Article

UCAV Air Combat Maneuver Decisions Based on a Proximal Policy Optimization Algorithm with Situation Reward Shaping

Kaibiao Yang [1], Wenhan Dong [1], Ming Cai [1,*], Shengde Jia [2] and Ri Liu [3]

[1] Aviation Engineering School, Air Force Engineering University, Xi'an 710072, China
[2] College of Intelligent Sciences, National University of Defense Technology, Changsha 410073, China
[3] Theory Training Department, Harbin Air Force Flight Academy, Harbin 150009, China
* Correspondence: caim1124@163.com

Abstract: Autonomous maneuver decision by an unmanned combat air vehicle (UCAV) is a critical part of air combat that requires both flight safety and tactical maneuvering. In this paper, an unmanned combat air vehicle air combat maneuver decision method based on a proximal policy optimization algorithm (PPO) is proposed. Firstly, a motion model of the unmanned combat air vehicle and a situation assessment model of air combat was established to describe the motion situation of the unmanned combat air vehicle. An enemy maneuver policy based on a situation assessment with a greedy algorithm was also proposed for air combat confrontation, which aimed to verify the performance of the proximal policy optimization algorithm. Then, an action space based on a basic maneuver library and a state observation space of the proximal policy optimization algorithm were constructed, and a reward function with situation reward shaping was designed for accelerating the convergence rate. Finally, a simulation of air combat confrontation was carried out, which showed that the agent using the proximal policy optimization algorithm learned to combine a series of basic maneuvers, such as diving, climb and circling, into tactical maneuvers and eventually defeated the enemy. The winning rate of the proximal policy optimization algorithm reached 62%, and the corresponding losing rate was only 11%.

Keywords: air combat; maneuver decision; deep reinforcement learning; reward shaping

check for updates

Citation: Yang, K.; Dong, W.; Cai, M.; Jia, S.; Liu, R. UCAV Air Combat Maneuver Decisions Based on a Proximal Policy Optimization Algorithm with Situation Reward Shaping. *Electronics* **2022**, *11*, 2602. https://doi.org/10.3390/electronics11162602

Academic Editor: Hamid Reza Karimi

Received: 24 July 2022
Accepted: 17 August 2022
Published: 19 August 2022

1. Introduction

As a critical part of modern war, air combat intensively embodies the trend that emerging technologies are widely applied in war. Autonomous decision by an unmanned combat air vehicle (UCAV) is one of the main research directions in air combat. In the 1960s, the National Aeronautics and Space Administration (NASA) began an experiment using artificial intelligence (AI) to replace human pilots in making air combat decisions [1,2]. In 2016, Psibernetix Inc., Air Force Research Laboratory and University of Cincinnati, disclosed an AI called ALPHA, which controls the flights of UCAVs in air combat missions [3]. In 2019, the Defense Advanced Research Project Agency (DARPA) kicked off the Air Combat Evolution program (ACE), which is aimed at developing trusted, scalable, human-level, AI-driven autonomy for air combat [4].

The main advantages of UCAV autonomous air combat are as follows: improving the combat level of the pilot by man–machine confrontation training; easing the operational pressure of the pilot with decision support in air combat; reducing training time, costs and casualties by autonomous air combat or human–machine collaborative combat.

The theory and technology regarding autonomous air combat maneuver decision have been evolving in recent years. Traditional research methods include game theory, expert system and optimization theory.

Differential game theory is a typical representative air combat maneuver decision method based on game theory. Differential game of pursuit and evasion was introduced by Isaacs [5]. Sufficient conditions of qualitative differential game, which assure the termination of a particular target of a pursuit–evasion problem, were derived by Leitmann [6]. A differential game theory-based maneuver algorithm following a hierarchical decision structure was developed in [7]. The algorithm establishes a scoring function matrix computed from the relative geometry, relative distance and velocity of two aircrafts, and then it performs a scoring function matrix calculation based on differential game theory to determine the optimal maneuvers in a dynamic and challenging combat situation. The combat results and maneuvers under the controlled computation environment are similar to those of real air combat.

