SCIEPublish

Article

Multi-Objective Distributed Real-Time Trajectory Planning for Gliding Aircraft Cluster

Jiaming Yu, Qinglin Sun * and Hao Sun

College of Artificial Intelligence, Nankai University, Tianjin 300350, China; 1120230239@mail.nankai.edu.cn (J.Y.); sunh@nankai.edu.cn (H.S.)

* Corresponding author. E-mail: sunql@nankai.edu.cn (Q.S.)

Received: 4 September 2024; Accepted: 14 October 2024; Available online: 18 October 2024

ABSTRACT: A new combat strategy that enables coordinated operations of gliding aircraft clusters for multi-target strikes imposes higher demands on the coordination, real-time responsiveness, and strike accuracy of gliding aircraft clusters. Due to the high speed and large inertia characteristics of gliding aircraft, traditional trajectory planning methods often face challenges such as long computation times and difficulty in responding to dynamic environments in real-time when dealing with large-scale gliding aircraft clusters. This paper proposes a distributed cooperative trajectory planning method for multi-target strikes by gliding aircraft clusters to address this issue. By introducing a multi-objective distributed real-time trajectory planning approach based on Multi-Agent Deep Deterministic Policy Gradients (MADDPG), the gliding aircraft execute distributed cooperative trajectory planning based on the trained model. Due to its robust real-time performance, the gliding aircraft do not need to recalculate trajectories for different initial positions of the cluster. Simulation results show that the average error between the gliding aircraft cluster and the target point is 2.1 km, with a minimum error of 0.06 km and a hit rate of 96.6%, verifying the significant advantages of this method in real-time planning capability and strike accuracy.

Keywords: Gliding aircraft cluster; Trajectory planning; Multi agent-deep deterministic policy gradient; Distributed collaboration

1. Introduction

Gliding aircraft are launched by rocket boosters or lifted to a certain altitude or operational area by other carriers. Then, they utilize aerodynamic principles to begin unpowered gliding outside the Earth's atmosphere, ultimately engaging in target strikes or landing. Gliding aircraft have advantages in speed and maneuverability and possess the capability to execute long-range missions and penetrate air defense systems. Early gliding aircraft generally performed single-mission operations. However, as military technology has advanced, the traditional 'one-on-one' combat mode for gliding aircraft has become increasingly difficult to handle more challenging missions [1]. The new combat mode adopts gliding aircraft clusters, engaging in point-to-point coordinated operations. Compared to the traditional 'one-on-one' combat mode, this method offers multiple advantages: rapid dynamic clustering based on real-time battlefield conditions, executing different tasks for different targets, providing diversity in combat strategies; the gliding aircraft can strike targets simultaneously, greatly improving strike efficiency and battlefield coverage; due to the large number and diverse routes of the cluster, it increases the difficulty of the enemy's defense; the deterrence power of cluster attacks far exceeds that of traditional 'one-on-one' methods, creating greater strategic pressure on the enemy; even if some aircraft are intercepted or malfunction, other aircraft can continue the mission, ensuring the continuity and reliability of the operation. Based on these advantages, the real-time dynamic clustering and intra-cluster distributed space-time cooperative guidance architecture of gliding aircraft clusters can enhance penetration probability and achieve precise, synchronized strikes.

Gliding aircraft represent a category of vehicles, exemplified by missiles, spaceplanes, airdrop gliders, and re-entry vehicles. As shown in Figure 1, they differ from traditional powered aircraft in that they are unpowered and have high cruising speeds, making timely adjustments to their flight trajectories challenging. Therefore, real-time planning of

https://doi.org/10.70322/amsm.2024.10008

feasible trajectories for clusters of gliding aircraft becomes an essential prerequisite for coordinated control and precise landing [2]. In the static task space, the multi-gliding aircraft cooperative problem is a typical trajectory planning problem under complex constraints [3]. In recent years, some scholars have conducted research on trajectory planning methods for clusters or formations. Reference [4] discusses the overall 'resilience' requirements of clusters and swarm-based agents such as UAVs. Reference [5] proposed a multi-aircraft collaborative trajectory planning method based on an improved Dubins-RVO method and symplectic pseudospectral method, which can generate feasible trajectories and achieve high-precision tracking in complex environments. Reference [6] addressed the path planning problem for multi-UAV formations in a known environment using an improved artificial potential field method combined with optimal control techniques. Reference [7] investigated the use of an improved grey wolf optimizer algorithm to solve the multi-UAV cooperative path planning problem in complex confrontation environments. It is evident that most of these studies focus on unmanned aerial vehicles, with few dedicated to trajectory planning for gliding aircraft clusters based on their specific characteristics. Moreover, traditional trajectory planning methods often entail long computation times and pre-set trajectories that struggle to adapt to dynamic environments, leading to delayed responses and low damage efficiency in gliding aircraft clusters. Due to current onboard computing capacity limitations, trajectory optimization should be considered in an offline state [8,9]. Deep Reinforcement Learning offers an innovative solution for multi-objective cooperative trajectory planning for gliding aircraft clusters, enhancing aircraft clusters. coordination and combat capabilities.

Figure 1. Gliding aircraft.

Traditional deep reinforcement learning methods are mostly applied to individual learning tasks, such as value function-based reinforcement learning methods [10,11] and policy search-based reinforcement learning methods [12–16]. There is already a considerable amount of research applying deep reinforcement learning to trajectory planning tasks for individuals. Reference [17] addressed the three-dimensional path planning problem for UAVs in complex environments using a deep reinforcement learning approach. Reference [18] optimized UAV trajectory and UAV-TU association using a double deep Q-network algorithm to enhance system performance and quality of service in mobile edge computing networks. Reference [19] studied Artificial Intelligence methods and key algorithms applied to UAV swarm navigation and trajectory planning. Reference [20] studied a method based on explainable deep neural networks to solve the problem of autonomous navigation for quadrotor UAVs in unknown environments. However, this study focuses on trajectory planning for a cluster of multiple gliding aircraft. Some scholars have already applied deep reinforcement learning methods for multi-agent collaboration to trajectory planning. Reference [21] proposed a STAPP method based on a multi-agent deep reinforcement learning algorithm to simultaneously solve the target assignment and path planning problems for multiple UAVs in dynamic environments. Reference [22] proposed a multi-layer path planning algorithm based on reinforcement learning, which improves UAV path planning performance in various environments by combining global and local information. Reference [23] used a multi-agent reinforcement learning approach to solve the problem of flexible data collection path planning for UAV teams in complex environments.

Accordingly, this paper proposes a multi-objective cooperative trajectory planning method for gliding aircraft clusters based on Multi-Agent Deep Deterministic Policy Gradients (MADDPG), utilizing the three degrees of freedom of gliding aircraft, and designing a reward function with initial constraints, terminal constraints, real-time path constraints, and collision avoidance. This method can plan feasible flight trajectories in real-time for each unit in the gliding aircraft cluster, achieving coordinated multi-aircraft strikes.

