aerospace

MDPI

Article

A Multi-UCAV Cooperative Decision-Making Method Based on an MAPPO Algorithm for Beyond-Visual-Range Air Combat

Xiaoxiong Liu *, Yi Yin, Yuzhan Su and Ruichen Ming

School of Automation, Northwestern Polytechnical University, Xi'an 710129, China
* Correspondence: nwpulxx@outlook.com

Abstract: To solve the problems of autonomous decision making and the cooperative operation of multiple unmanned combat aerial vehicles (UCAVs) in beyond-visual-range air combat, this paper proposes an air combat decision-making method that is based on a multi-agent proximal policy optimization (MAPPO) algorithm. Firstly, the model of the unmanned combat aircraft is established on the simulation platform, and the corresponding maneuver library is designed. In order to simulate the real beyond-visual-range air combat, the missile attack area model is established, and the probability of damage occurring is given according to both the enemy and us. Secondly, to overcome the sparse return problem of traditional reinforcement learning, according to the angle, speed, altitude, distance of the unmanned combat aircraft, and the damage of the missile attack area, this paper designs a comprehensive reward function. Finally, the idea of centralized training and distributed implementation is adopted to improve the decision-making ability of the unmanned combat aircraft and improve the training efficiency of the algorithm. The simulation results show that this algorithm can carry out a multi-aircraft air combat confrontation drill, form new tactical decisions in the drill process, and provide new ideas for multi-UCAV air combat.

Keywords: multiple unmanned combat aerial vehicles; multi-agent proximal policy optimization; the missile attack area model; comprehensive reward; centralized training and distributed execution

Citation: Liu, X.; Yin, Y.; Su, Y.; Ming, R. A Multi-UCAV Cooperative Decision-Making Method Based on an MAPPO Algorithm for Beyond-Visual-Range Air Combat. *Aerospace* **2022**, *9*, 563. https://doi.org/10.3390/aerospace9100563

Academic Editor: Chao-Yang Lee and Ang-Hsun Tsai

Received: 7 August 2022
Accepted: 23 September 2022
Published: 28 September 2022

1. Introduction

In actual air combat, pilots often have to make a series of tactical decisions according to the established combat tasks, air combat tactics, aircraft performance, and the current situation between themselves and the enemy. In this process, a series of decisions that are shown to be made by air combat pilots will become certain air combat tactics, and the more abundant that number of air combat tactics that they have, then the greater the possibility of wiping out the enemy is. With the development of air combat weapons, modern air combat has formed a pattern of beyond-visual-range air combat, which is supplemented by short-range air combat [1,2]. At present, air combat is mainly based on multiple unmanned combat aerial vehicles (UCAVs) performing cooperative operations, wherein, UCAVs cooperate to complete combat tasks, including performing cooperative maneuvers, coordinated strikes, and providing fire cover [3]. Compared with single UCAV decision making, multi-UCAV air combat decision making involves more air combat entities, a more complex air combat environment, and more space for air combat decision making [4,5].

At present, the research on air combat decision making mostly focuses on differential games [6–8], the expert system method [9,10], the Bayesian network method [11,12], the fuzzy reasoning method [13–15], and approximate dynamic programming [16]. Although the application of these theories can solve many air combat decision-related problems, with the continuous demand for the improvement of air combat decision making and the deepening of the research, the relevant methods of game theory have also exposed many defects:

(1) The first of these is the complexity of modeling the real air combat problem. The huge amount of information in a real air combat environment and the rapidly changing state of it brings problems to those that are conducting accurate modeling.
(2) The second of these is the dimension explosion problem that is caused by the growth of the number of game individuals and the decision-making space. The decision-making problems that occur due to there being a large number of game participants bring huge decision-making space dimensions which will directly affect the efficiency and accuracy of the solution.
(3) Finally, it is difficult to solve the problem with an optimal strategy. In the face of complex and dynamic air combat decision-making problems, it is impossible to obtain the analytical solution of the Nash equilibrium.

In contrast, deep reinforcement learning has a unique information perception ability and a strong learning ability [17–19], which can be employed to effectively solve high-dimensional, sequential decision-making problems. Therefore, the application of deep reinforcement learning generates new ideas that can be applied to complex air combat decision-making problems.

Intelligent air combat, which combines deep reinforcement learning and air combat decision making, has been explored and studied by many scholars [20–26]. The reinforcement learning algorithm controls the UCAV in order for it to perform actions and constantly interact with the environment to obtain corresponding returns which will be fed back to the UCAV to guide it to take the next action. At the same time, it constantly adjusts its own strategies according to the feedback of the environment, and finally generates a set of the most worrying strategies. For UCAV autonomous maneuver decision making, Yong-Feng Li proposed a UCAV autonomous maneuver decision-making model for short-range air combat that is based on an MS-DDQN algorithm [20]. To improve the dogfight ability of the unmanned combat aerial vehicles and avoid the deficiencies of traditional methods, such as poor flexibility and a weak decision-making ability, a maneuver method that uses deep learning was proposed by [21]. To improve the efficiency of the reinforcement learning algorithm for the exploration of strategy space, [22] Zhang proposed a heuristic Q-Network method that integrates expert experience and uses expert experience as a heuristic signal to guide the search process. Q. Yang proposed the use of the optimization algorithm to generate the air combat maneuver action value, and the addition of the optimization action as the initial sample of the DDPG replay buffer to filter out a large number of invalid action values and guarantee the correctness of the action value [23].

However, from the research results that have been published, current air combat decisions that have been made are based on deep reinforcement learning procedures and these are mainly aimed at their application for two-dimensional air combat environments, the air combat decisions of single aircraft confrontations, or the decision simulations of some typical air combat tactics. There are many problems in the cooperative over-the-horizon air combat of multiple unmanned aerial vehicles, such as there being many entity types, a large decision-making space, and a high-complexity and slow nature of the training and convergence speed. With the rise of multi-agent reinforcement learning, there are new solutions for multi-agent collaborative decision making [27,28]. In order to solve these problems, the following methods are adopted in this paper:

(1) To simulate the real air combat environment, this paper establishes a six-degree-of-freedom model of the unmanned combat aircraft, constructs the action library of the unmanned combat aircraft, and establishes the missile attack area and the corresponding probability of damage occurring according to both the enemy and us.
(2) This paper designs a set of comprehensive reward functions according to the angle, height, speed, and distance of the unmanned combat aircraft, and the probability of damage occurring to the missile attack area to overcome the problem of sparse rewards in reinforcement learning and improve the training efficiency.
(3) The multi-agent proximal policy optimization approach is designed to optimize the air combat decision-making algorithm, which adopts the architecture of the centralized

training-distributed execution mechanism to improve the decision-making ability of the UCAVs.

(4) A two-to-one battlefield environment is used to simulate and verify the algorithm. The experimental results show that deep reinforcement learning can guide UCAVs to fight, find a way to quickly expand their superiority when they are superior, and find a suitable way to attack to win in an air combat scenario when it is at an absolute disadvantage.

2. Task Scene and Decision Structure Design

A UCAV that is engaged in autonomous air combat can obtain strategy instructions from an intelligent decision-making algorithm according to the current air combat situation that it is in, which can greatly improve the autonomous ability of the UCAV and the efficiency of the mission's execution. In the future air combat system, the air combat mission is set as shown in Figure 1. The blue is an enemy UCAV that is intruding into our airspace and scouting our base. When the initial position of the target UCAV is known, we send two UCAVs to intercept and pursue it. The battlefield environment is composed of three-dimensional space, and the two sides, which are composed of the enemy and us, will engage in a contest that is beyond the visual range in the vast space battlefield. In this paper, a three-dimensional battlefield map is constructed to perform as the task environment, and both of the red and blue UCAVs are the agents, and the action selection of the agents will be completed by the use of a discrete action library. The scene that is in this paper is a situation of an enemy UCAV intruding into our airspace; considering that the enemy UCAV can maneuver and evade and attack the ones on the red side, the task of the red ones is to intercept and drive the blue one away, and when the blue one is shot down, then the red ones complete the interception and pursuit task. If one of the red UCAVs is shot down, it is considered that the one on the blue side has successfully escaped the pursuit of the red ones and completed the reconnaissance mission of the red ones' base.

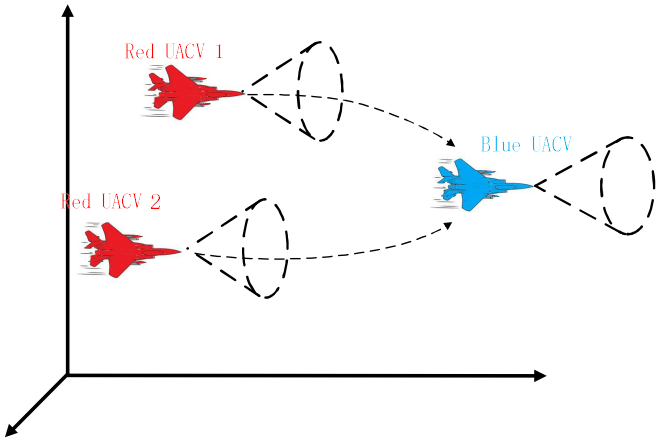


Figure 1. Schematic diagram of air combat.

