information of the UAV

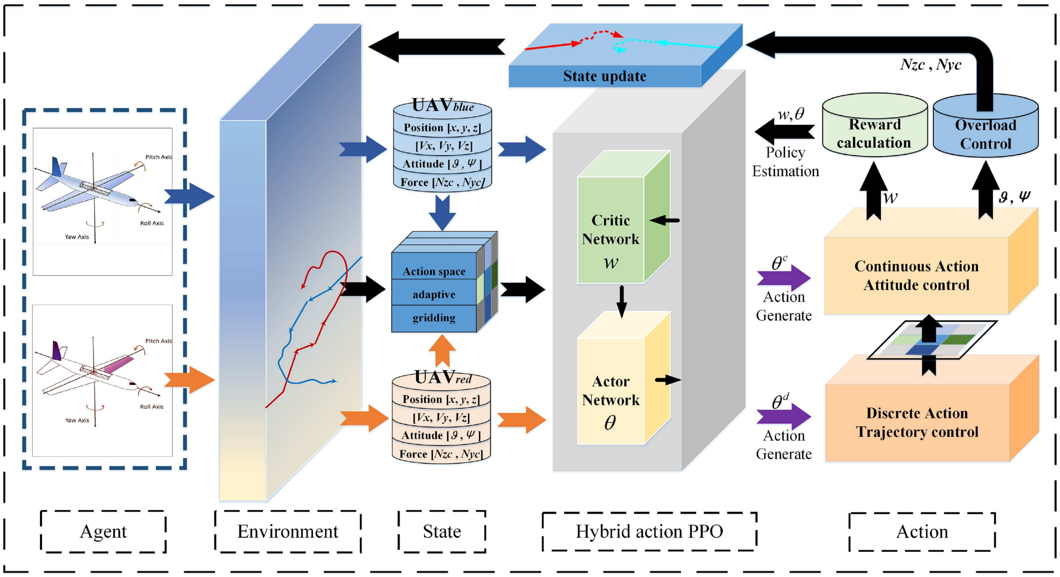

Figure 9 (Color online) Dynamic control flow chart based on hybrid-action-PPO method for fixed-wing UAV gaming.

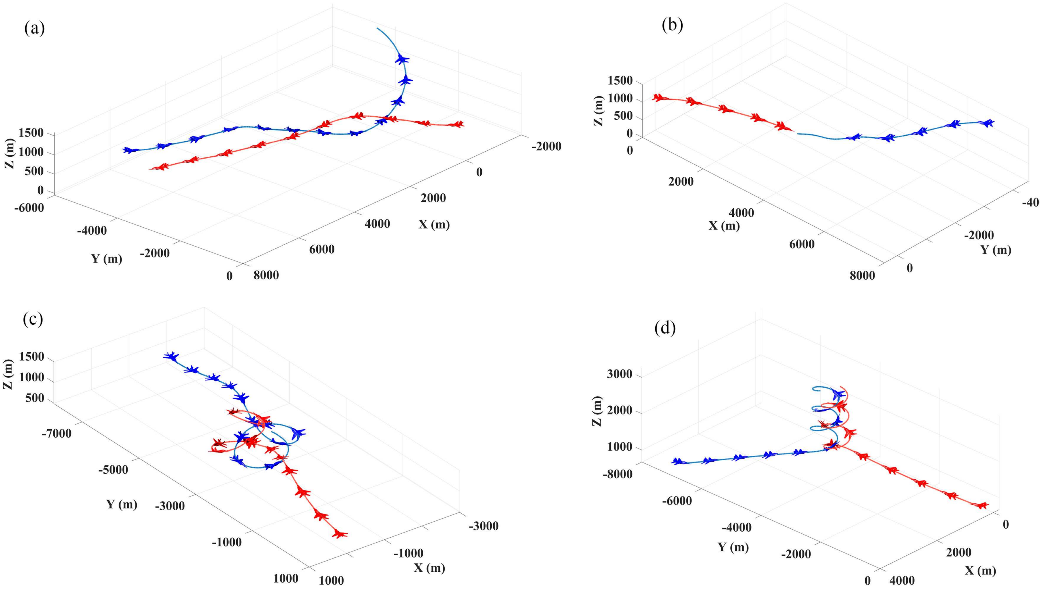

Figure 10 (Color online) UAV control of hybrid-action-PPO dynamic game decision-making in symmetric environments. (a) Space and action exploration; (b) medium and long-range waypoint guidance; (c) close-range attitude control; (d) game equilibrium situation.

based on the control system.

5 Experiment and analysis

This paper introduces a hybrid-action-RL framework for dynamic UAV games, simulating typical pursuit-evasion adversarial behaviors in a 1v1 game scenario. During the game, there are symmetric and asymmetric UAV state parameters. As a result, in the dynamic game with the red UAV as the main subjective player, there are three possible outcomes: advantage, equilibrium, and disadvantage. The simulation training environment is Python3.8, GPU Nvidia GeForce RTX3070, RAM 32G. The neural network is a three-layer fully connected network with the number of neurons 64, 128, and 32, respectively. The simulation results of the dynamic game for fixed-wing UAVs are shown in Figure 10.

Figure 10 demonstrates the learning effect of the UAV game decision model in a symmetric environment,