2. Problem Statement

Illustrated in Figure 2 is the working principle of a gliding aircraft, divided into four main stages:

1. Boost Phase: In this stage, the aircraft relies on its propulsion system to accelerate and ascend. This phase typically involves the aircraft gaining initial speed and altitude, ensuring it can enter the subsequent gliding flight phase.
2. Inertial Phase: The aircraft enters the inertial phase after the boost phase. During this stage, the propulsion system has ceased operation, and the aircraft continues to ascend or maintain speed through inertia.
3. Glide Phase: The glide phase is the primary flight stage of the aircraft, where it utilizes its aerodynamic design to glide without power. This phase usually covers a long distance, and the aircraft can adjust its attitude to alter its trajectory, achieving greater maneuverability and stealth.
4. Dive Phase: The dive phase begins with a rapid descent towards the target. This phase is typically aimed at increasing attack speed, enhancing the element of surprise, and reducing the likelihood of interception by enemy defense systems.

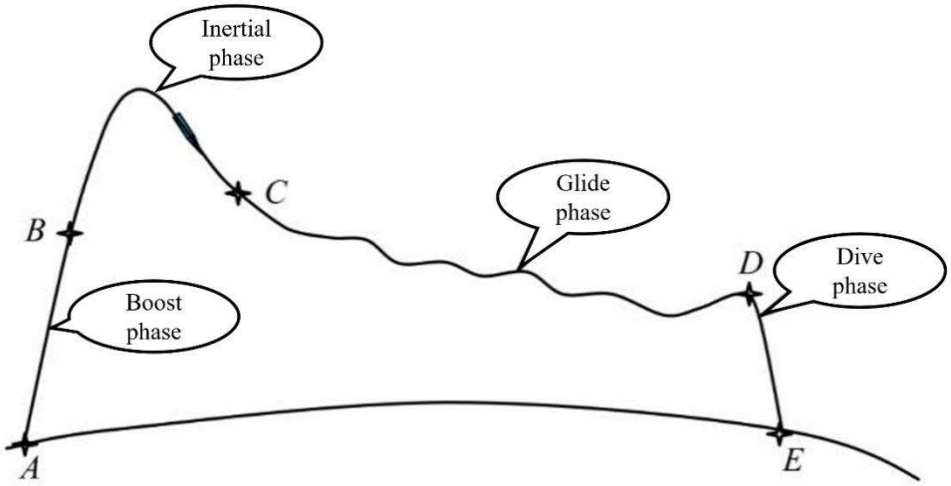


Figure 2. The working principle of the gliding aircraft.

In the various flight phases of a gliding aircraft cluster, trajectory planning during the glide phase is relatively more important. First, the glide phase typically involves long-distance flight, where each aircraft in the cluster must adjust its trajectory based on the overall mission objectives and tactical requirements to avoid collisions and reach the target. Second, since the aircraft no longer relies on a propulsion system during this phase, trajectory planning must take into account the effective management of energy. Finally, the trajectory planning in the glide phase directly affects the accuracy of the gliding aircraft's distance to the target at the terminal moment, thereby creating conditions for success in the final strike phase (such as the dive phase).

3. Gliding Aircraft Model and Trajectory Constraints

3.1. Gliding Aircraft Model

This paper primarily focuses on trajectory planning during the gliding phase. Real-time responsiveness is a key consideration in the design and trajectory planning of gliding aircraft. Due to the high-speed characteristics of the aircraft, their motion states may undergo significant changes within an extremely short period. Hence, a model capable of rapid adaptation and computation is necessary to ensure the accuracy of trajectory planning. We have employed a 3-DOF model for the trajectory planning of gliding aircraft. This choice is primarily based on the high cruising speed characteristics of the gliding aircraft and the high demand for real-time computation of trajectories. Compared to more complex models with higher degrees of freedom, the 3-DOF model reduces computational load, allowing trajectory planning to be completed in a shorter time frame, thus meeting the real-time response requirements of gliding aircraft in dynamic environments. Each aircraft within the gliding aircraft cluster is established based on the following 3-DOF model:

$$\begin{cases} \dot{x} = v_{xy}\cos\varphi + v_{wx} \\ \dot{y} = v_{xy}\sin\varphi + v_{wy} \\ \dot{z} = v_z \\ \ddot{\varphi} = u \end{cases} \tag{1}$$

where $[x, y, z]$ represents the position of each gliding aircraft in the cluster; $[wx, wy, wz]$ represents the wind speed components along the *x*, *y*, and *z* axes, respectively; v_{xy} denotes the horizontal velocity of each gliding aircraft; v_z represents the vertical velocity of each gliding aircraft, which in this study is assumed to be constant as the aircraft are unpowered; φ indicates the yaw angle of each gliding aircraft; $\ddot{\varphi}$ denotes the angular acceleration of each gliding aircraft; and u represents the control input during flight, which in this study is set as angular acceleration.

3.2. Trajectory Constraints

The trajectory constraints of the gliding aircraft cluster determine the optimization direction of each vehicle's trajectory planning method. Based on the mission characteristics of the gliding aircraft, the trajectory constraints should first include initial constraints and terminal constraints. The initial constraints for each individual in the gliding aircraft swarm are represented as follows:

$$\begin{cases} x(t_0) = x_0 \\ y(t_0) = y_0 \\ z(t_0) = z_0 \\ \varphi(t_0) = \varphi_0 \end{cases} \tag{2}$$

where t_0 represents the initial time of the gliding aircraft's mission; [*x*, *y*, *z*] represents the initial position of the individual vehicle; and φ represents the initial yaw angle of the individual vehicle.

The terminal constraints for each gliding aircraft are as follows:

$$\begin{cases} t_f = \dfrac{z_0 - z_f}{v_z} \\ x(t_f) = x_f \\ y(t_f) = y_f \\ z(t_f) = z_f \end{cases} \tag{3}$$

where t_f represents the terminal time; $\left[x(t_f), y(t_f)\right]$ represents the terminal position of the individual gliding aircraft. This paper assumes $z_0 = 0$, so the value of t_f is determined by the initial altitude z_0 and the vertical velocity v_z. Equation (3) indicates that if the vertical velocity is constant, the terminal time is also a fixed value.

In the trajectory planning model for the gliding aircraft cluster, the reward function for each aircraft needs to consider not only the terminal constraints but also real-time path constraints, control input constraints, and collision avoidance within the cluster. The reward function will be designed based on these constraints.