Based on the above task scenarios and considering that the proximal policy optimization algorithm does not rely on the value function, the strategy function is easy to calculate and has the advantages of having good stability and convergence. This paper designs a multi-agent air combat decision-making framework that is based on a multi-agent proximal policy optimization algorithm (MAPPO). The framework is mainly composed of three parts: a deep reinforcement learning module, an environment module, and a data processing module. The specific structure is shown in Figure 2. The reinforcement learning module is mainly divided into two parts, the central controller and the distributed execution. The central controller is primarily responsible for evaluating the actions that are performed by the UCAVs. The distributed execution mainly outputs the actions that the red UCAVs

and the blue-side UCAV need to perform through the new actor network. The new actor network is updated according to the comparison of the old and new actor networks, and the advantage function is outputted by the critic network. The environment module is mainly composed of the UCAV model, the target allocation, the missile attack area, and the reward module. The target allocation module is responsible for the allocation of air combat targets, while the missile attack zone module will judge the attack result of the UCAV, and the reward module conducts a situational assessment according to the battlefield situation. The function of the data processing module is to process the information that is sent by the environment module, normalize and package the information, and send it to the experience base. Two red UCAVs and one blue UCAV choose their initial actions according to the initial parameters of their policy network, and ones that are on the red and blue sides will perform the corresponding action in the environment to get a new state and reward. At this time, the status, actions, rewards, and other data of both of the sides will be processed into a joint status, with joint actions and joint rewards, by the data processing module after it conducts the packaging, normalization, and formatting procedures. After the reinforcement learning module receives the information data, the critic networks of the red UCAVs and the blue UCAV sample the data, and send the sampled joint state, joint action, and joint reward to the central controller to guide the policy network to update the policy. Then, the red UCAVs and the blue UCAV take their private observations as the input of their updated policy networks, and the new actor networks output their corresponding action decisions. After receiving the action decision of the reinforcement learning module, the UCAVs execute corresponding actions to interact with the air combat environment. The new UCAV status information and the corresponding reward information that is obtained by executing the new actions are processed by the data processing module and sent to the experience base. When the neural network is trained, these data are sampled from the experience base and passed into the neural network training for their training to occur.

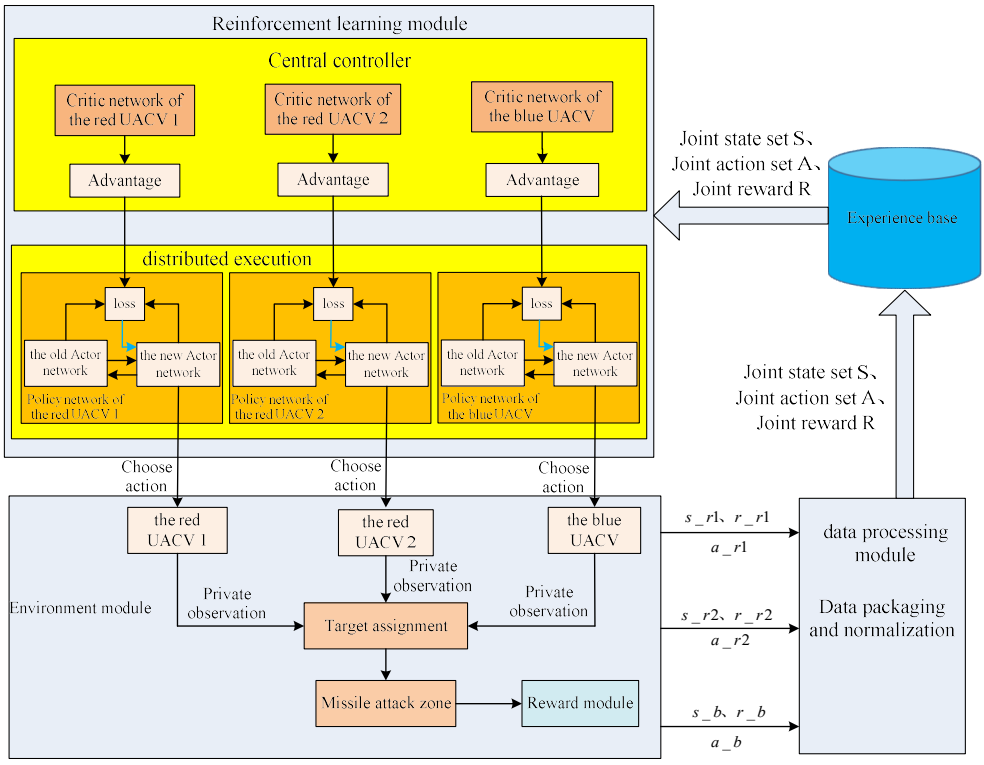


Figure 2. Decision-making framework of two UCAVs and a single UCAV that are engaging in air combat.

As can be seen from the decision structure that has been designed (above), as an agent of autonomous combat, a UCAV must have a decision-making ability. As a new strategy gradient reinforcement learning algorithm, the proximal policy optimization algorithm is especially suitable for autonomous decision making in air combat. According to the UCAV air combat environment, the construct of the UCAV and the missile attack zone model, the design of a comprehensive reward function for optimal decision making, and the construct of a multi-agent air combat decision-making algorithm complete the decision-making design of air combat. The following article will be discussed and designed from these aspects.

3. Air Combat Model of the UCAV under the Decision Structure

3.1. Motion Control Model of the UCAV

The three-dimensional kinematic equations of the UCAV in-ground inertial coordinate system are defined as follows:

$$\begin{cases} dv/dt = g(N_x - \sin\theta) \\ d\psi/dt = gN_z \sin\varphi / v\cos\theta \\ d\theta/dt = (g/v)(N_z\cos\varphi - \cos\theta) \\ dx/dt = v\cos\theta\cos\psi \\ dy/dt = v\cos\theta\sin\psi \\ dz/dt = v\sin\theta \end{cases} \tag{1}$$

Each term in the equations represents the first-order differential equations of the UCAV's flight speed, yaw angle, pitch angle, and three-dimensional coordinate values, respectively, which together determine the calculation method of the UCAV's state update in the process of air combat. N_x is the tangential overload of the UCAV, which indicates the ratio of the thrust to the gravity of the UCAV in the forward speed direction. By changing the size of this, the UCAV can be controlled to accelerate, decelerate, or fly at a uniform speed. N_z is the normal overload of the UCAV, which means that the UCAV is subjected to the overload that is perpendicular to that of the fuselage and upward direction. The dive, pull-up, and smooth flight of the UCAV can be controlled by changing of the size and the vertical height of the UCAV. φ is the roll angle of the body, it represents the angle of the body around its speed direction, and the UCAV can be controlled to turn laterally by changing the size of this.

In this paper, the action coding is carried out by selecting the tangential overload, normal overload, and roll angle of the UCAV; this triad $[N_x, N_z, \varphi]$ is used to represent the actions that are taken at each point in the simulation. The action library that can be taken by the UCAV is shown in Table 1.

Table 1. Instruction analysis of basic air combat maneuver.

Maneuvering Action	Maneuver Coding		
	Tangential Overload N_x	**Normal Overload N_z**	**Roll Angle φ**
Flat flight with fixed length	0	1	0
Accelerate flight	0.3	1	0
Deceleration flight	−0.3	1	0
Turn left	0	1.985	$\pi/3$
Turn right	0	−1.985	$-\pi/3$
Pull up	0	1.5	0
Dive down	0	−1.5	0

3.2. Modeling of the Missile Attack Zone

As shown in Figure 3, the attack area of the UCAV is similar to a cone. To better analyze the attack area, the attack area is cross-sectioned, and the cross-section of the attack

area is a sector. Combined with the performance of the missile and the maneuverability of the enemy UCAV, the probability of damage occurring to the target is analyzed. The sector attack is divided into five parts, and the attack probabilities of these five parts are also different.

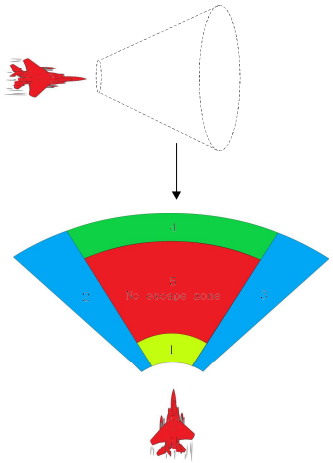


Figure 3. Schematic diagram of the attack area.

The specific division of the attack zone is as follows:

$$\begin{cases} 1, ATA < \varphi_{Mk\max} \text{ and } D_{M\min} < d < D_{Mk\min} \\ 2, \varphi_{M\max} < ATA < \varphi_{Mk\max} \text{ and } D_{M\min} < d < D_{M\max} \text{ and } \arctan \Delta y/\Delta x < 0 \\ 3, \varphi_{M\max} < ATA < \varphi_{Mk\max} \text{ and } D_{M\min} < d < D_{M\max} \text{ and } \arctan \Delta y/\Delta x >= 0 \\ 4, ATA < \varphi_{MK\max} \text{ and } D_{Mk\max} < d < D_{M\max} \\ 5, ATA < \varphi_{MK\max} \text{ and } D_{Mk\min} < d < D_{Mk\max} \end{cases} \quad (2)$$

where ATA is the deviation angle of the drone, $\varphi_{M\max}$ is the maximum off-axis launch angle of the missile, $D_{M\min}$ is the minimum attack distance of the missile, $D_{M\max}$ is the maximum range of missile attack, d is the distance between the drone and the target, $D_{Mk\min}$ is the minimum inescapable distance of a missile, $D_{Mk\max}$ is the maximum inescapable distance of a missile, and $\varphi_{MK\max}$ is the conical angle.