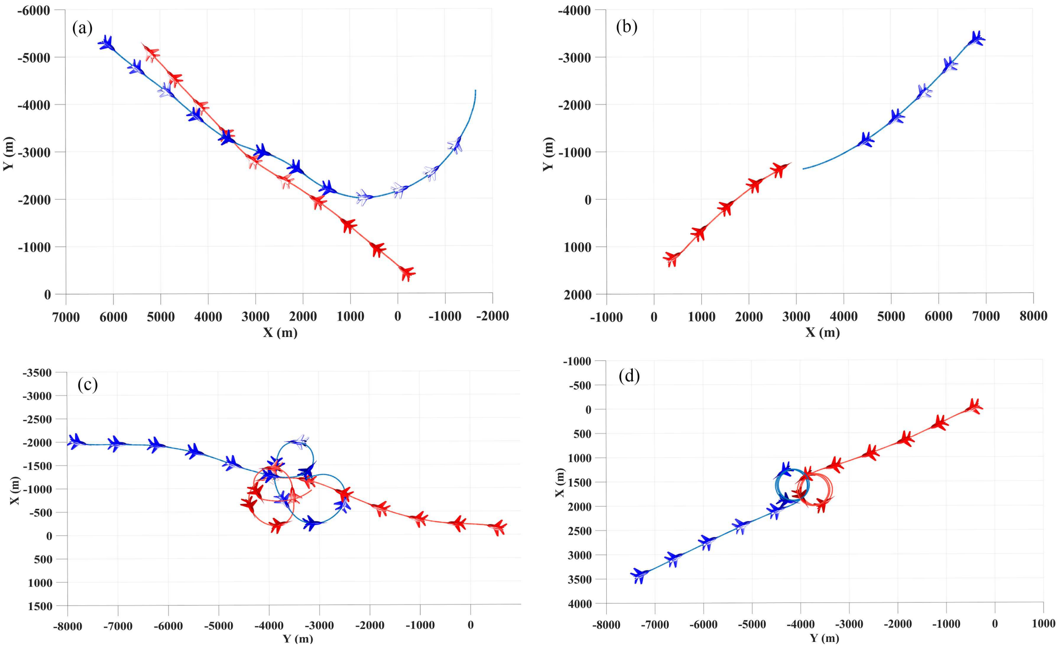

Figure 11 (Color online) 2D displays of the aircraft dynamic game decision control effect. (a) Space and action exploration; (b) long-range waypoint guidance; (c) close-range attitude control; (d) game equilibrium situation.

where Figure 10(a) shows the UAV exploring the environment at the initial stage, with the maneuvering range covering the overall game space of 8 km × 6 km. Figures 10(b) and (c) show that the hybrid-action-PPO algorithm guides the UAV to dynamically track the target through discrete waypoint control during the far-range gaming process, with the search space on the yaw plane gradually shrinking, and performs continuous attitude changes after entering the attack distance. Figure 10(d) shows the convergence of the algorithms to reach the game homogeneous situation due to the same maneuvering capability of the UAVs and the same intention, at which time the range on the UAV's yaw plane is narrowed down to less than 4000 m.

In conjunction with the situational definitions in Table 2, it can be observed that in the early stages of learning the decision model, both aircraft generally maintain a relatively balanced situation. The vectors representing the red-blue confrontation situation mostly appear in a state of $(0, 0)$. As the red side optimizes its decisions in close-range combat, the situational vectors transition from a balanced state $(0, 0)$ to an advantageous position $(1, 0)$, $(1, -1)$, or an absolute advantage $(2, -1)$. Subsequently, the decision performance of the blue side evolves, and the situational vectors continuously engage in a game between $(2, -1)$, $(1, -1)$, and $(-1, 1)$, $(-1, 2)$ until both sides reach a balanced state of $(0, 0)$ to $(1, 1)$ when the decision model converges. Figure 10 clearly demonstrates the discrete trajectory point control strategy and the continuous heading attitude control strategy when the UAV is dynamically gaming in 3D space, and a more intuitive 2D depiction is shown in Figure 11.

Figure 11 shows that the decision algorithm model is biased towards the exploration of maneuver control strategies in the early stages of training, with a tendency to use a large velocity declination for the UAV's maneuver control. With the training of hybrid-action-PPO, the dynamic tracking capability of the strategy algorithm against the enemy target is rapidly enhanced in Figure 11(b). In Figure 11(c), the UAV close-range gaming quickly adjusts the velocity declination and velocity inclination to lock on the enemy tail once it occupies a favorable position. After the model is gradually stabilized, due to the two sides' maneuvering ability is the same, the two sides tend to adopt mutual circling after entering the close-range game state, which is aimed at pursuing locking on the enemy while avoiding being attacked by the enemy in Figure 11(d).

The convergence effect of hybrid-action-PPO in the symmetric environment is shown in Figure 12(a). Figure 12 demonstrates the variation of the reward values of the UAV controlled by the hybrid-action-PPO strategy under different game scenarios. The advantageous situation is defined as when UAV locks on to the other at attack range. Due to the iterative game approach used in this paper's method for training, the design of blue UAV strategy parameters is based on the previous moment's red UAV strategy model.

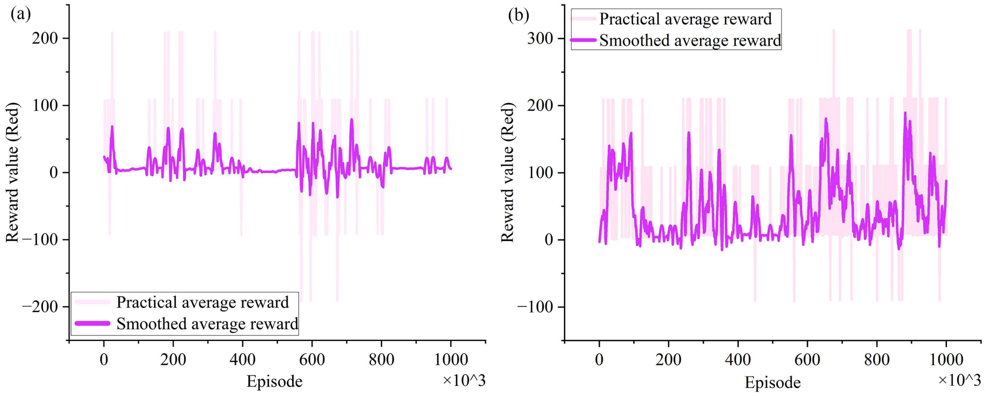

Figure 12 (Color online) Convergence values of the reward function of hybrid-action-PPO. The maximum step per episode is 500. (a) Symmetric games, where both the red and blue's attack lock line-of-sight angles are set to 30°; (b) asymmetric games, where the red's locking line-of-sight angle is 70° while the blue's is unchanged.

Table 3 Red UAV gains a tactical advantage situation with different game view angles.

Line-of-sight angles (°)	Advantage situation probability	Multiple advantage situation probability	Reward (mean ± std)
red = 20, blue = 30	0.038	0.003	8.839 ± 25.411
red = 30, blue = 30	0.064	0.014	11.348 ± 36.803
red = 50, blue = 30	0.115	0.055	22.749 ± 51.259
red = 70, blue = 30	0.292	0.099	46.003 ± 70.048
red = 60, blue = 40	0.130	0.016	21.944 ± 43.437
red = 70, blue = 40	0.187	0.085	33.189 ± 62.576

Consequently, in Figure 12(a), it can be observed that in the early stages of strategy model training, under symmetric conditions, the probability of the red team entering the advantageous situation is greater than that of the blue team. However, after the strategy models converge, the gradients of both the red and blue teams' strategy models stabilize, resulting in both teams' probabilities of entering the advantageous situation becoming nearly equal.