4. Method

4.1. Reward Function

The reward function is crucial in cooperative trajectory planning for gliding aircraft clusters. Its primary purpose is to provide a quantified evaluation for the start, end, and every action step during the mission execution, guiding the gliding aircraft to make optimal decisions. The real-time reward focuses on guiding gliding aircraft towards the target point at each step and ensuring they precisely reach the target at the stipulated moment. The real-time path reward should be considered first:

$$D_t = \sqrt{\left(x_t - x_f\right)^2 + \left(y_t - y_f\right)^2} \tag{4}$$

$$d = \left| \frac{z_t}{v_z} \times v_{xy} - D_t \right| \tag{5}$$

$$r_d = \begin{cases} 2, \text{ if } d < 1 \text{ and } d' < 1 \\ 0.1 \times \left(d' - d\right), \text{ otherwise} \end{cases} \tag{6}$$

where D_t represents the horizontal distance between the system in S_t state and the target point $[x_f, y_f]$; d represents real-time path constraint, which is the absolute error value between the remaining flight distance of the parafoil system in the S_t state and the horizontal distance from the target point. v_z is the vertical velocity of a high-speed gliding aircraft, and v_{xy} is the horizontal velocity.

The real-time reward value should also consider the constraints of the control input:

$$r_u = 1 - |a_t| - |a_t - a_t'| \tag{7}$$

where r_u represents the real-time control input constraint, which needs to consider both the absolute value of the control input and its fluctuation frequency. Therefore, the real-time reward value can be described as follows:

$$r_t = K1 \times r_d + K2 \times r_u \tag{8}$$

where $K1$ and $K2$ are weight coefficients.

The terminal reward value focuses on the accuracy with which the aircraft reaches the predetermined target point at the end of the trajectory planning task. This paper calculates this reward based on the final distance between the aircraft and its target point. Higher rewards are given when the aircraft accurately reaches or approaches the target point:

$$D_f = \sqrt{\left(x_{t_f} - x_f\right)^2 + \left(y_{t_f} - y_f\right)^2} \tag{9}$$

$$r_f = \begin{cases} K - D_f, t = t_f, D_f \le 5 \\ M, t = t_f, D_f > 5 \end{cases} \tag{10}$$

where $[x_{tf}, y_{tf}]$ represents the final landing position of the gliding aircraft, D_f represents the terminal error to be minimized in the trajectory planning; K and M are constants, with M typically being negative. To prevent the terminal reward r_f from being diluted by the cumulative value of the real-time reward r_t, K should be adjusted in conjunction with the value of γ and tailored according to the specific task requirements.

The collision avoidance reward is designed to encourage gliding aircraft to maintain a safe distance from each other during the mission and to ensure they stay within the designated mission area. If a collision occurs between vehicles or if they go beyond the mission area, a negative reward r_a is given.

4.2. Deep Deterministic Policy Gradient

The Deep Deterministic Policy Gradient (DDPG) algorithm is a deep reinforcement learning method specifically designed for complex problems with continuous action spaces [24]. The structure of DDPG is shown in Figure 3. DDPG combines the advantages of policy-based and value-based approaches, utilizing an Actor-Critic framework. In this framework, the Actor is responsible for generating the policy by directly mapping states to actions, while the Critic evaluates the value of those actions under the given policy.

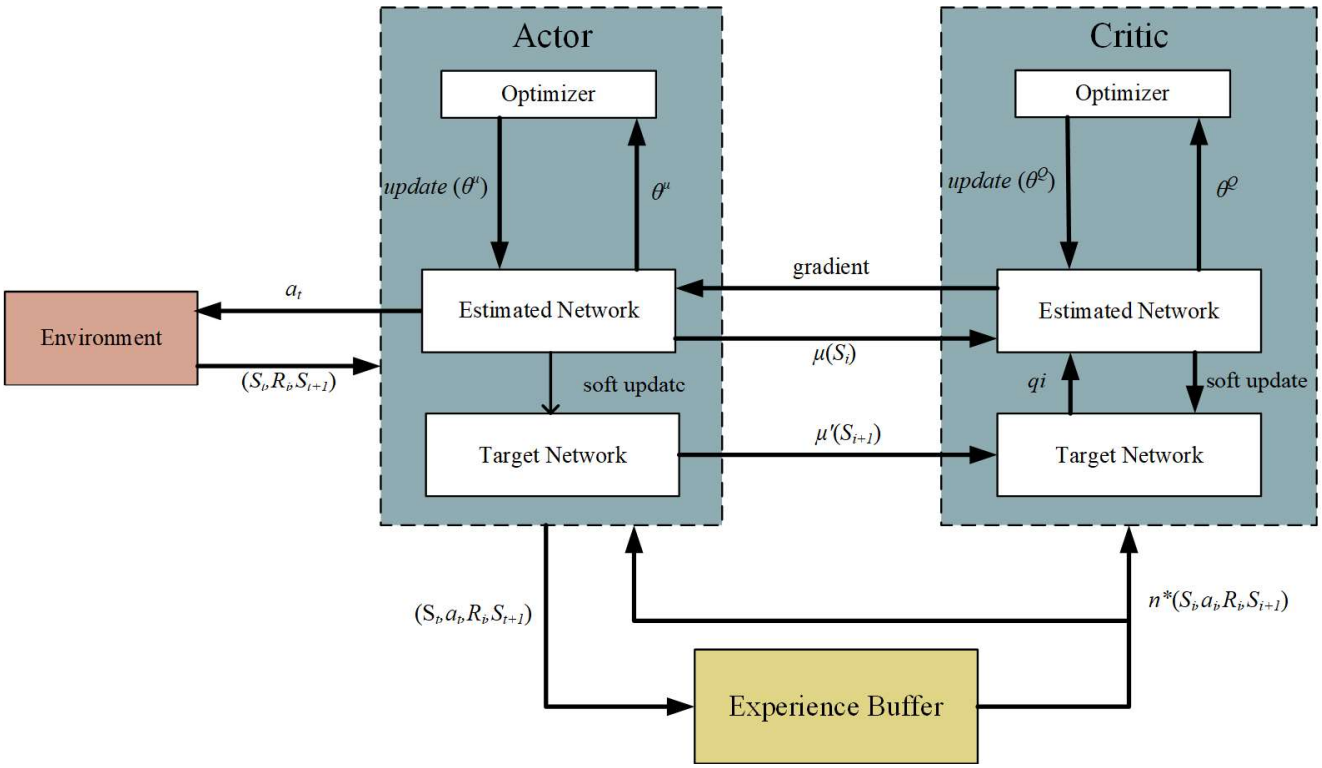


Figure 3. DDPG Structure Diagram.