The probability of the target being destroyed is also different in different attack areas; of course, the maneuvering action of the target UCAV will also affect the probability of the target being destroyed.

$$P = \begin{cases} 0.5 \times P_a + 0.5 \times P_d & if\ position(aircraft\_aim) \in 1 \\ 0.8 \times P_a + 0.8 \times P_d & if\ position(aircraft\_aim) \in 2 \\ 0.8 \times P_a + 0.8 \times P_d & if\ position(aircraft\_aim) \in 3 \\ 0.5 \times P_a + 0.5 \times P_d & if\ position(aircraft\_aim) \in 4 \\ 1 & if\ position(aircraft\_aim) \in 5 \end{cases} \quad (3)$$

Among them, $position(aircraf\_aim)$ is the area of the attack zone in which the target is located. The details of P_a and P_d are as follows:

$$P_a = \begin{cases} AA/\pi + 1 & if\ position(aircraft\_aim) \in 1 \\ -(AA/(\pi/6 + ATA)) \times AA + 0.5 & if\ position(aircraft\_aim) \in 2\ and\ \arctan v_y/v_x >= 0\ and\ AA < 5\pi/6 - ATA \\ -(0.5/(5\pi/6 - ATA)) \times AA - 0.5 \times \pi/6 + ATA/(5\pi/6 - ATA) & if\ position(aircraft\_aim) \in 2\ and\ \arctan v_y/v_x >= 0\ and\ AA >= 5\pi/6 - ATA \\ (0.3/(\pi/6 + ATA)) \times AA + 0.5 & if\ position(aircraft\_aim) \in 2\ and\ \arctan v_y/v_x < 0\ and\ AA < 5\pi/6 - ATA \\ (0.3/(ATA - 5\pi/6)) \times AA + (0.5 \times ATA - 43/60)/(ATA - 5\pi/6) & if\ position(aircraft\_aim) \in 2\ and\ \arctan v_y/v_x < 0\ and\ AA >= 5\pi/6 - ATA \\ (0.3/(\pi/6 + ATA)) \times AA + 0.5 & if\ position(aircraft\_aim) \in 3\ and\ \arctan v_y/v_x >= 0\ and\ AA < 5\pi/6 - ATA \\ (0.3/(ATA - 5\pi/6)) \times AA + (0.5 \times ATA - 43/60)/(ATA - 5\pi/6) & if\ position(aircraft\_aim) \in 3\ and\ \arctan v_y/v_x >= 0\ and\ AA >= 5\pi/6 - ATA \\ -(AA/(\pi/6 + ATA)) \times AA + 0.5 & if\ position(aircraft\_aim) \in 3\ and\ \arctan v_y/v_x < 0\ and\ AA < 5\pi/6 - ATA \\ -(0.5/(5\pi/6 - ATA)) \times AA - 0.5 \times (\pi/6 + ATA)/(5\pi/6 - ATA) & if\ position(aircraft\_aim) \in 3\ and\ \arctan v_y/v_x < 0\ and\ AA >= 5\pi/6 - ATA \\ P_a = AA/\pi & if\ position(aircraft\_aim) \in 4 \end{cases} \quad (4)$$

$$P_d = \begin{cases} d/5000 - 2 & if\ position(aircraft\_aim) \in 1 \\ -(12/25)^2 \times (d/1000 - 22.5)^2 + 0.8 & if\ position(aircraft\_aim) \in 2 \\ -(12/25)^2 \times (d/1000 - 22.5)^2 + 0.8 & if\ position(aircraft\_aim) \in 3 \\ -d/10000 + 3.5 & if\ position(aircraft\_aim) \in 4 \end{cases} \tag{5}$$

3.3. Design of a Comprehensive Reward Function

To solve the problem of sparse reward, this paper adopts the method of combining a continuous reward with a sparse reward, which includes the angle reward, height reward, speed reward, and distance reward, which guide the UCAV to carry out air combat and increase the effective reward. As a result, the learning speed is accelerated, and the sparse reward is used as an event reward, as a sign of the end of air combat, and as an evaluation of the UCAV's completion of its mission. The combination of a continuous reward and a sparse reward not only maintains the exploratory nature of the UCAV, but also increases the effective reward and improves the learning efficiency of the algorithm.

The angle reward, which combines the departure angle of the target and the deviation angle of the UCAV in the course of combat, is defined as:

$$r\_a = \begin{cases} \left(1 - \frac{|ATA|}{5\varphi_{Mk\max}}\right)^{\frac{3}{4}} \left(e^{-\frac{90^\circ - |AA|}{60^\circ}}\right)^{\frac{1}{4}} & 0 \le |ATA| < \varphi_{Mk\max}, 0 \le |AA| < 60^\circ \\ \left(1 - \frac{|ATA|}{5\varphi_{Mk\max}}\right)^{\frac{3}{4}} \left(e^{\frac{|AA| - 180^\circ}{360^\circ}}\right)^{\frac{1}{4}} & 0 \le |ATA| < \varphi_{Mk\max}, 60^\circ \le |AA| \le 180^\circ \\ \left(0.8 - \frac{|ATA| - \varphi_{Mk\max}}{2(\varphi_{M\max} - \varphi_{M\max})}\right)^{\frac{2}{3}} \left(e^{-\frac{90^\circ - |AA|}{60^\circ}}\right)^{\frac{1}{3}} & \varphi_{Mk\max} \le |ATA| < \varphi_{M\max}, 0 \le |AA| < 60^\circ \\ \left(0.8 - \frac{|ATA| - \varphi_{Mk\max}}{2(\varphi_{M\max} - \varphi_{M\max})}\right)^{\frac{2}{3}} \left(e^{\frac{|AA| - 180^\circ}{360^\circ}}\right)^{\frac{1}{3}} & \varphi_{Mk\max} \le |ATA| < \varphi_{M\max}, 60^\circ \le |AA| \le 180^\circ \\ \left(0.3 - \frac{|ATA| - \varphi_{M_{\max}}}{10(\varphi_{R_{\max}} - \varphi_{M\max})}\right)^{\frac{1}{2}} \left(e^{-\frac{90^\circ - |AA|}{60^\circ}}\right)^{\frac{1}{2}} & \varphi_{M\max} \le |ATA| < \varphi_{R_{\max}}, 0 \le |AA| < 60^\circ \\ \left(0.3 - \frac{|ATA| - \varphi_{M_{\max}}}{10(\varphi_{R_{\max}} - \varphi_{M\max})}\right)^{\frac{1}{2}} \left(e^{\frac{|AA| - 180^\circ}{360^\circ}}\right)^{\frac{1}{2}} & \varphi_{M\max} \le |ATA| < \varphi_{R_{\max}}, 60^\circ \le |AA| \le 180^\circ \\ \left(0.1 - \frac{|ATA| - \varphi_{R\max}}{10(180^\circ - \varphi_{R\max})}\right)^{\frac{2}{5}} \left(e^{-\frac{90^\circ - |AA|}{60^\circ}}\right)^{\frac{3}{5}} & \varphi_{R_{\max}} \le |ATA| \le 180^\circ, 0 \le |AA| < 60^\circ \\ \left(0.1 - \frac{|ATA| - \varphi_{R\max}}{10(180^\circ - \varphi_{R\max})}\right)^{\frac{2}{5}} \left(e^{\frac{|AA| - 180^\circ}{360^\circ}}\right)^{\frac{3}{5}} & \varphi_{R_{\max}} \le |ATA| \le 180^\circ, 60^\circ \le |AA| \le 180^\circ \end{cases} \tag{6}$$

In this form, $\varphi_{R_{\max}}$, $\varphi_{M\max}$, and $\varphi_{Mk\max}$ are the maximum search deflection angle of the radar, the maximum off-axis launch angle of the missile, and the maximum deviation angle of the target that cannot escape, respectively. There being a positive value for the angle reward means that the UCAV occupies the dominant angle, and therefore, the target is at a disadvantage; there being a negative value for the angle reward means that the target occupies the dominant angle, and therefore, the UCAV is at a disadvantage.

The height reward function is defined as follows:

$$r\_h = \begin{cases} e^{\frac{z_r - z_0}{H_0}} & z_0 \le z_b \\ e^{\frac{z_r - z_0}{z_b}} & z_b \le z_r < z_0 \\ -\frac{1}{2} + \frac{z_r}{z_b} & z_r < z_b \end{cases} \tag{7}$$

In this form, *r_h* represents a normalized height reward that is determined by the height difference.

When the UCAV has a certain speed advantage relative to the target, it can quickly avoid the threat of the target in the course of combat, and it can also enter its superior attack zone first because it has a faster flight speed that is uses to threaten the target. There is a relationship between speed reward and speed difference, which is defined as:

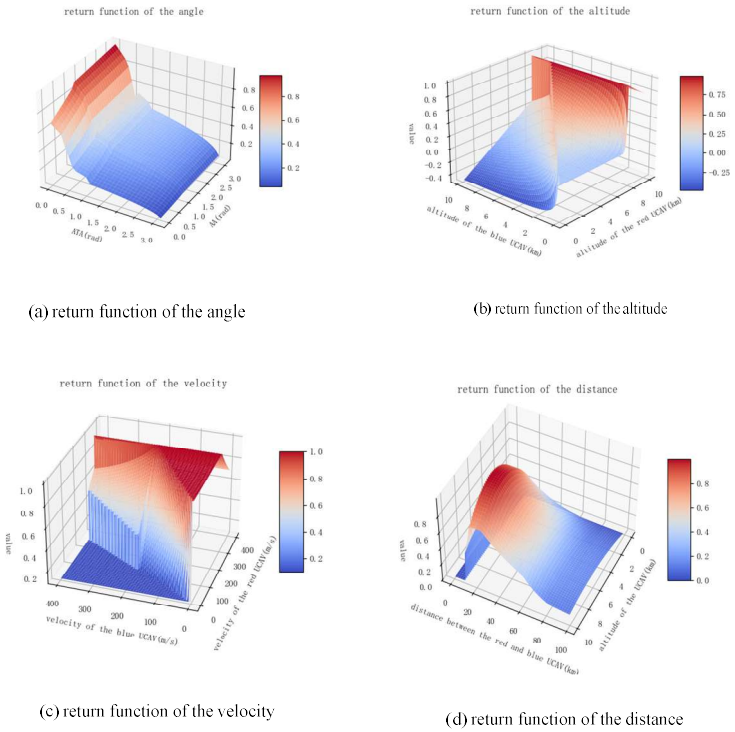
$$r\_v = \begin{cases} e^{-\frac{v_r - v_b}{v_0}} & v_0 \le v_r \\ \frac{2}{5}\left(\frac{v_r}{v_0} + \frac{v_r}{v_b}\right) & 0.6v_b < v_r < v_0 \\ 0.1 & v_P \le 0.6v_b \end{cases} \tag{8}$$