In conjunction with Algorithm 2, from the perspective of the red aircraft, the maximum reward is obtained only when the red side has an absolute advantage and the blue side is in an absolute disadvantage situation. In Figure 12(a), it can be observed that the red UAV enters the absolute advantageous $(2, -1)$ situation at most 2 times in a single game round with a limited number of actions. On the other hand, in Figure 12(b), there are 3 occurrences of the absolute advantageous $(2, -1)$ situation, indicating that the UAV's locking posture directly affects the change in the game situation. This also suggests that the strategy model can quickly learn the correlation between UAV performance and game situation in different game environments, enabling UAVs with better maneuverability to achieve more game victories. Moreover, in Figure 12(b), even when the red UAV has significantly better maneuver locking capability than the blue team, the blue UAV can still gain advantages in the game, demonstrating that hybrid-action-PPO possesses strong autonomous decision-making ability even in disadvantageous environments.

This paper conducted comparative experiments for different locking posture angle parameters, with a total of 80000 episodes in a single scenario. The results are shown in Table 3. Furthermore, in Figure 13, we compared the convergence of the algorithm under different aircraft performance parameters, where the threshold for the attackable line-of-sight angle for the blue aircraft is consistently set to 30°.

The policy loss in Figure 13 exhibits some fluctuations, attributed to the algorithm simultaneously computing control decisions for both adversarial parties. Consequently, compared to conventional single-aircraft control methods, the curve appears less smooth. However, analyzing its trend, when the performance parameters of the red aircraft are lower than those of the blue aircraft, the maximum value of the strategy loss occurs in the mid-training phase. This suggests that in the early stages of the algorithm, the red aircraft gains fewer advantageous situations in control decisions, and the blue aircraft dominates more often. Therefore, the red aircraft needs to explore the strategy space more extensively, leading to a larger increment in strategy loss. Once the red aircraft's strategy converges, the loss gradually decreases towards Nash equilibrium. As the attack line of sight angle threshold increases for the red aircraft, its strategy model's loss function exhibits a similar converging trend. Notably, the similarity between Fig-

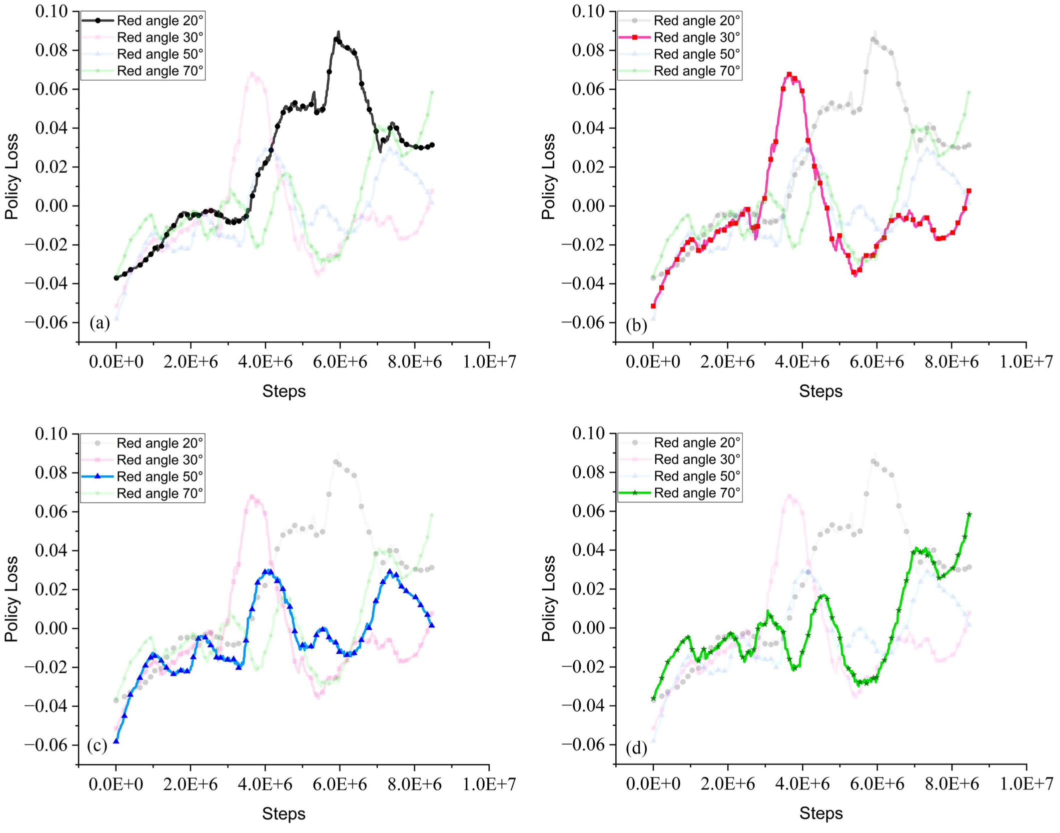

Figure 13 (Color online) Convergence values of the policy loss of hybrid-action-PPO, where the attackable line-of-sight angles are (a) 20°, (b) 30°, (c) 50°, and (d) 70°, respectively.

ures 13(c) and (d) is high, contrasting with the fluctuation in Figure 13(a). With an increase in training iterations, the algorithm shows earlier exploratory behavior, aiming to obtain possibilities of situational changes under disadvantaged confrontations. This indicates that under different performance parameters, hybrid-action-PPO consistently achieves good control results and maintains stable convergence. In symmetric situations, due to the asynchronous time difference in the decision algorithms employed by the red and blue aircraft, the red aircraft explores a larger space in the early stages, leading to a faster strategy search speed. The algorithm exhibits the best convergence stability. As the training iterations gradually saturate, the decision situations of the red and blue aircraft maintain stable equilibrium, showing similar fluctuation patterns to the reward function in Figure 12(a).