Q is the evaluated reward value after selecting control input a in the gliding state S:

$$Q\left(S,a|\ \theta^{Q}\right) \tag{11}$$

The DDPG algorithm employs a deterministic policy μ. Under this deterministic policy, the algorithm outputs the optimal action for a given state:

$$a_t = \mu\left(S_t|\ \theta^{\mu}\right) \tag{12}$$

where a_t is the deterministic angular acceleration value obtained directly under each state through the deterministic policy function μ, and θ^{μ} represents the parameters of the Actor Network used to generate deterministic angular acceleration.

The Actor Network $\mu\left(S_t|\ \theta^{\mu}\right)$ in DDPG outputs a deterministic angular acceleration, meaning that the same gliding state S will produce the same angular acceleration a, which can result in limited exploration samples. To increase the planning randomness and exploration of angular acceleration in the DDPG algorithm, random noise should be added to the selected angular acceleration, causing the angular acceleration value to fluctuate. The angular acceleration a after adding noise can be expressed by the following formula:

$$a_t \sim \text{clip}\left(N\left(\mu\left(S_t|\ \theta^{\mu}\right),\sigma^2\right),a_{min},a_{max}\right) \tag{13}$$

where N represents Gaussian noise with a normal distribution; σ represents the variance; a_{min} is the minimum value of the angular acceleration; a_{max} is the maximum value of the angular acceleration.

The Actor and Critic have an estimation network and a target network, respectively. The parameters of the estimation network are trained, while the parameters of the target network are updated using a soft update method. Since the output of the target network is more stable, DDPG uses the target network to calculate the target value. The formula for the soft update of the target network is as follows:

$$\begin{cases} \theta^{Q'} \leftarrow \tau\theta^{Q} + (1-\tau)\theta^{Q'} \\ \theta^{\mu'} \leftarrow \tau\theta^{\mu} + (1-\tau)\theta^{\mu'} \end{cases} \tag{14}$$

where τ represents the soft update rate; θ^{Q} and θ^{μ} are the parameters of the Actor and Critic estimation networks; $\theta^{Q'}$ and $\theta^{\mu'}$ are the parameters of the Actor and Critic target networks.

The actions determined by the target network of the Actor, coupled with the observed values of the environmental state, are utilized as inputs to the target network of the Critic. This process dictates the direction in which the Critic's target network parameters are updated. The formula for the parameter update within the Critic Network is as follows:

$$q_i = r_i + \gamma Q'(S_{i+1}, \mu'(S_{i+1}|\ \theta^{\mu'})|\ \theta^{Q'}) \tag{15}$$

$$L = \frac{1}{n}\sum_{i}^{n}\left(y_i - Q\left(S_i, a_i|\ \theta^{Q}\right)\right)^2 \tag{16}$$

where q_i is the actual evaluation value, computed using the target network, r_i refers to the real reward received γ and represents the reward decay rate. The loss function, denoted as L, is defined as the sum of squared deviations between the actual q_i and the estimated values.

The update of the Actor Network parameters adheres to a deterministic policy:

$$\nabla_{\theta^{\mu}} J = \frac{1}{n}\sum_{i}^{n}\nabla_a Q\left(S, a|\ \theta^{Q}\right)|_{S=S_i, a=\mu(S_i)}\ \nabla_{\theta^{\mu}}\mu\left(S|\ \theta^{\mu}\right)|\mathrm{S}_i \tag{17}$$

where ∇Q, sourced from the critic, guides the direction for updating the parameters of the actor's network. where $\nabla_{\theta^{\mu}}$, sourced from the actor, guides the Actor Network to be more likely to choose the above actions.

4.3. Multi-Agent Deep Deterministic Policy Gradient

The coordinated strike strategy of gliding aircraft clusters involves forming a group of multiple aircraft to jointly attack various targets, enhancing the effectiveness of strikes and breach capabilities and increasing the likelihood of mission completion. Each aircraft in the glider cluster possesses autonomous decision-making and execution capabilities, representing a multi-agent problem where each agent has unique observations and actions.

Using traditional deep reinforcement learning methods to solve the gliding aircraft cluster task presents two challenges. The first challenge arises from the continuous update of planning strategies for each aircraft during the training process. Assuming a cluster involving *N* aircraft, the environment becomes unstable from the perspective of an individual aircraft. The next state is $S' = P\left(S, \mu_1\left(\mathbf{o}_1\right), \cdots, \mu_N\left(\mathbf{o}_N\right)\right), (i = 1, \cdots, N)$, where oi represents the observation of the *i*-th aircraft, and $\mu_i\left(\mathbf{o}_i\right)$ is the angular acceleration chosen based on this observation. Therefore, the next state is influenced by the strategies of all *N* aircraft, making it infeasible to attribute environmental changes to a single planning strategy.

The second challenge is due to the differences in planning strategies among the aircraft during the collaboration process, which typically introduces high variance when using policy gradient methods. In contrast, the MADDPG algorithm is specifically designed to address the issues involved in the coordination of multiple gliding aircraft [25]. It is based on the Actor-Critic framework of DDPG and incorporates the following improvements:

1. During the training phase, each gliding aircraft can utilize the positions and observations of other aircraft.
2. During the testing phase, each gliding aircraft plans its trajectory based solely on its observations.

MADDPG uses a centralized approach during the training phase and allows for distributed execution during the testing phase. Each gliding aircraft acts as an Actor and is trained with the help of a Critic. To accelerate the training process, the Critic of each aircraft utilizes other aircraft's observations and planning strategies. Each aircraft can learn the trajectory planning models of other gliding aircraft and effectively use these models to optimize its trajectory planning strategy. After the training is completed, the Actor plans trajectories based on the observations of a single aircraft, making this a distributed trajectory planning method that can be tested in a decentralized task space.

As shown in Figure 4, the centralized Critic not only uses the observations of the current aircraft for training but also leverages the observation information and trajectory planning models of other aircraft. During the testing phase, the trained actor is used to distribute planning decisions. In the experience replay buffer of MADDPG, both a and r are vectors composed of the angular acceleration values and rewards of multiple gliding aircraft.