In the process of air combat, the distance between the UCAV and the target directly affects the battle result. The UCAV needs to make the target fall into its attack zone, but not into the target's attack zone. When the attack angle of the UCAV to the target is appropriate, the UCAV only needs to shorten the distance between the two UCAVs. In a fixed-point environment, the return of this distance is designed as follows:

$$r\_d = \begin{cases} (1-m^2)0.1839e^{-\frac{d-D_{\mathrm{Rmax}}}{D_{\mathrm{Rmax}}}} & D_{\mathrm{Rmax}} \le d \\ (1-m^2)0.5e^{-\frac{d-D_{\mathrm{Mmax}}}{D_{\mathrm{Rmax}}-D_{\mathrm{Mmax}}}} & D_{\mathrm{Mmax}} \le d \le D_{\mathrm{Rmax}} \\ (1-m^2)2^{-\frac{d-D_{MK\mathrm{max}}}{D_{\mathrm{Mmax}}-D_{MK\mathrm{max}}}} & D_{Mk\mathrm{max}} \le d \le D_{\mathrm{Mmax}} \\ (1-m^2) & D_{Mk\mathrm{min}} \le d \le D_{MK\mathrm{max}} \\ (1-m^2)2^{-\frac{d-D_{MK\mathrm{min}}}{10-D_{MK\mathrm{min}}}} & 10 \le d \le D_{Mk\mathrm{min}} \end{cases} \tag{9}$$

In this form, D_{Rmax}, D_{Mmax}, $D_{Mk\mathrm{max}}$, and $D_{Mk\mathrm{min}}$ are the maximum search range of the radar, the maximum attack range of the missile, the maximum distance of the missile's non-escape zone, and the minimum distance of missile's non-escape zone, respectively.

The return functions are shown in the Figure 4:



(a) return function of the angle

(b) return function of the altitude

(c) return function of the velocity

(d) return function of the distance

Figure 4. Return functions.

The comprehensive reward function is the key factor in whether the air combat strategy can converge with the optimal state, and the reward function must give the evaluation of the actions of the UCAV. The reward function R consists of a sparse reward function and a continuous reward function of the intermediate state. That is:

$$R = R_c + R_g \tag{10}$$

The continuous return function R_c consists of the angle reward, height reward, speed reward, and distance reward through a certain weight.

$$R_c = a_1 \cdot r\_a + a_2 \cdot r\_h + a_3 \cdot r\_v + a_4 \cdot r\_d \tag{11}$$

The sparse return function is related to iconic events such as drones being hit, the enemy being hit, and crashes that might occur.

$$R_g = \begin{cases} 2 & \text{Lock the enemy aircraft to launch missiles} \\ 5 & \text{Hit the enemy plane} \\ -2 & \text{Locked by enemy missiles} \\ -5 & \text{Be shot down by an enemy plane} \end{cases} \tag{12}$$

4. Algorithm Design of Air Combat Decisions That Are Based on MAPPO

According to the air combat mission and decision framework, and considering the characteristics of multi-UCAV cooperative combat, this paper designs a multi-agent proximal policy optimization algorithm for multi-UCAV air combat decision making that is based on a proximal policy optimization approach. The centralized training-distributed execution architecture is used to improve the flexibility of multi-agent, design action evaluation mechanism and corresponding network structure to improve the efficiency of the algorithm, and finally form a complete set of air combat decision-making algorithms.

4.1. Algorithm of Proximal Policy Optimization

Because air combat decision making considers the efficiency of real-time decision making and discrete instructions, there is almost no prior data or an algorithm model that can be used, however, considering the stability and convergence of the algorithm, this paper uses a near-end strategy to optimize the reinforcement learning algorithm to achieve satisfactory decision-making results.

The proximal policy optimization approach uses the method of comparing the pre- and post-strategy of the objective function, instead of modifying the learning rate like the general deep reinforcement learning algorithm does, which makes the strategy change more stable. The algorithm adopts an actor–critic (AC) framework, which includes two networks, namely, the actor network and the critic network. The update to the actor network defines the agent's objective function:

$$L^{CPI}(\theta) = \hat{E}_t\left[\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta old}(a_t|s_t)}\hat{A}_t\right] = \hat{E}_t\left[r_t(\theta)\hat{A}_t\right] \tag{13}$$

where the superscript CPI represents the conservative strategy iteration. $\hat{E}_t$ is used to find the mean at $[0, t]$. π_θ represents the strategy that is adopted by the drones in the moment. $\pi_{\theta old}$ represents the strategy that is adopted by the drone in the previous moment. a_t represents the actions that are taken by the UCAV in the moment of t. s_t indicates the status of the UCAV in the moment of t. $\hat{A}_t$ represents the dominance function of the action a_t in

the state s_t. To prevent the policy update from being too large when optimizing the agent objective function in the PPO algorithm, the agent objective function is clipped:

$$L^{CLIP}(\theta) = \hat{E}_t\left[\min\left(r_t(\theta)\hat{A}_t, clip(r_t(\theta), 1-\varepsilon, 1+\varepsilon)\hat{A}_t\right)\right]$$
$$clip(r_t(\theta), 1-\varepsilon, 1+\varepsilon) = \begin{cases} r_t(\theta) & if\ 1-\varepsilon \le r_t(\theta) \le 1+\varepsilon \\ 1-\varepsilon & if\ r_t(\theta) < 1-\varepsilon \\ 1+\varepsilon & if\ 1+\varepsilon < r_t(\theta) \end{cases} \tag{14}$$

In this form, ε is a super parameter. This method combines the gradient descent method with the trust region correction method and uses the adaptive KL penalty factor to judge whether the strategy update of the UCAV should be stopped or not to simplify the algorithm.

The critic network uses the TD-error method to train the neural network, and the update method is used to maximize the loss function. The specific formula is as follows:

$$closs = -\sum_{t=1}^{T}\left(\sum_{t'<t}\gamma^{t'-t}r_{t'} - V_\phi(s_t)\right)^2 \tag{15}$$

Among them, γ is the discount factor of the reward, r is the return of the state of the UCAV that is acting on the environment in the moment t', s_t is the state in the moment t, $V_\phi(s_t)$ is the evaluation of the state of the current moment, which is given by the critic network.

4.2. Centralized Training and Distribution Execution Strategy

In air combat, if the PPO algorithm is used to train multiple UCAVs, then the immediate reward for each UCAV is the same, so no matter what the UCAV does, the reward is the same, which leads to the phenomenon of laziness in some UCAVs. UCAVs that have made great contributions and drones that have made no contribution get the same immediate reward. The distribution of this immediate reward is unfair. It leads to the inefficiency of decision-making abilities and the algorithm training of UCAV individuals. To solve this problem, this paper forms a multi-agent, near-end policy optimization algorithm that is based on the PPO algorithm and the centralized training distributed execution architecture.

"Centralized training" refers to the training of the UCAVs by using the joint state-action value function $V_\phi(s_{r1}, s_{r2}, s_b, a_{r1}, a_{r2}, a_b)$ of the two red UCAVs and the one blue UCAV in the process of neural network training. The input of the value network in the central controller is the joint state (s_{r1}, s_{r2}, s_b) and joint action (a_{r1}, a_{r2}, a_b) of the UCAVs on the red and blue sides, thus, the joint state-action value function $V_\phi(s_{r1}, s_{r2}, s_b, a_{r1}, a_{r2}, a_b)$ is calculated, which not only takes into account the influence of the state and action of the UCAVs on the overall air combat environment, but it also takes into account the changes that are brought about by the choice of actions that are taken by the other UCAVs in different states, which solves the problem of laziness in some UCAVs in the decision making processes of multi-UCAVs cooperative air combat. This makes it possible for the ones on the red side to cooperate with the two drones to make them pursue the one on the blue side, and the one on the blue side can also confront the ones on the red side.

"Distributed execution" means that the UCAV only inputs its private observations $s = (v, \gamma, \phi, x, y, z)$ into the policy network to select its actions $a = (a_{r1}, a_{r2}, a_b)$. When the strategy is being updated, the UCAV will get a new action selection strategy $\Pi\left(s, a\middle|\theta^{\Pi}\right)$, according to the joint dominance function $R = (r_{r1}, r_{r2}, r_b)$ of the value network output. This method effectively solves the communication problem during the action selection of each UCAV, in which v, γ, and ϕ represent the speed, the track pitch angle, and the track roll angle of the UCAV, respectively. (x, y, z) represents the spatial three-axis position of the UCAV. a_{r1}, a_{r2}, and a_b represent the action that is selected by the first red UCAV 1, the selected action of the second red UCAV 2, and the selected action of the blue UCAV, respectively. r_{r1}, r_{r2}, and r_b indicate the reward that is obtained by the first red drone after

acting, the reward that is obtained by the second red drone after acting, and the reward that is obtained by the blue drone after acting, respectively.