From Table 3, it is evident that the UAVs controlled by the hybrid-action-PPO method exhibit a positive correlation between locking posture and game advantage in 1v1 scenarios. Under larger locking angles, the UAVs can enter the advantageous state multiple times. In a symmetrical environment, the algorithm's reward values converge to around 11, and the game situation stabilizes into a balanced state of close-range circling between the UAVs. Consistent with Figure 11(d), when the red UAV's attack locking range exceeds that of the blue UAV, the algorithm's convergence value exceeds 20, indicating a significant enhancement in the drone's autonomous decision-making capability. In terms of trajectories, this is manifested by the drone being more likely to occupy advantageous positions, as shown in Figure 13. From the comparison of different line-of-sight angle parameters, it can be observed that as the angular difference between the two UAVs decreases, the decision-making model is more likely to enter an equilibrium state. Under the same line-of-sight angle difference, when the line-of-sight angle parameter of the blue UAV is larger, the red UAV experiences fewer instances of advantageous situations. This indicates that the decision-making model can quickly adjust the UAV's control strategy upon sensing the disadvantage of the blue UAV, reducing the probability of being pursued by the red UAV.

In Figure 14(a), the enlarged view in the top right corner illustrates the drone's attitude in a close-range scenario, with the red and blue arrows representing their current velocity directions. At this moment, the red UAV occupies a more advantageous pursuit position based on the control strategy, achieving a

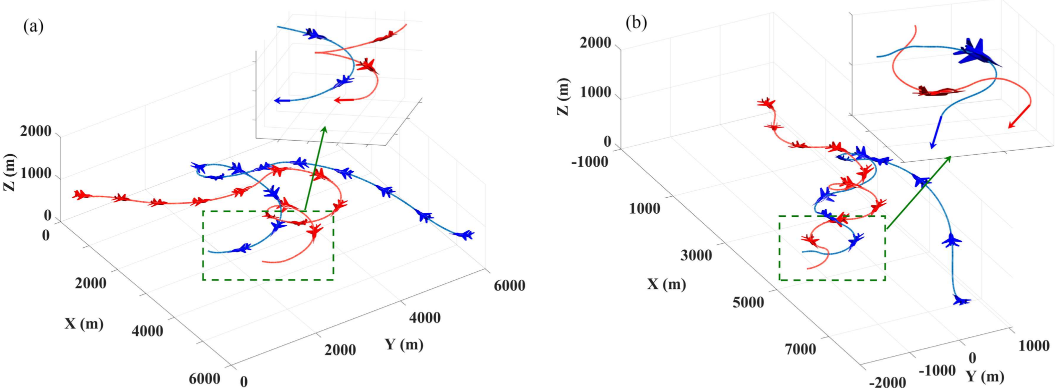

Figure 14 (Color online) Occupying a favorable position during close-range gaming. (a) Advantageous tail-chase position (red maneuvering to the tail of blue); (b) advantageous dive-chase position (red locks onto blue from above on a diagonal).

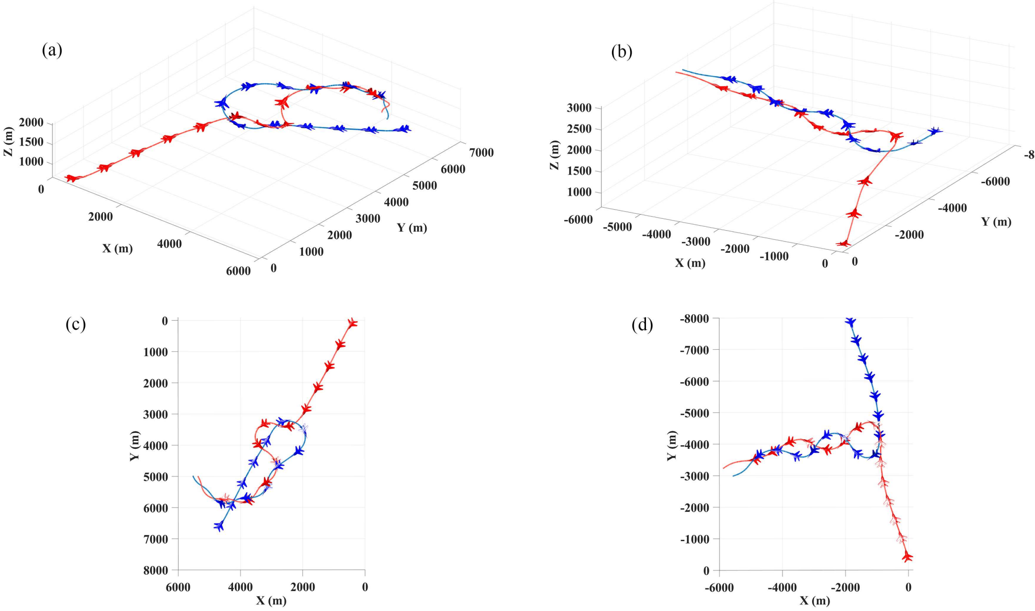

Figure 15 (Color online) Close-range dogfighting with small-angle maneuvers. (a) Asymmetric strategy model's parameters; (b) asymmetric attitude parameters; (c) 2D display of (a); (d) 2D display of (b).

tail-lock on the blue UAV. Analyzing the situation, the red side has an absolute advantage, while the blue side is at an absolute disadvantage, resulting in a situation vector of $(2,-1)$. This indicates a scenario where the red side is pursuing the blue side attempting to escape. Figure 14(b) shows the red UAV occupying a favorable position for a diving attack, the blue side is at a positional disadvantage, while the red side has a positional advantage with a situation vector of $(1,0)$, transitioning towards the $(1,-1)$ situation.

In Figure 15, the blue UAV, despite its maneuverability disadvantage, can still secure advantageous positions during the convergence of the strategy algorithm. This forces the opponent's UAV into a small-angle dogfight, the blue aircraft transitions from a disadvantaged situation $(1,0)$ or $(1,-1)$ through autonomous decision-making, avoiding a shift towards a situation of red absolute advantage and blue absolute disadvantage represented by $(2,-1)$.

Figure 15(a) indicates that our proposed hybrid-action-PPO method achieves decision-making performance comparable to stronger models even when the training level of the decision model is weaker during the drone game. From Figure 15(b), it is evident that this method significantly improves the decision-making performance of weaker drones in asymmetric maneuvering capabilities.