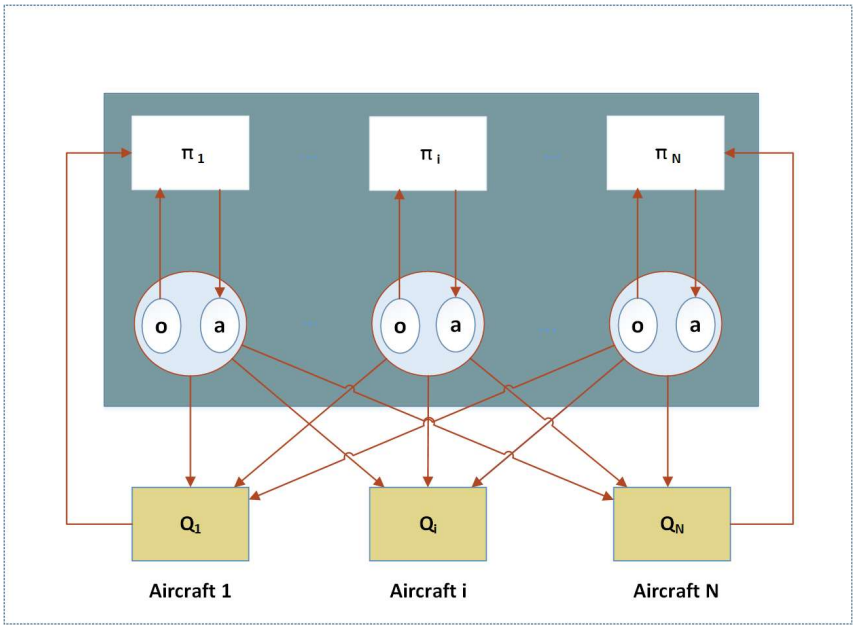


Figure 4. MADDPG framework.

For the gliding aircraft cluster task, MADDPG has *N* Actors and 2*N* Critics. $\boldsymbol{\mu}=\{\mu_1,\cdots,\mu_N\}$ represents the trajectory planning strategies of *N* gliding aircraft, with parameters denoted as $\boldsymbol{\theta}=\{\theta_1,\ldots,\theta_N\}$. The angular acceleration of the *i*-th gliding aircraft under a certain observation is given by $\mu_i(a_i|\ o_i)$. The policy gradient for the *i*-th gliding aircraft is given by the following formula:

$$\nabla_{\theta_i}J(\mu_i)=\mathrm{E}_{s,a\sim B}\left[\nabla_{\theta_i}\mu_i(a_i|\ o_i)Q_i^{\mu}(x,a_1,\ldots,a_N)|_{a_i=\mu_i(o_i)}\right] \tag{18}$$

where x represents the complete environmental information, that is, the observation set of the gliding aircraft cluster, denoted as $x=(o_1,\ldots,o_N)$. In the training and testing process of the trajectory planning model, each gliding aircraft needs to obtain the observation information of other aircraft to achieve coordination and collision avoidance, o_i represents the observation information of the *i*-th gliding aircraft. Q_i^{μ} is the centralized evaluation function for the *i*-th aircraft. Its input consists of the angular acceleration a_i chosen by each aircraft and the environmental information x. Each Q_i is learned independently, allowing the reward for each aircraft to be designed independently. Each element in the experience replay buffer B is a four-tuple (x_t,a_t,r_t,x_{t+1}), recording the flight experiences of all gliding aircraft. Its structure is shown in Figure 5, where $a_t=\{a_1,\ldots,a_N\}$ and $r_t=\{r_1,\ldots,r_N\}$.

The state space represents the environmental information that each glider can perceive during decision-making, typically including:

(1) The current position and velocity vector of the glider.

(2) The target location for the glider's current mission, which guides the glider toward a predetermined destination.The relative position and velocity of nearby gliders to enable collision avoidance and coordinated planning.

Therefore, the state space can be expressed as $o_{i_t}=(x_{i_t},y_{i_t},z_{i_t},v_{x_{i_t}},v_{z_{i_t}},x_f,y_f)$.

The action space represents the actions each glider can choose at each decision step, which in this paper is set as the angular acceleration of the yaw angle.

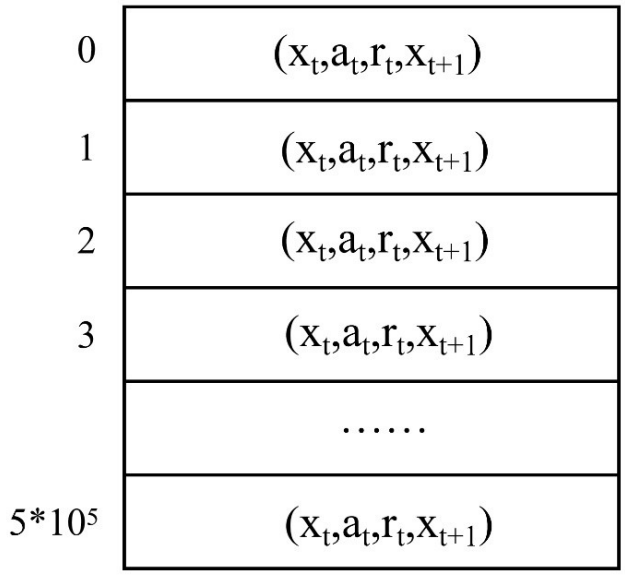


Figure 5. Experience replay buffer.

The centralized action-value function is updated through backpropagation:

$$L\left(\boldsymbol{\theta}_i\right)=\mathrm{E}_{\mathbf{x},\mathbf{a},\mathbf{r},\mathbf{x}'}\left[\left(Q_i^{\mu}\left(\mathbf{x},\mathbf{a}_1,\cdots,\mathbf{a}_N\right)-q_i\right)^2\right] \tag{19}$$

$$q_i=r_i+\gamma Q_i^{'}\left(\mathbf{x}^{'},\mathbf{a}_1^{'},\cdots,\mathbf{a}_N^{'}\right)|_{a_j^{'}=\mu_j^{'}\left(o_j\right)} \tag{20}$$

where $\mu^{'}=\left\{\mu_{\theta_i}{}^{'},\ldots,\mu_{\theta_N}{}^{'}\right\}$ represents the action function of the target network, corresponding to the planning strategy parameters $\theta_i^{'}$ of the *i*-th aircraft in the target network. $Q_i^{'}$ is the evaluation function for the *i*-th aircraft in the target network.

The pseudocode of MADDPG is shown in Algorithm 1.

Algorithm 1 MADDPG algorithm pseudocode

1: Initial estimated Critic Network parameters θ^{Q}, and estimated Actor Network parameter θ^{μ}

2: Initial target network parameters $\theta^{Q^{'}}\leftarrow\theta^{Q}$

3: Set initial values of hyper-parameters according to the task requirements: experience playback buffer pool B, mini batch size n, Actor Network learning rate l_a, Critic Network learning rate l_c, maximum episode E, soft update rate τ

4: Initialize environment state x

5: **for** *t* = 1 to *T* **do**

6: **for** each gliding aircraft select angular acceleration $a_t\sim\mathrm{clip}\left(N\left(\mu\left(S_t|\ \theta^{\mu}\right),\sigma^2\right),a_{min},a_{max}\right)$ **do**

7: Execute action $a_t=\left(a_1,\ldots,a_N\right)$

8: Observe reward r_t in current state and new state x_{t+1}

9: Store transition tuple $\left(x_t,a_t,r_t,x_{t+1}\right)$ of this step in B

end for

10: Sample mini-batch of n transactions $\left(x^j,a^j,r^j,S^{j+1}\right)$ from B