Compared with the traditional single-agent reinforcement learning framework (using local action-value function $V_\phi(s_i, a_i)$ training to input the local state s_i and action a_i of a single UCAV), the joint action-value function inputs the global state information s and the action information $a = (a_{r1}, a_{r2}, a_b)$ of all of the entities, which is the real evaluation of the joint state strategy. The advantage of using this architecture to design an air combat decision algorithm is that all of the UCAVs share a set of network structures, and the cooperative relationship between the UCAVs can be taken into account when they are making decisions. Because the whole structure of the air combat and the generation of the return function of the UCAVs are related to the joint action, this solves the problem of the decision algorithm being difficult to converge when there are more decision subjects in the air combat.

4.3. Network Structure Design of MAPPO

Under the multi-agent reinforcement learning framework of centralized training and distributed execution, the centralized critic and distributed actor of the UCAV have different observation horizons, so the neural network structure that has shared parameters cannot be used. The calculation of the loss of the critic and the actor in the MAPPO approach should be carried out separately, and two independent networks should be used to realize the strategy and state estimations. The MAPPO approach uses a technique of centralized dominance estimation, which enables it to observe the observations and movements of all of the drones, taking full account of the state of all of the drones in the air combat environment to obtain a more accurate estimations. The TD-ERROR for the centralized dominance estimation is as follows:

$$\begin{aligned} y^j = r_i^j + \gamma Q_i^{\pi'}\left(x'^j, a_1', \ldots, a_N'\right)_{a_k'=\pi_k'(o_k^j)} \\ -V_i^{\mu'}\left(x'^j, a_1', \ldots, a_{k-1}', \ldots, a_N'\right)_{a_k'=\pi_k'(o_k^j)} \end{aligned} \tag{16}$$

The error L_t^{VE} calculation method for the critic is as follows:

$$L_t^{VE}(\theta_i) = \frac{1}{S}\sum\nolimits_j \left(y^j - A_i^\mu\left(X^j, a_1^j, \ldots, a_N^j\right)\right)^2 \tag{17}$$

where S is the batch size, and i indicates the number of the drone. The MAPPO approach can update the network and the target network to ensure that the parameters of the critic network do not diverge during the update.

The input of the artificial neural network of the critic network is the state of the UCAV, including the three-axis position $[x_i, y_i, z_i]$ and the speed v_i of the state space of the UCAV, the three-axis position $[x_a, y_a, z_a]$ and the speed v_a of the detected target state space, and the three-axis position $[x_j, y_j, z_j]$ and the speed v_j of the state space of other the UCAVs which are obtained by information sharing, wherein, the total number of input dimensions are 12. The output of the network is the value of the current state. The number of nodes in the network is (1, 64, 64, 128, 128, 256), and the optimizer that is used in the network's training is AdamOptimizer.

When updating the policy network, the MAPPO approach uses the tailored proxy objective function. Unlike PPO, MAPPO uses centralized dominance functions $\hat{A}_i^\mu\left(X^j, a_1^j, \ldots, a_N^j\right)$ to guide the policy updates. The policy update objective function of MAPPO is:

$$\begin{aligned} L^{CPI}(\theta_i) &= \hat{E}\left[\frac{\pi_{\theta_i}(a_i|X)}{\pi_{\theta_{i,old}}(a_i|X)}\hat{A}_i^\mu\left(X^j, a_1^j, \ldots, a_N^j\right)\right]_{a_k'=\pi_k'(o_k^j)} \\ &= \hat{E}_t\left[r_i(\theta)\hat{A}_i^\mu\left(X^j, a_1^j, \ldots, a_N^j\right)\right]_{a_k'=\pi_k'(o_k^j)} \end{aligned} \tag{18}$$

In the process of training, the UCAV updates the strategy network using the decision sequence that is explored in the environment and updates the target strategy network $\pi_{\theta_{i,old}}$ regularly.

The input of the artificial neural network of the actor network is the state of the UCAV, including the three-axis position $[x_i, y_i, z_i]$ and the speed v_i of the state space of the UCAV, the three-axis position $[x_a, y_a, z_a]$ and the speed v_a of the detected target state space, and the three-axis position $[x_j, y_j, z_j]$ and the speed v_j of the state space of other the UCAVs which are obtained by information sharing, wherein, a total of 12 input dimensions are obtained. The output of the network is the selection probability of seven actions in the maneuver action database. The number of nodes in the network is (7, 64, 64, 128, 128, 256), and the optimizer that is used in the network's training is AdamOptimizer.

4.4. Algorithm Implementation Flow

The MAPPO algorithm is used to train the decision-making tasks of the multi-UCAVs during air combat. The procedure flow is shown in Algorithm 1:

Algorithm 1. Specific implementation steps of the MAPPO algorithm.

1. put UCAV model, aerial combat environment, and missile model
2. initialize hyper-parameters: Training episode E, Incentive discount rate, the Learning rate of actor net work, the Learning rate of critic network, batch_size
3. **For** each episode 1,2,3, … ..E, **do**
4. **For** each UCAV, it is normalized according to current private observations $s = (v, \gamma, \phi, x, y, z)$
5. Input the normalized *s_nor* to the policy network to selection action $a_i = \pi_{\theta_i}(o_i)$, execute the action set (a_{r1}, a_{r2}, a_b) in the environment to get the state $s\_$
6. Input the s_{i-} of UCAV i and the $s_{j\_}$ of adversary UCAV j to the target selection module to obtain the primary target and secondary target
7. Input the s_{i-} of UCAV i and the $s_{j\_}$ of adversary UCAV j to the missile module, calculate whether the enemy UCAV is in the attack area of its UCAV and the missile hit rate, and obtain the discrete return r_g according to the judgment conditions such as missile hit rate, crash and stall.
8. Obtain continuous rewards according to the main and secondary targets of the UCAV and the angle reward r_a, distance reward r_d, speed reward r_v and high reward r_h of the current state. Add discrete returns and continuous returns to obtain comprehensive returns $r = r_g + r_c$.
9. When the data in the experience pool reaches the set min_ BATCH_ The normalized joint state, joint action, and joint reward of all agents are input into the value network.
10. For UCAV $i = 1, \ldots, N$
$$y^j = r_i^j + \gamma Q_i^{\pi'}\left(x'^j, a_1', \ldots, a_N'\right)_{a_k' = \pi_k'(o_k^j)} - V_i^{\mu'}\left(x'^j, a_1', \ldots, a_{k-1}', \ldots, a_N'\right)_{a_k' = \pi_k'(o_k^j)}$$
11. calculate and update the advantage estimation of each UCAV $\hat{A}_1, \hat{A}_2, \ldots, \hat{A}_n$.
12. Update the value network, and the objective function is:
$L_t^{VE}(\theta_i) = \frac{1}{S}\sum_j \left(y^j - A_i^{\mu}\left(X^j, a_1^j, \ldots, a_N^j\right)\right)^2$
13. The advantage estimation function $\hat{A}_i$ of each UCAV is input into the strategy network. Input the private state observation s_i of each UCAV in the experience pool to the strategy network.
14. Calculate the action selection strategy of the policy network.
15. Select the old action strategy and update the new action strategy.
16. The objective function of the policy network is:
$$L^{CPI}(\theta_i) = \hat{E}\left[\frac{\pi_{\theta_i}(a_i|X)}{\pi_{\theta_{i,old}}(a_i|X)}\hat{A}_i^{\mu}\left(X^j, a_1^j, \ldots, a_N^j\right)\right]_{a_k' = \pi_k'(o_k^j)} = \hat{E}_t\left[r_i(\theta)\hat{A}_i^{\mu}\left(X^j, a_1^j, \ldots, a_N^j\right)\right]_{a_k' = \pi_k'(o_k^j)}$$
17. Regularly update the target critical network.
18. Regularly update the target actor network
19. **end for**
20. **end for**

5. Intelligent Air Combat Decision Simulation Experiment

The simulation environment of the algorithm was: the official version of Intel Xeon gold 6242R CPU was the processor, with the main frequency being 3.1 GHz. The running memory was 126 GB; the graphics card was the RTX3090 24GB.

As shown in Figure 5, when the distance difference in the XY of the UCAVs between the red ones and the blue one was still 50 km, the red UCAVs and blue UCAV were equal in height. At this time, the initial position of the red UCAV 1 was [−50,000 m, 0 m, 3000 m] and the speed was 100 m/s, and the initial position of the red UCAV 2 was [−50,000 m, 20,000 m, 3000 m] and the speed was 100 m/s. The initial position of the blue UCAV was [0 m, 10,000 m, 3000 m] and the speed was 100 m/s.

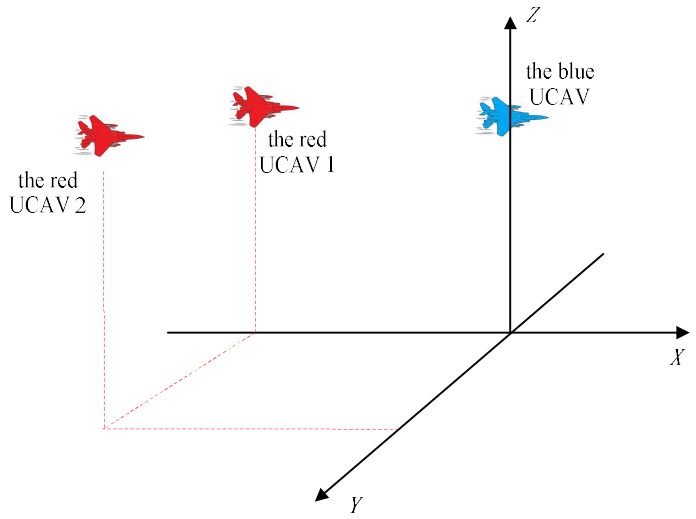


Figure 5. Initial position setting.