Additionally, the decision method proposed in this paper exhibits high generalization capabilities in different game environments. Figure 16 displays the flight trajectories of drones controlled by the hybrid-

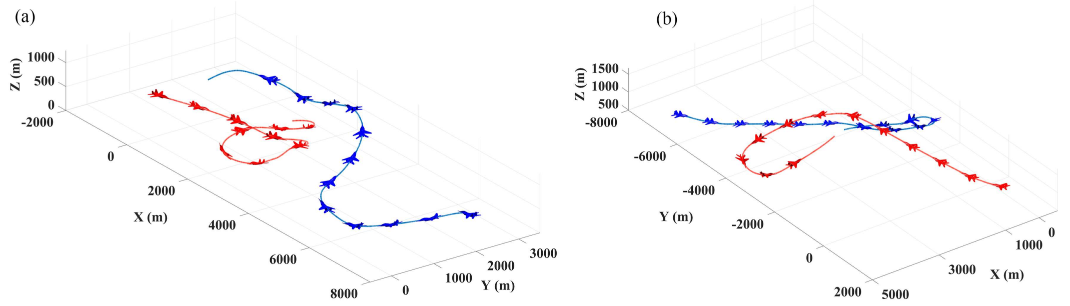

Figure 16 (Color online) Generalization test for 1v1 dynamic game typical scenarios. (a) The interception and evasion scenario; (b) the head-on interception scenario.

action PPO method in other typical dynamic game scenarios.

Figure 16 shows the results of the generalization test of the hybrid-action-PPO method. In this test, red utilizes proportional guidance for terminal guidance, while blue employs the hybrid-action-PPO method, various initial positions and velocity distributions of the unmanned aircraft are designed, and interception evasion and head-on confrontation game objectives are defined. The test validates the method's generalization capability and robustness, demonstrating that even in previously unseen situations, the method can adapt to new game scenarios and achieve good performance.

Figure 17 demonstrates the generalization capability of the hybrid-action-PPO method in a 1v2 asymmetric game scenario. Figure 17 illustrates the typical adversarial scenario in a 1v2 situation, where the blue UAVs achieve coordinated pursuit against the red UAV solely based on autonomous decision-making without information exchange. The blue and green lines represent the trajectories of the two blue UAVs, while the red line depicts the trajectory of the red UAV. From Figures 17(a) and (b), it can be observed that the blue UAVs achieve coordinated pursuit against the red UAV, taking advantage of their numerical superiority, implementing both front and rear encirclement, as well as coordinated pursuit in high and low altitudes. Figure 17(c) demonstrates the red UAV's ability to penetrate the encirclement by employing high angle-of-attack maneuvers in a disadvantageous situation. Figure 17 illustrates the good performance of our approach in UAV adversarial decision-making in an asymmetric environment, showcasing the emergence of collaborative decision-making among UAVs and indicating the robustness and generalization capabilities of our method.

Figure 18 presents the dynamic coordination of UAV trajectories in emergency avoidance during asymmetric adversarial scenarios. In Figure 18(a), when there is a tendency for trajectory collision during the pursuit of the two blue UAVs, the decision algorithm demonstrates the ability to control one UAV for emergency evasion. Figure 18(b) illustrates the coordinated decision-making ability of the blue UAVs, with one intercepting the red UAV from the front and the other accelerating through a descent from the rear, highlighting the algorithm's potential for intelligent applications in cluster scenarios.

Table 4 shows the comparison of this method with other methods, it can be seen from the data that our method is superior to the traditional PPO method in both symmetric and asymmetric environments. Due to the proposed mixed strategy structure of discrete and continuous actions, compared with the DDPG method used solely for deterministic policy, our method has greatly improved. Compared with the SAC method which also solves discrete and continuous mixed actions, our method has also improved, and the SAC method is prone to local convergence of strategies, resulting in the premature emergence of equilibrium situations during game playing, so that the reward average is lower than expected. In summary, both in terms of the generalization performance of the strategy and the superiority of the method, our method can achieve better results in the solution of dynamic control decisions in the UAV game process and realize the advanced nature of the algorithm.

6 Conclusion

This paper focuses on dynamic control strategies and methods for UAVs in pursuit and evasion scenarios, with applications in civilian and military contexts. The aim is to enhance the accuracy and real-time capability of fixed-wing UAVs in actual combat situations. To address the errors caused by waypoint control in close-range UAV encounters, this study proposes a more accurate and stable heading attitude