11: Compute Q-targets $q_i=r_i+\gamma Q_i^{'}\left(\mathbf{x}^{i},\mathbf{a}_1^{'},\cdots,\mathbf{a}_N^{'}\right)|_{a_j^{'}=\mu_j^{'}\left(o_j\right)}$

12: Update estimated network parameters of the critic by minimizing loss:

$$L=\frac{1}{n}\sum_{i}^{n}\left(y_i-Q\left(S_i,a_i|\ \theta^{Q}\right)\right)^2$$

13: Update the actor policy using sampled policy gradient:

$$\nabla_{\theta^\mu} J = \frac{1}{n}\sum_{i}^{n} \nabla_a Q\left(S, a|\ \theta^Q\right)|_{S=S_i, a=\mu(S_i)}\ \nabla_{\theta^\mu}\mu\left(S|\ \theta^\mu\right)|\mathrm{S}_i$$

14: Update parameters of target network of the critic and the actor:

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

end for

4.4. Experimental Setup and Algorithm Parameters

To verify the correctness of the algorithm proposed in this study, PyCharm was used as the development and simulation environment in Python language version 3.9. Table 1 shows the parameters of the MADDPG algorithm, and Table 2 shows the configuration of the experimental machine.

Table 1. MADDPG algorithm parameters.

Parameter Name	Value
Number of training episodes	3000
γ	0.99
τ	0.01
l_c	0.0005
l_a	0.001
Experience replay buffer B size	5×10^5
Batch size	256

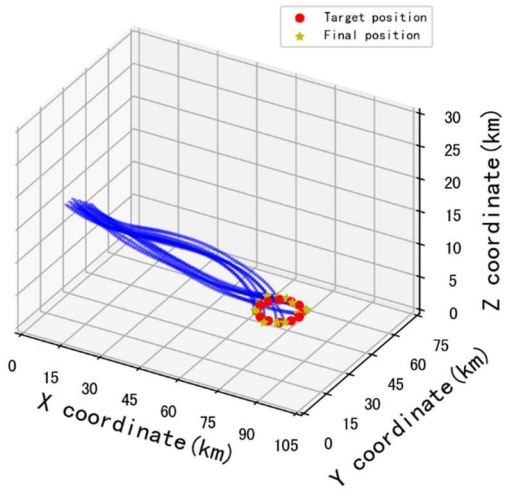
Table 2. Experimental machine configuration.

Parameter Name	Value
GPU	RTX 3060
CPU	12th Gen Intel(R) Core(TM) i7-12700F
Memory	16.0 GB
Solid State Drive	1 TB
Operating System	Windows 11
Programming Environment	Python 3.9, Tensorflow 2.0

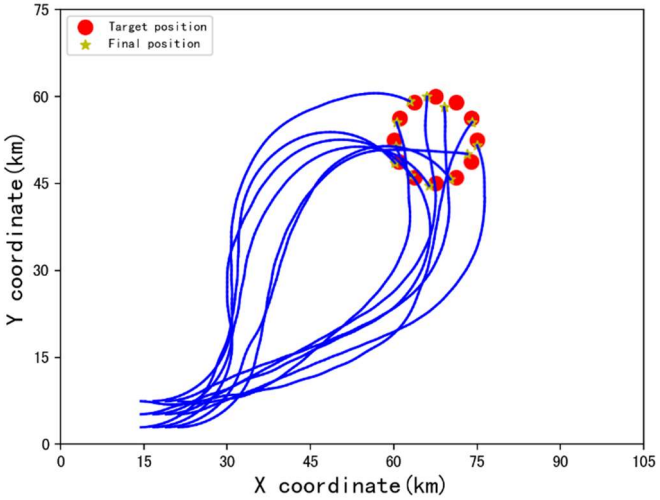
5. Simulation

The task space size is 105 km × 75 km × 30 km. The initial altitude of the gliding aircraft cluster is set to 15 km. The horizontal and vertical speeds of the aircraft can be set based on the aircraft model and mission requirements. In this study, the horizontal speed is set to 2 Mach, while the vertical speed is a constant value, fixing the flight time. The aircraft are required to arrive near their mission targets at the designated time. A random wind field with a maximum speed of 30 m/s was added during testing. The trajectory planning model can use each aircraft's positional information to plan trajectories in real-time, and this ability to adjust trajectories in real-time effectively reduces positional deviations caused by wind disturbances.

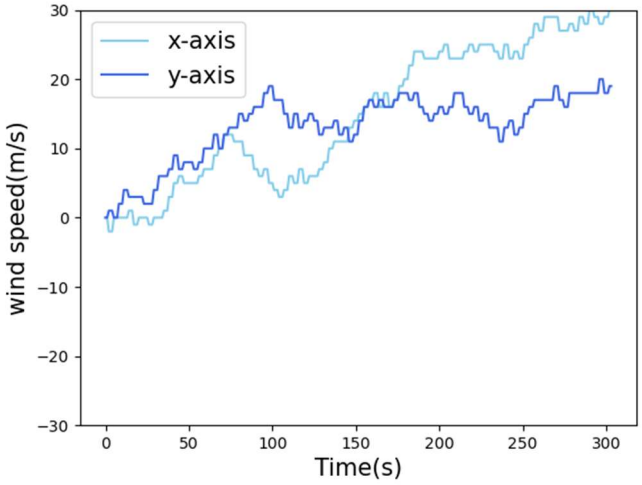
At the beginning of each round, the gliding aircraft cluster will randomly initialize its position in the task area. After 100 rounds of testing, where the gliding aircraft operated in a distributed cooperative manner, the average error between the gliding aircraft cluster and the target point was 2.1 km, with a minimum error of 0.06 km and a maximum error of 6.3 km. This study assumes that an error of more than 5 km is considered a miss, and the hit rate reaches 96.6%. Figure 6, Figure 7, and Figure 8 display different trajectory planning cases.