5.1. Training Process

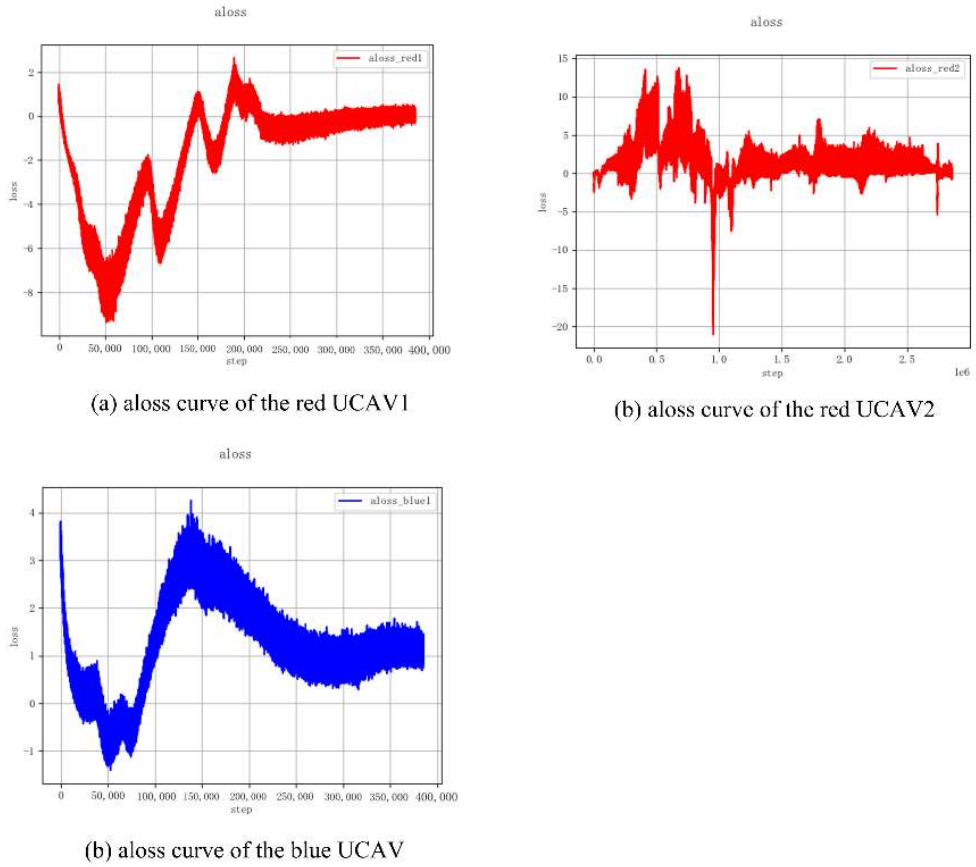
The main super parameter settings of the algorithm are shown in Table 2.

Table 2. Super parameter setting.

Parameter	Value	Parameter	Value
Training episode	1,000,000	$\varphi_{R_{\max}}$	60°
Incentive discount rate	0.9	$\varphi_{M\max}$	35°
Learning rate of actor network	0.0001	$\varphi_{Mk\max}$	20°
Learning rate of critic network	0.0001	v_0	350 m/s
Batch_size	64	z_0	8000 m
$D_{R\max}$	80km	$D_{Mk\max}$	25km
$D_{M\max}$	60km	$D_{Mk\min}$	15km

In the simulation, three identical UCAVs were used for training. To detect the training state of the algorithm and prevent the phenomenon of gradient disappearance and gradient explosion, this paper observed the loss of the actor neural network and the loss of the critical neural network, as shown in Figure 6.

As shown in Figure 6, it can be seen from the curves that the loss of the actor network (aloss) and the loss of the critic network (closs) fluctuated in the training process. With the progress of the training, the aloss and closs tended to be stable, and the artificial neural network gradually tended to be stable in the training process.



(a) aloss curve of the red UCAV1

(b) aloss curve of the red UCAV2

(b) aloss curve of the blue UCAV

Figure 6. Loss curve.

5.2. Simulation Verification

The combat situation after 1000 tests is shown in Table 3.

Table 3. Statistics of training results.

Situation	Frequency	Winning Probability
Red 1 hits blue	314	31.4%
Red 2 hits blue	388	38.8%
Blue hits Red 1	146	14.6%
Blue hits Red 2	152	15.2%

This paper takes two cases for analysis:

(1) Red 1 hits the blue UCAV.

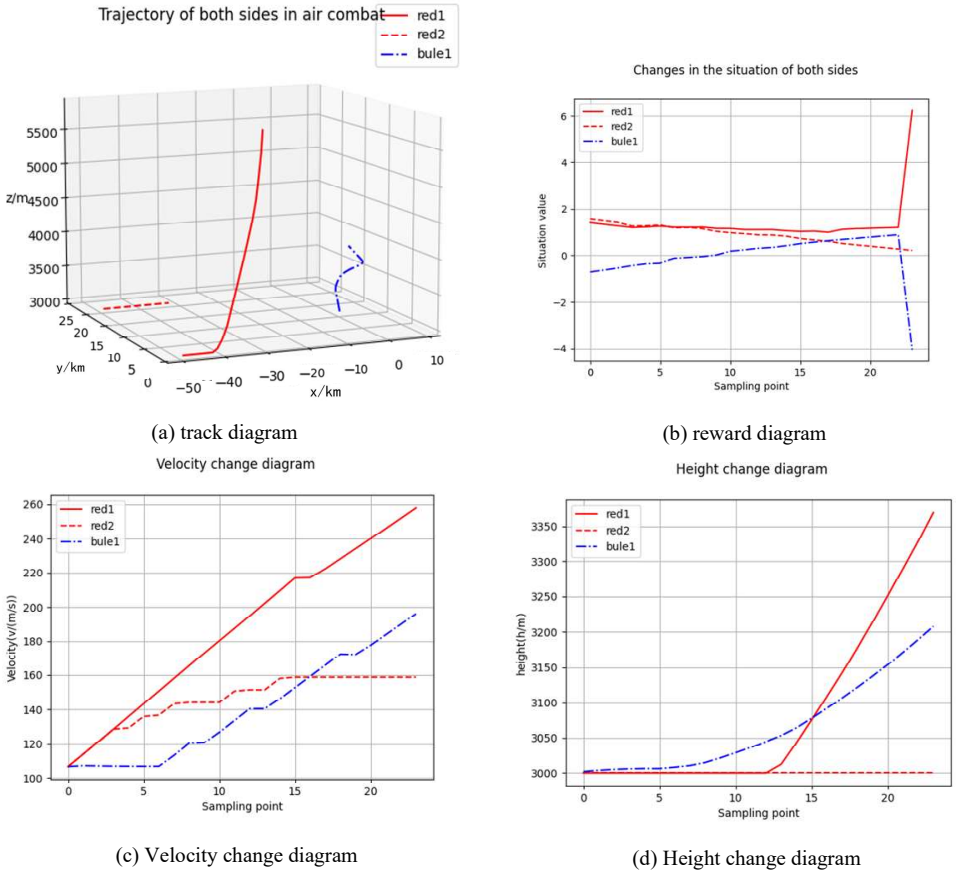
The action sequence that will be selected by red 1 is listed as follows: [‘acc’, ‘up’, ‘slow’, ‘acc’, ‘acc’, ‘up’, ‘acc’].

The action sequence that will be selected by red 2 is listed as follows: [‘acc’, ‘acc’, ‘acc’, ‘acc’, ‘fly’, ‘fly’, ‘fly’, ‘acc’, ‘fly’, ‘acc’, ‘acc’, ‘fly’, ‘fly’, ‘fly’, ‘fly’, ‘fly’, ‘fly’, ‘fly’, ‘fly’, ‘fly’, ‘fly’, ‘fly’, ‘fly’, ‘fly’, ‘fly’, ‘fly’].

The action sequence that will be selected by the blue UCAV is listed as follows: [‘acc’, ‘left’, ‘left’, ‘left’, ‘left’, ‘acc’, ‘left’, ‘left’, ‘left’, ‘up‘, ‘left’, ‘acc’, ‘left’, ‘acc’, ‘acc’, ‘ up ‘, ‘ up ‘, ‘ up ‘, ‘up’, ‘acc’, ‘acc’, ‘acc’, ‘acc’, ‘acc’, ‘acc’].

The red and blue simulation track is shown in Figure 7. The red curve represents red UCAV 1, the red dotted line represents red UCAV 2, and the blue curve represents the blue UCAV. Red 1 and red 2 will be behind the blue UCAV. According to the situation analysis, red 1 and red 2 will be superior to the blue. According to the target distribution, red 1 is

the leader and it will be responsible for the attack, and red 2 is the wingman and it will be responsible for reconnaissance and vigilance but it also will be able to launch attacks if it is necessary. According to the action sequence that is described above, red 1 first will choose to accelerate, shorten the distance between itself and the blue UCAV, and bring it into its missile attack zone, while the blue UCAV will carry out a series of actions such as acceleration, a continuous left turn, and climbing. Red 1 will also choose to accelerate and climb when it finds that the blue UCAV is maneuvering in an attempt to lock the blue one firmly to make it so that the blue UCAV is unable to escape the missile attack area of the red UCAVs. Finally, the missile of red UCAV 1 will successfully hit the blue UCAV. It can also be seen from the situation change chart in Figure 7b that although the blue one will make a series of maneuvers to gradually improve its situation when it is at a disadvantage, red 1 better will respond to the maneuvers that will be made by the blue, making it impossible for the blue UCAV to escape.