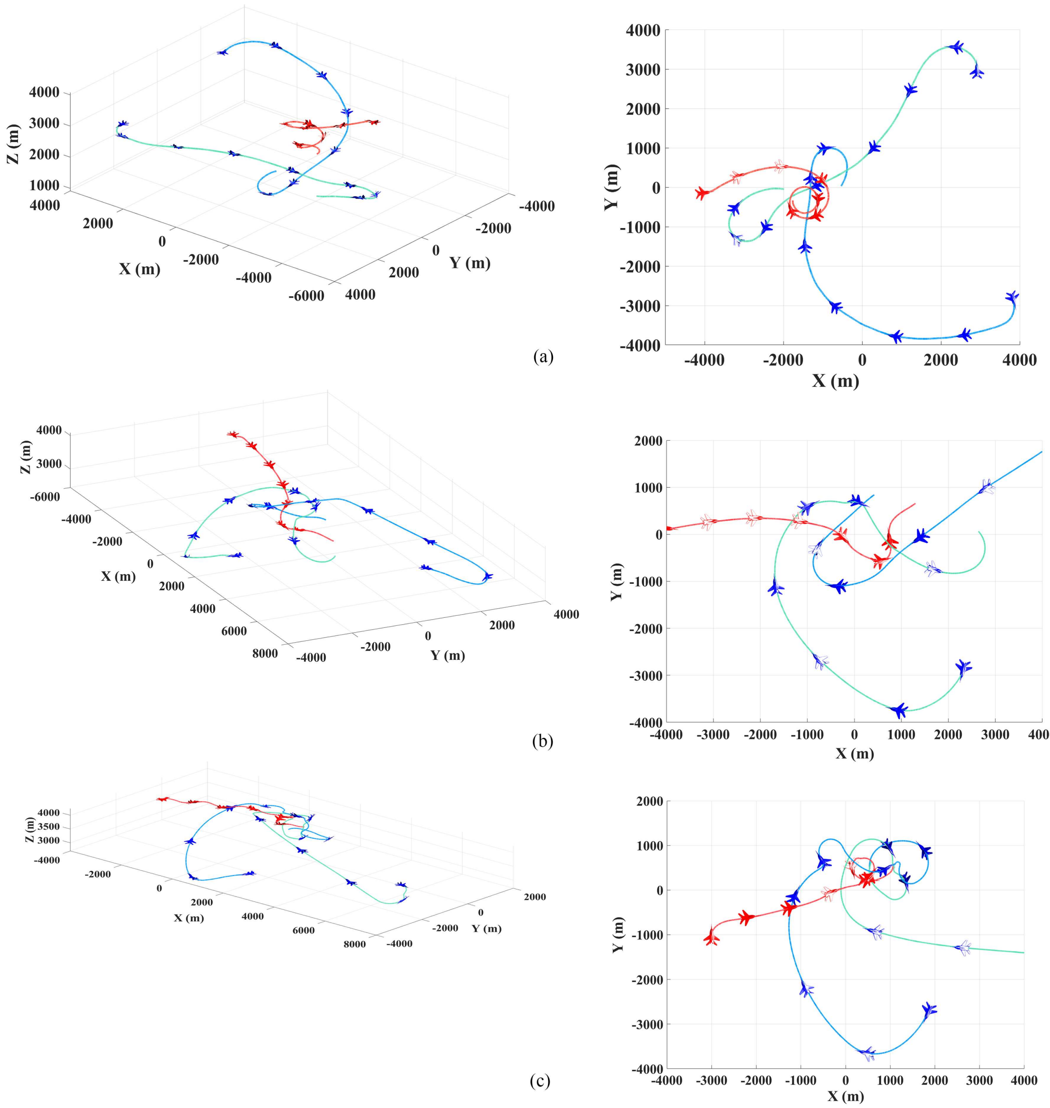

Figure 17 (Color online) Generalization tests for 1v2 dynamic game typical scenarios. (a) Blue aircraft surrounds the red aircraft from both front and rear; (b) blue aircraft pursues the red aircraft in both high and low altitudes; (c) red aircraft performs a high angle of attack maneuver to escape the encirclement by the blue aircraft.

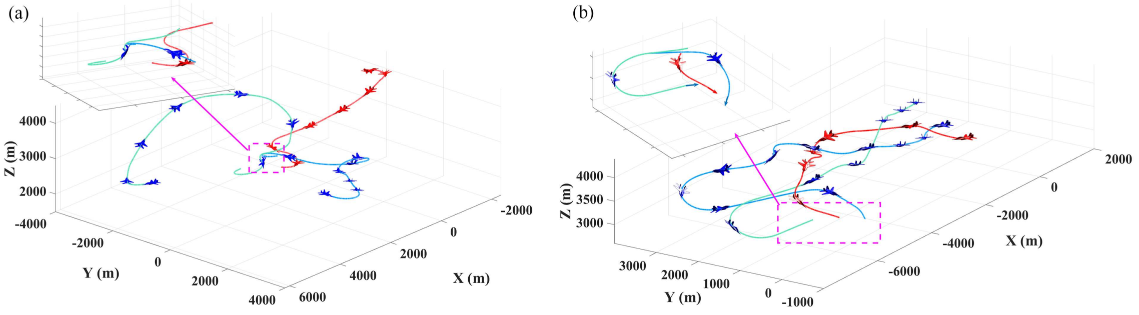

Figure 18 (Color online) Emergency avoidance decision-making and coordinated decision-making for UAVs in an asymmetric environment. (a) The blue aircraft cluster makes an emergency avoidance decision; (b) dynamic coordinated decision of the blue aircraft cluster.

control approach that combines heading and speed. Furthermore, to enhance the maneuverability and strategy responsiveness of UAVs in dynamic encounters at different distances, a flight space adaptive grid model is introduced to dynamically adjust the grid scale, improving the response speed of the decision

Table 4 Comparison of our work with other methods (red UAV gains tactical advantage situation).

Line-of-sight angles (°)	Method	Advantage situation probability	Reward (mean ± std)
red = 30, blue = 30	Ours	0.064	11.348 ± 36.803
	PPO	0.052	9.774 ± 22.513
	DDPG	0.037	7.662 ± 19.219
	SAC	0.058	11.020 ± 30.117
red = 50, blue = 30	Ours	0.115	22.749 ± 51.259
	PPO	0.098	17.519 ± 40.314
	DDPG	0.062	15.275 ± 3 0.579
	SAC	0.105	17.231 ± 49.718
red = 70, blue = 30	Ours	0.292	46.003 ± 70.048
	PPO	0.152	32.637 ± 50.126
	DDPG	0.138	28.856 ± 30.599
	SAC	0.213	38.102 ± 59.717

algorithm and the flight maneuverability of the aircraft. Based on the adaptive grid design and heading attitude control theory, a reinforcement learning control method that integrates discrete and continuous actions is proposed, providing a unified decision learning framework for the UAV's game process. This approach enhances the convergence speed and attitude stability of the decision model in close-range dynamic encounters, as well as its decision robustness. Additionally, this paper verifies the algorithm's generalization performance in interception, evasion, and head-on engagement scenarios. The results demonstrate the method's strong robustness, improving the control accuracy and winning probability of fixed-wing UAVs in unknown combat scenarios. In conclusion, the research findings in this paper provide valuable references for dynamic control strategies and methods in UAV pursuit and evasion games, offering new insights and approaches for related fields of research and application. In future work, we plan to combine flight tests to achieve tactical action design, refinement, and optimization for small fixed-wing UAVs using this approach, laying a technological foundation for UAV swarm game and combat.

Acknowledgements This work was supported by China National Defense Basic Research Programs (Grant No. JCKY2021204B-104).