(**a**)

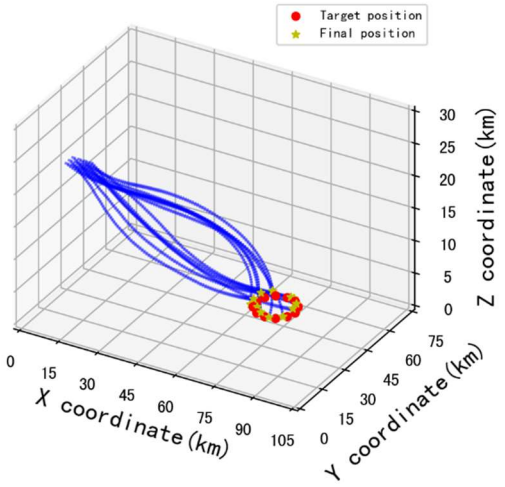


(**b**)

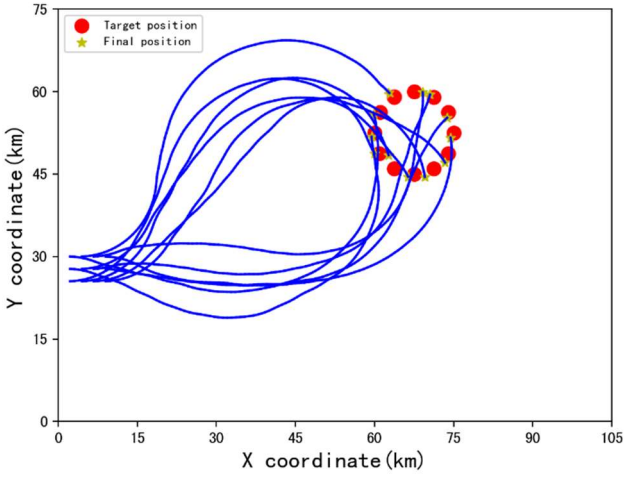


(**c**)

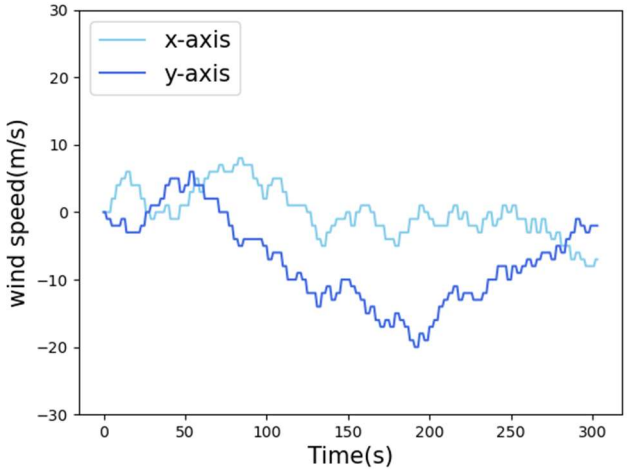
Figure 6. Simulation result I. (**a**) 3D trajectory; (**b**) 2D trajectory; (**c**) wind speed.



(**a**)

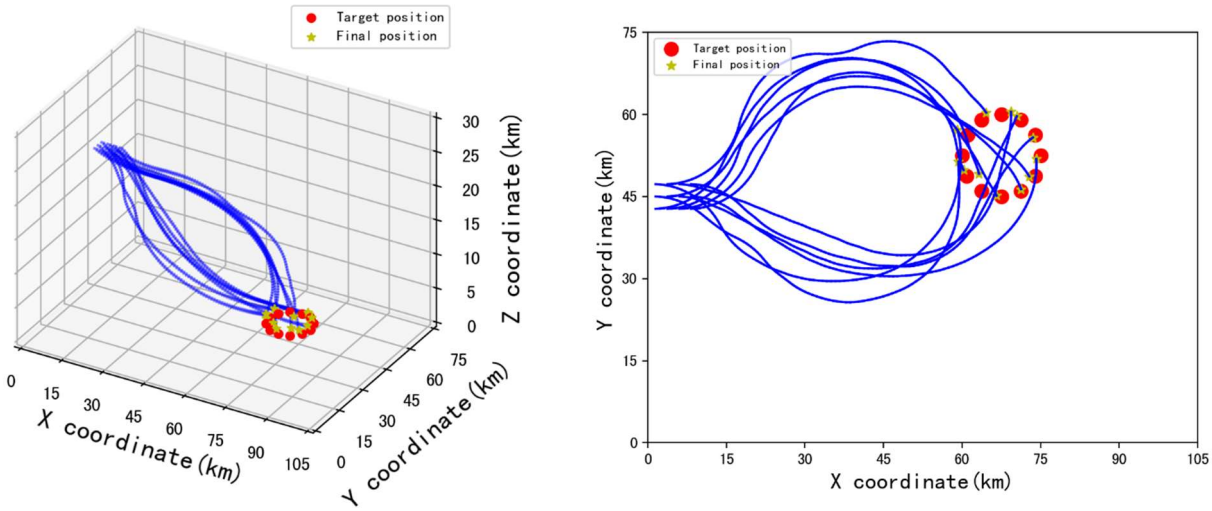


(**b**)

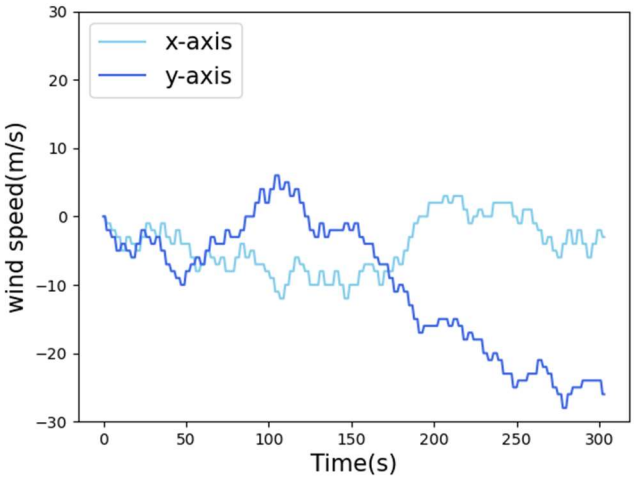


(**c**)

Figure 7. Simulation result II. (**a**) 3D trajectory; (**b**) 2D trajectory; (**c**) wind speed.



(**a**) (**b**)



(**c**)

Figure 8. Simulation result III. (**a**) 3D trajectory; (**b**) 2D trajectory; (**c**) wind speed.

The above simulation results demonstrate that the multi-objective trajectory planning method for gliding aircraft clusters proposed in this paper successfully plans feasible trajectories for each of the 12 aircraft in the trajectory planning task, enabling them to reach their respective target points at the designated times concurrently.

6. Conclusions

Traditional trajectory planning methods often rely on complex computational models, which substantially increase computation time in large-scale scenarios and make adaptation to dynamic environmental changes challenging. This is especially challenging when dealing with gliding aircraft clusters, where obtaining effective planning results within a short time becomes difficult. This paper proposes a multi-objective trajectory planning method for gliding aircraft clusters based on the MADDPG algorithm to address this issue. Each gliding aircraft executes trajectory planning strategies in a distributed manner based on a pre-trained model, eliminating the need to recalculate trajectories for different initial positions. Additionally, a reward function tailored to multi-objective tasks is designed for each aircraft in the cluster, considering trajectory accuracy, energy minimization, and collision avoidance within the cluster. Simulation results show that the proposed trajectory planning method can plan optimal trajectories for gliding aircraft at different positions in real-time.