(a) track diagram

(b) reward diagram

(c) Velocity change diagram

(d) Height change diagram

Figure 7. Simulation 1 of red 1 hitting the blue UCAV.

(2) The blue UCAV hits the red 1.

The action sequence that will be selected by red 1 is listed as follows: [‘acc’, ‘acc’, ‘acc’, ‘acc’, ‘acc’, ‘acc’, ‘acc’, ‘acc’, ‘acc’, ‘acc’, ‘acc’, ‘acc’, ‘slow‘, ‘acc’, ‘slow‘, ‘acc’, ‘acc’, ‘slow‘, ‘acc’, ‘up’, ‘slow’, ‘acc’, ‘acc’, ‘up’, ‘acc’].

The action sequence that will be selected by red 2 is listed as follows: [‘acc’, ‘acc’, ‘acc’, ‘acc’, ‘fly’, ‘fly’, ‘fly’, ‘acc’, ‘fly’, ‘acc’, ‘acc’, ‘fly’, ‘fly’, ‘fly’, ‘fly’, ‘fly’, ‘fly’, ‘fly’, ‘acc’, ‘acc’, ‘acc’, ‘fly’, ‘fly’, ‘fly’].

The action sequence that will be selected by the blue UCAV is listed as follows: [‘acc’, ‘left’, ‘left’, ‘left’, ‘left’, ‘acc’, ‘left’, ‘up‘, ‘up‘, ‘up‘, ‘up‘, ‘acc’, ‘left’, ‘acc’, ‘acc’, ‘up‘, ‘up‘, ‘up‘, ‘up’, ‘acc’, ‘up‘, ‘up‘, ‘acc’, ‘up‘, ‘up‘].

The red and blue simulation track is shown in Figure 8. The initial situation of the red and blue sides will be the same as the first case. Red UCAV 1 and red UCAV 2 will cooperate to attack the blue UCAV. The blue will find itself in a disadvantageous situation and begin to maneuver to escape. Compared with the first situation, the blue one will choose to climb many times, but reduce the number of left turns that it will make, and the blue UCAV's situation will gradually become better. As red 1 blindly pursues it, the blue one will find the opportunity to attack red 1 and hit it in the process of it rising to avoid being attacked by the red UCAV. As shown in Figure 8b, the situation of the red UCAVs will be higher than that of the blue UCAV at first. With the maneuvering of the blue, the situation will turn from bad to excellent, and it will even complete the counter-kill against red 1.

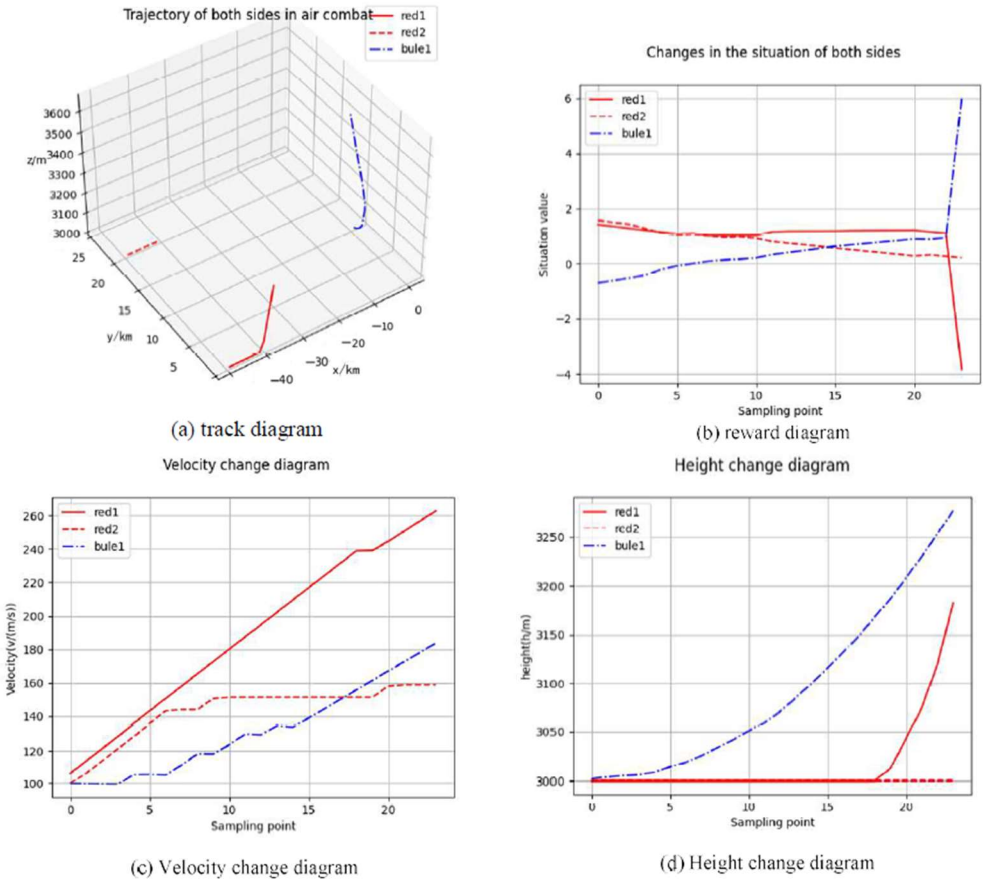


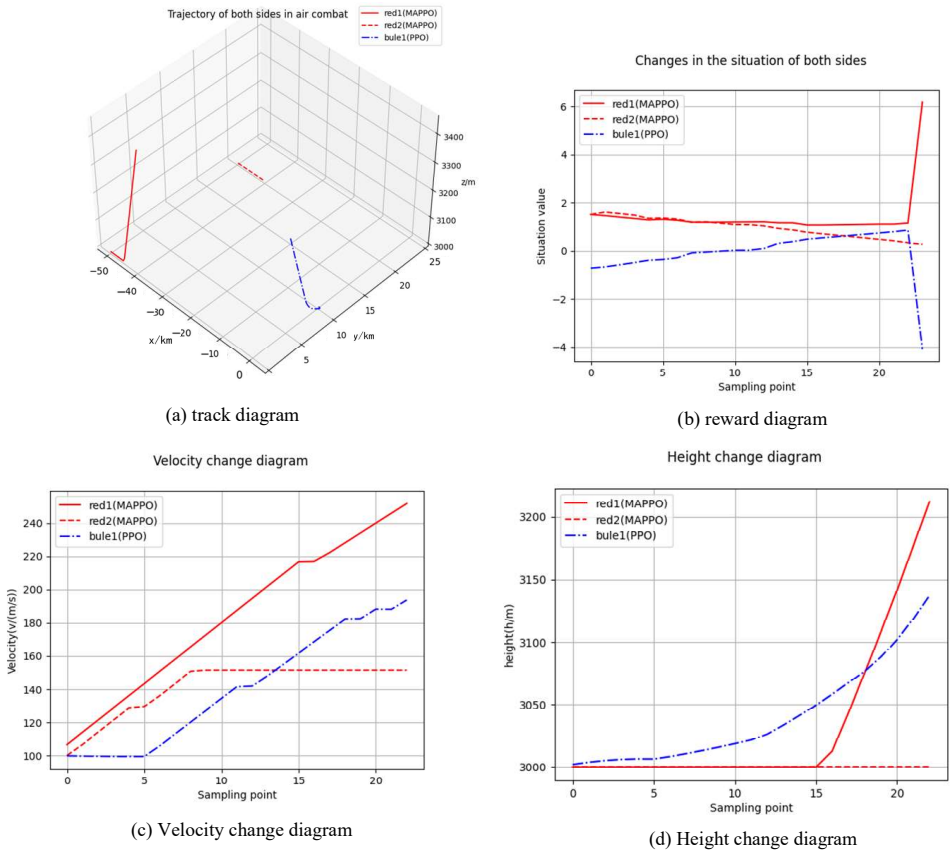
Figure 8. Simulation 2 of the blue UCAV hitting red 1.

These simulation experiments show that deep reinforcement learning can guide the UCAVs to fight, find a way to quickly expand their advantages when they have advantages, and find an appropriate attack mode when they are at an absolute disadvantage to win the air battle. The combination of deep reinforcement learning and air combat decision making has good development prospects and guiding significance.

5.3. Comparison of Algorithms

The simulations of UCAV aerial combat were performed for training using the MAPPO and PPO algorithms. The algorithm environment settings were the same as in Section 5.2. Among them, the UCAVs adopted the MAPPO algorithm for training and to generate corresponding air combat maneuvers. The target adopted the PPO algorithm for training, and the action space was the same as that of the UCAVs. Figure 7a shows the combat trajectory of the UCAV. Figure 7b–d show the reward, speed, and altitude of the UCAVs and the target.

The blue UCAV adopted the MAPPO algorithm in Section 5.2, while the blue UCAV adopted the PPO algorithm in this section. Compared with Figures 7 and 9, the blue UCAV that used the MAPPO algorithm flew faster, climbed higher, and had a higher advantage of angle. The blue UCAV's winning rate was higher, as shown in Table 4. Compared with the blue UCAV that used the MAPPO algorithm, the winning rate of the blue UCAV that used the PPO algorithm dropped from 29.8% to 14.3%.



(a) track diagram (b) reward diagram

(c) Velocity change diagram (d) Height change diagram

Figure 9. Comparison of the MAPPO and PPO algorithms.

Table 4. Statistics of results.