References

1 Fraga-Lamas P, Ramos L, Mondéjar-Guerra V, et al. A review on IoT deep learning UAV systems for autonomous obstacle detection and collision avoidance. Remote Sens, 2019, 11: 2144

2 Tahir M A, Mir I, Islam T U. A review of UAV platforms for autonomous applications: comprehensive analysis and future directions. IEEE Access, 2023, 11: 52540–52554

3 Wei Z, Zhu M, Zhang N, et al. UAV-assisted data collection for Internet of Things: a survey. IEEE Int Things J, 2022, 9: 15460–15483

4 Hamajima K, Yasukawa K, Ueba M, et al. Design and evaluation on onboard antenna pointing control system for a wireless relay system using fixed-wing UAV. Aerospace, 2023, 10: 323

5 Melkou L, Hamerlain M, Rezoug A. Fixed-wing UAV attitude and altitude control via adaptive second-order sliding mode. Arab J Sci Eng, 2018, 43: 6837–6848

6 Choi J, Seo M, Shin H S, et al. Adversarial swarm defence using multiple fixed-wing unmanned aerial vehicles. IEEE Trans Aerosp Electron Syst, 2022, 58: 5204–5219

7 Giagkos A, Tuci E, Wilson M S, et al. UAV flight coordination for communication networks: genetic algorithms versus game theory. Soft Comput, 2021, 25: 9483–9503

8 Espinoza T, Dzul A, García L, et al. Nonlinear controllers applied to fixed-wing UAV. In: Proceedings of the 9th Electronics, Robotics and Automotive Mechanics Conference, 2012. 243–248

9 Espinoza T, Parada R, Dzul A, et al. Linear controllers implementation for a fixed-wing MAV. In: Proceedings of International Conference on Unmanned Aircraft Systems (ICUAS), 2014. 1081–1090

10 Wang B, Zhang Y, Zhang W. A composite adaptive fault-tolerant attitude control for a quadrotor UAV with multiple uncertainties. J Syst Sci Complex, 2022, 35: 81–104

11 Zheng X, Li H, Ahn C K, et al. NN-based fixed-time attitude tracking control for multiple unmanned aerial vehicles with nonlinear faults. IEEE Trans Aerosp Electron Syst, 2022, 59: 1738–1748

12 Gugan G, Haque A. Path planning for autonomous drones: challenges and future directions. Drones, 2023, 7: 169

13 Liu L, Wang X, Yang X, et al. Path planning techniques for mobile robots: review and prospect. Expert Syst Appl, 2023, 227: 120254

14 Zhang Z, Jiang J, Wu J, et al. Efficient and optimal penetration path planning for stealth unmanned aerial vehicle using minimal radar cross-section tactics and modified A-Star algorithm. ISA Trans, 2023, 134: 42–57

15 Lu S, Liu D, Li D, et al. Enhanced teaching-learning-based optimization algorithm for the mobile robot path planning problem. Appl Sci, 2023, 13: 2291

16 Wu X, Bai W, Xie Y, et al. A hybrid algorithm of particle swarm optimization, metropolis criterion and RTS smoother for path planning of UAVs. Appl Soft Comput, 2018, 73: 735–747

17 Lin S, Li F, Li X, et al. Improved artificial bee colony algorithm based on multi-strategy synthesis for UAV path planning. IEEE Access, 2022, 10: 119269

18 Wang F, Wang J, Chen X. Evacuation entropy path planning model based on hybrid ant colony-artificial fish swarm algorithms. In: Proceedings of IOP Conference Series: Materials Science and Engineering, 2019

19 Contreras-Cruz M A, Ayala-Ramirez V, Hernandez-Belmonte U H. Mobile robot path planning using artificial bee colony and evolutionary programming. Appl Soft Comput, 2015, 30: 319–328

20 Wang J, Li B, Meng M Q H. Kinematic constrained bi-directional RRT with efficient branch pruning for robot path planning. Expert Syst Appl, 2021, 170: 114541

21 Huang C, Zhou X, Ran X, et al. Adaptive cylinder vector particle swarm optimization with differential evolution for UAV path planning. Eng Appl Artif Intell, 2023, 121: 105942

22 Zhou Y, Su Y, Xie A, et al. A newly bio-inspired path planning algorithm for autonomous obstacle avoidance of UAV. Chin J Aeronautics, 2021, 34: 199–209

23 Diao Q, Zhang J, Liu M, et al. A disaster relief UAV path planning based on APF-IRRT* fusion algorithm. Drones, 2023, 7: 323

24 Roberge V, Tarbouchi M, Labonte G. Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning. IEEE Trans Ind Inf, 2012, 9: 132–141

25 Sandberg A, Sands T. Autonomous trajectory generation algorithms for spacecraft slew maneuvers. Aerospace, 2022, 9: 135

26 Raigoza K, Sands T. Autonomous trajectory generation comparison for de-orbiting with multiple collision avoidance. Sensors, 2022, 22: 7066

27 Li K, Wang Y, Zhuang X, et al. A penetration method for UAV based on distributed reinforcement learning and demonstrations. Drones, 2023, 7: 232

28 Wang Y, Li X, Zhuang X, et al. A sampling-based distributed exploration method for UAV cluster in unknown environments. Drones, 2023, 7: 246

29 Zhen Y, Hao M, Sun W. Deep reinforcement learning attitude control of fixed-wing UAVs. In: Proceedings of the 3rd International Conference on Unmanned Systems (ICUS), 2020. 239–244

30 Huang X, Luo W, Liu J. Attitude control of fixed-wing UAV based on DDQN. In: Proceedings of Chinese Automation Congress (CAC), 2019. 4722–4726

31 Bøhn E, Coates E M, Reinhardt D, et al. Data-efficient deep reinforcement learning for attitude control of fixed-wing UAVs: field experiments. IEEE Trans Neural Netw Learn Syst, 2024, 35: 3168–3180

32 Xie T, Xian B, Gu X. Fixed-time convergence attitude control for a tilt trirotor unmanned aerial vehicle based on reinforcement learning. ISA Trans, 2023, 132: 477–489

33 ud Din A F, Mir I, Gul F, et al. Deep reinforcement learning for integrated non-linear control of autonomous UAVs. Processes, 2022, 10: 1307

34 Zhang L, Jabbari B, Ansari N. Deep reinforcement learning driven UAV-assisted edge computing. IEEE Int Things J, 2022, 9: 25449–25459

35 Liu Y, Wang H, Liu B, et al. Learning-based compound docking control for UAV aerial recovery: methodology and implementation. IEEE ASME Trans Mechatron, 2022, 28: 1706–1717

36 Wei Q, Yang Z, Su H, et al. Monte Carlo-based reinforcement learning control for unmanned aerial vehicle systems. Neurocomputing, 2022, 507: 282–291

37 Wan K, Gao X, Hu Z, et al. Robust motion control for UAV in dynamic uncertain environments using deep reinforcement learning. Remote Sens, 2020, 12: 640