To reduce the training time of the trajectory planning model and ensure real-time decision efficiency, this paper simplifies the gliding aircraft model to a 3DOF model and pre-assigns a target to each aircraft. Future research could first consider using a higher degree of freedom model while maximizing decision efficiency. Additionally, it could incorporate real-time target allocation based on the positions of different aircraft within the flight mission to improve landing accuracy and mission success rate.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No.62103204).

Author Contributions

Conceptualization, Q.S.; Methodology, H.S.; Validation, J.Y.; Writing—Original Draft Preparation, J.Y.; Writing—Review & Editing, J.Y.

Ethics Statement

Not applicable.

Informed Consent Statement

Not applicable.

Funding

This research received no external funding.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Li Y, Wu Y, Qu X. Chicken Swarm–Based Method for Ascent Trajectory Optimization of Hypersonic Vehicles. *J. Aerosp. Eng.* **2017**, *30*, 04017043.
2. Shen Z, Lu P. Onboard Generation of Three-Dimensional Constrained Entry Trajectories. *J. Guid. Control Dyn.* **2003**, *26*, 111–121.
3. Xu X-P, Yan X-T, Yang W-Y, An K, Huang W, Wang Y. Algorithms and applications of intelligent swarm cooperative control: A comprehensive survey. *Prog. Aerosp. Sci.* **2022**, *135*, 100869.
4. Abhishek P, Medrano FA. Examining application-specific resiliency implementations in UAV swarm scenarios. *Intell. Robot.* **2023**, *3*, 453–478.
5. Liu J, Han W, Wang X, Li J. Research on Cooperative Trajectory Planning and Tracking Problem for Multiple Carrier Aircraft on the Deck. *IEEE Syst. J.* **2020**, *14*, 3027–3038.
6. Chen Y, Yu J, Su X, Luo G. Path Planning for Multi-UAV Formation. *J. Intell. Robot. Syst.* **2014**, *77*, 229–246.
7. Xu C, Xu M, Yin C. Optimized multi-UAV cooperative path planning under the complex confrontation environment. *Comput. Commun.* **2020**, *162*, 196–203.

8. Zhao J, Zhou R, Jin X. Progress in reentry trajectory planning for hypersonic vehicle. *J. Syst. Eng. Electron.* **2014**, *25*, 627–639.
9. Wei Z, Huang C, Ding D, Huang H, Zhou H. UCAV Formation Online Collaborative Trajectory Planning Using hp Adaptive Pseudospectral Method. *Math. Probl. Eng.* **2018**, *2018*, 1–25.
10. Wang Y, Liu H, Zheng W, Xia Y, Li Y, Chen P, et al. Multi-Objective Workflow Scheduling With Deep-Q-Network-Based Multi-Agent Reinforcement Learning. *IEEE Access* **2019**, *7*, 39974–39982.
11. Fahrmann D, Jorek N, Damer N, Kirchbuchner F, Kuijper A. Double Deep Q-Learning With Prioritized Experience Replay for Anomaly Detection in Smart Environments. *IEEE Access* **2022**, *10*, 60836–60848.
12. Xu Y-H, Yang C-C, Hua M, Zhou W. Deep Deterministic Policy Gradient (DDPG)-Based Resource Allocation Scheme for NOMA Vehicular Communications. *IEEE Access* **2020**, *8*, 18797–18807.
13. Meng W, Zheng Q, Shi Y, Pan G. An Off-Policy Trust Region Policy Optimization Method With Monotonic Improvement Guarantee for Deep Reinforcement Learning. *IEEE Trans. Neural. Netw. Learn. Syst.* **2022**, *33*, 2223–2235.
14. Li B, Gan Z, Chen D, Sergey Aleksandrovich D. UAV Maneuvering Target Tracking in Uncertain Environments Based on Deep Reinforcement Learning and Meta-Learning. *Remote Sens.* **2020**, *12*, 3789.
15. Chikhaoui K, Ghazzai H, Massoud Y. PPO-based Reinforcement Learning for UAV Navigation in Urban Environments. In Proceedings of the 2022 IEEE 65th International Midwest Symposium on Circuits and Systems (MWSCAS), Fukuoka, Japan, 7–10 August 2022; pp. 1–4.
16. Wong C-C, Chien S-Y, Feng H-M, Aoyama H. Motion Planning for Dual-Arm Robot Based on Soft Actor-Critic. *IEEE Access* **2021**, *9*, 26871–26885.
17. Xie R, Meng Z, Wang L, Li H, Wang K, Wu Z. Unmanned Aerial Vehicle Path Planning Algorithm Based on Deep Reinforcement Learning in Large-Scale and Dynamic Environments. *IEEE Access* **2021**, *9*, 24884–24900.
18. Liu Q, Shi L, Sun L, Li J, Ding M, Shu FS. Path Planning for UAV-Mounted Mobile Edge Computing With Deep Reinforcement Learning. *IEEE Trans. Veh. Technol.* **2020**, *69*, 5723–5728.
19. Elfatih NM, Ali ES, Saeed RA. Navigation and Trajectory Planning Techniques for Unmanned Aerial Vehicles Swarm. In *Artificial Intelligence for Robotics and Autonomous Systems Applications*; Azar AT, Koubaa A, Eds.; Springer International Publishing: Cham, Switzerland, 2023; pp. 369–404.
20. He L, Aouf N, Song B. Explainable Deep Reinforcement Learning for UAV autonomous path planning. *Aerosp. Sci. Technol.* **2021**, *118*, 107052.
21. Qie H, Shi D, Shen T, Xu X, Li Y, Wang L. Joint Optimization of Multi-UAV Target Assignment and Path Planning Based on Multi-Agent Reinforcement Learning. *IEEE Access* **2019**, *7*, 146264–146272.
22. Cui Z, Wang Y. UAV Path Planning Based on Multi-Layer Reinforcement Learning Technique. *IEEE Access* **2021**, *9*, 59486–59497.
23. Bayerlein H, Theile M, Caccamo M, Gesbert D. Multi-UAV Path Planning for Wireless Data Harvesting With Deep Reinforcement Learning. *IEEE Open J. Commun. Soc.* **2021**, *2*, 1171–1187.
24. Sumiea EH, Abdulkadir SJ, Alhussian HS, Al-Selwi SM, Alqushaibi A, Ragab MG, et al. Deep deterministic policy gradient algorithm: A systematic review. *Heliyon* **2024**, *10*, e30697.
25. Zheng S, Liu H. Improved Multi-Agent Deep Deterministic Policy Gradient for Path Planning-Based Crowd Simulation. *IEEE Access* **2019**, *7*, 147755–147770.