Situation	Frequency	Winning Probability
Red 1 hits blue	418	41.8%
Red 2 hits blue	439	43.9%
Blue hits Red 1	68	6.8%
Blue hits Red 2	75	7.5%

6. Conclusions

With the background of multi-UCAV intelligent air combat, and considering the problems of many entity types, a large decision space, and a high complexity in the multi-UCAVs' air combat procedures, this paper designs an air combat decision algorithm that is based on a multi-agent proximal policy optimization approach, designs the UCAV and missile attack area model according to the combat scenario, constructs the deep reinforcement learning network and reward function, and forms a complete set of multi-UCAV intelligent air combat decision systems.

The simulation results show that the decision system that was designed and is reported in this paper can carry out multi-UCAV air combat confrontation drills, and new tactics can be formed in the process of the drill. The intelligent air combat decision algorithm

that is based on the MAPPO approach that is proposed in this paper has the characteristics of a good input/output interface and modular rapid transplantation, a fast autonomous decision-making solution cycle, an independent of expert experience, and a decision-making mission profile that can meet the requirements of a multi-UCAV cooperative operation and a stable algorithm.

Author Contributions: Conceptualization, X.L. and Y.Y.; methodology, X.L.; software, X.L.; validation, X.L.; formal analysis, X.L.; investigation, X.L. and Y.Y.; resources, X.L. and Y.S.; data curation, X.L., and R.M.; writing—original draft preparation, X.L.; writing—review and editing, X.L.; visualization, X.L.; supervision, X.L., Y.Y., Y.S. and R.M.; project administration, X.L.; funding acquisition, X.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number No. 62073266, and the Aeronautical Science Foundation of China, grant number No. 201905053003.

Data Availability Statement: Not applicable.

Acknowledgments: Gratitude is extended to the Shaanxi Province Key Laboratory of Flight Control and Simulation Technology.

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

References

1. Yang, Z.; Sun, Z.-X.; Piao, H.-Y.; Huang, J.-C.; Zhou, D.-Y.; Ren, Z. Online hierarchical recognition method for target tactical intention in beyond-visual-range air combat. *Def. Technol.* **2022**, *18*, 1349–1361. [CrossRef]
2. Yang, Z.; Zhou, D.; Piao, H.; Zhang, K.; Kong, W.; Pan, Q. Evasive Maneuver Strategy for UCAV in Beyond-Visual-Range Air Combat Based on Hierarchical Multi-Objective Evolutionary Algorithm. *IEEE Access* **2020**, *8*, 46605–46623. [CrossRef]
3. Li, W.-H.; Shi, J.-P.; Wu, Y.-Y.; Wang, Y.-P.; Lyu, Y.-X. A Multi-UCAV cooperative occupation method based on weapon engagement zones for beyond-visual-range air combat. *Def. Technol.* **2022**, *18*, 1006–1022. [CrossRef]
4. Li, S.-Y.; Chen, M.; Wang, Y.-H.; Wu, Q.-X. Air combat decision-making of multiple UCAVs based on constraint strategy games. *Def. Technol.* **2021**, *18*, 368–383. [CrossRef]
5. Wu, H.; Zhang, J.; Wang, Z.; Lin, Y.; Li, H. Sub-AVG: Overestimation reduction for cooperative multi-agent reinforcement learning. *Neurocomputing* **2021**, *474*, 94–106. [CrossRef]
6. Garcia, E.; Casbeer, D.W.; Pachter, M. Active target defence differential game: Fast defender case. *IET Control Theory Appl.* **2017**, *11*, 2985–2993. [CrossRef]
7. Park, H.; Lee, B.-Y.; Tahk, M.-J.; Yoo, D.-W. Differential Game Based Air Combat Maneuver Generation Using Scoring Function Matrix. *Int. J. Aeronaut. Space Sci.* **2016**, *17*, 204–213. [CrossRef]
8. Ma, Y.; Wang, G.; Hu, X.; Luo, H.; Lei, X. Cooperative Occupancy Decision Making of Multi-UAV in Beyond-Visual-Range Air Combat: A Game Theory Approach. *IEEE Access* **2019**, *8*, 11624–11634. [CrossRef]
9. Han, S.-J. Analysis of Relative Combat Power with Expert System. *J. Digit. Converg.* **2016**, *14*, 143–150. [CrossRef]
10. Zhou, K.; Wei, R.; Xu, Z.; Zhang, Q. A Brain like Air Combat Learning System Inspired by Human Learning Mechanism. In Proceedings of the 2018 IEEE CSAA Guidance, Navigation and Control Conference (CGNCC), Xiamen, China, 10–12 August 2018. [CrossRef]
11. Lu, C.; Zhou, Z.; Liu, H.; Yang, H. Situation Assessment of Far-Distance Attack Air Combat Based on Mixed Dynamic Bayesian Networks. In Proceedings of the 2018 37th Chinese Control Conference (CCC), Wuhan, China, 25–27 July 2018; pp. 4569–4574. [CrossRef]
12. Fu, L.; Liu, J.; Meng, G.; Xie, F. Research on beyond Visual Range Target Allocation and Multi-Aircraft Collaborative Decision-Making. In Proceedings of the 2013 25th Chinese Control and Decision Conference (CCDC), Guiyang, China, 25–27 May 2013; pp. 586–590. [CrossRef]
13. Ernest, N.; Carroll, D.; Schumacher, C.; Clark, M.; Cohen, K.; Lee, G. Genetic Fuzzy based Artificial Intelligence for Unmanned Combat Aerial Vehicle Control in Simulated Air Combat Missions. *J. Déf. Manag.* **2016**, *6*, 1–7. [CrossRef]
14. Sathyan, A.; Ernest, N.D.; Cohen, K. An Efficient Genetic Fuzzy Approach to UAV Swarm Routing. *Unmanned Syst.* **2016**, *4*, 117–127. [CrossRef]
15. Ernest, N.D.; Garcia, E.; Casbeer, D.; Cohen, K.; Schumacher, C. Multi-agent Cooperative Decision Making using Genetic Cascading Fuzzy Systems. *AIAA Infotech. Aerosp.* **2015**. [CrossRef]
16. Crumpacker, J.B.; Robbins, M.J.; Jenkins, P.R. An approximate dynamic programming approach for solving an air combat maneuvering problem. *Expert Syst. Appl.* **2022**, *203*. [CrossRef]
17. Hernandez-Leal, P.; Kartal, B.; Taylor, M.E. A survey and critique of multiagent deep reinforcement learning. *Auton. Agents Multi-Agent Syst.* **2019**, *33*, 750–797. [CrossRef]

18. Tampuu, A.; Matiisen, T.; Kodelja, D.; Kuzovkin, I.; Korjus, K.; Aru, J.; Aru, J.; Vicente, R. Multiagent cooperation and competition with deep reinforcement learning. *PLoS ONE* **2017**, *12*, e0172395. [CrossRef] [PubMed]
19. Heuillet, A.; Couthouis, F.; Díaz-Rodríguez, N. Explainability in deep reinforcement learning. *Knowl.-Based Syst.* **2020**, *214*, 106685. [CrossRef]
20. Li, Y.-F.; Shi, J.-P.; Jiang, W.; Zhang, W.-G.; Lyu, Y.-X. Autonomous maneuver decision-making for a UCAV in short-range aerial combat based on an MS-DDQN algorithm. *Def. Technol.* **2022**, *18*, 1697–1714. [CrossRef]
21. Zhang, H.; Huang, C. Maneuver Decision-Making of Deep Learning for UCAV Thorough Azimuth Angles. *IEEE Access* **2020**, *8*, 12976–12987. [CrossRef]
22. Zhang, X.; Liu, G.; Yang, C.; Wu, J. Research on Air Combat Maneuver Decision-Making Method Based on Reinforcement Learning. *Electronics* **2018**, *7*, 279. [CrossRef]
23. Yang, Q.; Zhu, Y.; Zhang, J.; Qiao, S.; Liu, J. UAV Air Combat Autonomous Maneuver Decision Based on DDPG Algorithm. In Proceedings of the 2019 IEEE 15th International Conference on Control and Automation (ICCA), Edinburgh, UK, 16–19 July 2019; pp. 37–42. [CrossRef]
24. Piao, H.; Sun, Z.; Meng, G.; Chen, H.; Qu, B.; Lang, K.; Sun, Y.; Yang, S.; Peng, X. Beyond-Visual-Range Air Combat Tactics Auto-Generation by Reinforcement Learning. In Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 19–24 July 2020; pp. 1–8. [CrossRef]
25. Liu, P.; Ma, Y. A Deep Reinforcement Learning Based Intelligent Decision Method for UCAV Air Combat. *Commun. Comput. Inf. Sci.* **2017**, *751*, 274–286. [CrossRef]
26. Hu, D.; Yang, R.; Zuo, J.; Zhang, Z.; Wu, J.; Wang, Y. Application of Deep Reinforcement Learning in Maneuver Planning of Beyond-Visual-Range Air Combat. *IEEE Access* **2021**, *9*, 32282–32297. [CrossRef]
27. Liang, W.; Wang, J.; Bao, W.; Zhu, X.; Wu, G.; Zhang, D.; Niu, L. Neurocomputing Qauxi: Cooperative multi-agent reinforcement learning with knowledge transferred from auxiliary task. *Neurocomputing* **2022**, *504*, 163–173. [CrossRef]
28. Yao, W.; Qi, N.; Wan, N.; Liu, Y. An iterative strategy for task assignment and path planning of distributed multiple unmanned aerial vehicles. *Aerosp. Sci. Technol.* **2019**, *86*, 455–464. [CrossRef]