38 Lee W, Park G, Joe I. UAV path planning based on reinforcement learning for fair resource allocation in UAV-relayed cellular networks. In: Proceedings of Information Science and Applications, 2019. 53–63

39 Omoniwa B, Galkin B, Dusparic I. Optimizing energy efficiency in UAV-assisted networks using deep reinforcement learning. IEEE Wireless Commun Lett, 2022, 11: 1590–1594

40 Xu S, Zhang X, Li C, et al. Deep reinforcement learning approach for joint trajectory design in multi-UAV IoT networks. IEEE Trans Veh Technol, 2022, 71: 3389–3394

41 Silvirianti, Shin S Y. Energy-efficient multidimensional trajectory of UAV-aided IoT networks with reinforcement learning. IEEE Int Things J, 2022, 9: 19214–19226

42 Huang H, Yang Y, Wang H, et al. Deep reinforcement learning for UAV navigation through massive MIMO technique. IEEE Trans Veh Technol, 2019, 69: 1117–1121

43 Byun H J, Nam H. Autonomous control of unmanned aerial vehicle for chemical detection using deep reinforcement learning. Electron Lett, 2022, 58: 423–425

44 Puente-Castro A, Rivero D, Pazos A, et al. UAV swarm path planning with reinforcement learning for field prospecting. Appl Intell, 2022, 52: 14101–14118

45 Hu Z, Wan K, Gao X, et al. Deep reinforcement learning approach with multiple experience pools for UAV's autonomous motion planning in complex unknown environments. Sensors, 2020, 20: 1890

46 Ma B, Liu Z, Dang Q, et al. Deep reinforcement learning of UAV tracking control under wind disturbances environments. IEEE Trans Instrum Meas, 2023, 72: 1–13

47 Li B, Gan Z, Chen D, et al. UAV maneuvering target tracking in uncertain environments based on deep reinforcement learning and meta-learning. Remote Sens, 2020, 12: 3789

48 Bhagat S, Sujit P B. UAV target tracking in urban environments using deep reinforcement learning. In: Proceedings of International Conference on Unmanned Aircraft Systems (ICUAS), 2020. 694–701

49 Akhloufi M A, Arola S, Bonnet A. Drones chasing drones: reinforcement learning and deep search area proposal. Drones, 2019, 3: 58

50 Ajmera Y, Singh S P. Autonomous UAV-based target search, tracking and following using reinforcement learning and YOLOFlow. In: Proceedings of IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), 2020. 15–20

51 Moon J, Papaioannou S, Laoudias C, et al. Deep reinforcement learning multi-UAV trajectory control for target tracking. IEEE Int Things J, 2021, 8: 15441–15455

52 Wang T, Qin R, Chen Y, et al. A reinforcement learning approach for UAV target searching and tracking. Multimed Tools Appl, 2019, 78: 4347–4364

53 Yin S, Zhao S, Zhao Y, et al. Intelligent trajectory design in UAV-aided communications with reinforcement learning. IEEE Trans Veh Technol, 2019, 68: 8227–8231

54 Yu Z, Li J, Xu Y, et al. Reinforcement learning-based fractional-order adaptive fault-tolerant formation control of networked fixed-wing UAVs with prescribed performance. IEEE Trans Neural Netw Learn Syst, 2024, 35: 3365–3379

55 Jiang Z, Song G. A deep reinforcement learning strategy for UAV autonomous landing on a platform. In: Proceedings of International Conference on Computing, Robotics and System Sciences (ICRSS), 2022. 104–109

56 Xie J, Peng X, Wang H, et al. UAV autonomous tracking and landing based on deep reinforcement learning strategy. Sensors, 2020, 20: 5630

57 Mosali N A, Shamsudin S S, Mostafa S A, et al. An adaptive multi-level quantization-based reinforcement learning model for enhancing UAV landing on moving targets. Sustainability, 2022, 14: 8825

58 Rodriguez-Ramos A, Sampedro C, Bavle H, et al. A deep reinforcement learning strategy for UAV autonomous landing on a moving platform. J Intell Robot Syst, 2019, 93: 351–366

59 Hu J, Zhang H, Song L. Reinforcement learning for decentralized trajectory design in cellular UAV networks with sense-and-send protocol. IEEE Int Things J, 2018, 6: 6177–6189

60 Ouahouah S, Bagaa M, Prados-Garzon J, et al. Deep-reinforcement-learning-based collision avoidance in UAV environment.

IEEE Int Things J, 2021, 9: 4015–4030

61 Singla A, Padakandla S, Bhatnagar S. Memory-based deep reinforcement learning for obstacle avoidance in UAV with limited environment knowledge. IEEE Trans Intell Transp Syst, 2019, 22: 107–118

62 Kim S, Park J, Yun J K, et al. Motion planning by reinforcement learning for an unmanned aerial vehicle in virtual open space with static obstacles. In: Proceedings of the 20th International Conference on Control, Automation and Systems (ICCAS), 2020. 784–787

63 Liu J, Wang Z, Zhang Z. The algorithm for UAV obstacle avoidance and route planning based on reinforcement learning. In: Proceedings of the 11th International Conference on Modelling, Identification and Control (ICMIC2019), 2020. 747–754

64 Xu G, Jiang W, Wang Z, et al. Autonomous obstacle avoidance and target tracking of UAV based on deep reinforcement learning. J Intell Robot Syst, 2022, 104: 60

65 Li Y, Zhang S, Ye F, et al. A UAV path planning method based on deep reinforcement learning. In: Proceedings of IEEE USNC-CNC-URSI North American Radio Science Meeting (Joint with AP-S Symposium), 2020. 93–94

66 Zhao Y J, Zheng Z, Zhang X Y, et al. Q learning algorithm based UAV path learning and obstacle avoidence approach. In: Proceedings of the 36th Chinese Control Conference (CCC), 2017. 3397–3402

67 Tu G T, Juang J G. UAV path planning and obstacle avoidance based on reinforcement learning in 3D environments. Actuators, 2023, 12: 57

68 Zhu J, Fu X, Qiao Z, et al. UAVs maneuver decision — making method based on transfer reinforcement learning. Comput Intell Neurosci, 2022, 2022: 1–12

69 AlMahamid F, Grolinger K. Reinforcement learning algorithms: an overview and classification. In: Proceedings of IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), 2021. 1–7