A rule-based expert system, which needs less effort to engineer and possesses high interpretability, constructs production rules with predicate logic similar to IF–ELSE–THEN. An air combat decision expert system was established in [8]. The system matches collected air combat situation information with the rules in an expert knowledge base, and then it outputs corresponding strategies. A systematic decision method based on an expert system was proposed in [9]. The method designs a maneuver selection auxiliary system, which is composed of a knowledge base, tactical planner, maneuver method selector, execution planner, special case processor and man–machine interface. The knowledge base is constructed according to expert evaluation of air combat decision. The other parts of the system are mainly constructed by fuzzy logic. This method can assist pilots in realizing the potential combat capability of the aircraft and meet the operational performance requirements within time limits. In [10], a rule base was established according to potential air combat situations faced by UCAVs, and the constraints of air combat decision were set with a priori knowledge. In [11], an evolutionary expert system tree method was proposed to solve the problem that traditional expert systems are unable to be applied in unexpected situations.

An air combat decision method based on an optimization algorithm formalizes decision-making problems into multiobjective optimization problems. In [12], a Bayesian inference-theory-based air combat situation assessment was calculated to adjust the weights of the maneuver decision factors. Functions based on fuzzy logic of the maneuver decision factors were established to enhance the robustness and effectiveness of the decision results. The simulation results demonstrated the intensity of the air combat game and displayed the drawbacks of an element maneuver method. An intelligent tactical aid decision system based on linear discriminant analysis (LDA) and improved glowworm swarm optimization (IGSO) was proposed in [13]. The simulation results showed that the LDA-IGSO-ELM algorithm can quickly and accurately solve the problem of threat assessment and provide effective support for firepower allocation and maneuver decision. The biological immune system protects the body against intruders by recognizing and destroying harmful cells or molecules. Inspired by this, [14] employed the genetic algorithm and evolutionary algorithm to imitate the immune mechanism to select and construct air combat maneuvers. The test results demonstrated the potential of immunized maneuver selection in terms of constructing effective motion-based trajectories over a relatively short (1–2 s) period of time. Reference [15] modeled air combat based on optimal control and game theory and then transformed dual aircraft air combat into an optimization problem using parameterized curves. The problem was then solved with a moving time horizon scheme to generate optimal strategies for the aircraft. The results from different scenarios showed that the method is capable of providing solutions in air combat for autonomous aerial vehicles.

Traditional air combat decision methods require accurate modeling of a complex and changeable air combat process, which is usually difficult to achieve. With the expansion of the decision space, it is also faced with dimension explosion, which leads to difficulties in solving problems and low real-time performance. Emerging research methods represented by deep reinforcement learning (DRL) train an agent through a trial-and-error mechanism, which removes accurate modeling and generates maneuver decision with data based on a neural network to improve solution efficiency [16].

The feasibility of autonomous maneuver decision based on reinforcement learning (RL) was proved in [17]. However, motion of UCAVs was only considered in the 2D plane, and research on the motion of UCAVs in the 3D space was lacking in the paper. An intelligent tactical decision method based on a deep Q-learning network (DQN) was proposed in [18]. In order to verify the validity of the method, an enemy based on Min-Max algorithm was also designed. The simulation results showed that the decision-making performance of the DQN approach was quicker and more effective than the Min-Max recursive approach. However, velocity was not considered as a feature vector in the input of the neural network, which may have affected the convergence rate of the method. In [19], a heuristic Q-network method was proposed to improve the efficiency of reinforcement learning, which realizes self-learning of air confrontation maneuver strategy and, finally, helps the fighters make reasonable maneuver decisions independently under different air confrontation situations. However, similar to [17], the influence of changes in altitude on the motion of UCAVs was not considered. In [20], an autonomous maneuver decision model based on a deep Q network was proposed for UCAVs in short-range air combat; however, velocity and angles were coupled in the definition of an air combat situational advantage, which may interfere in the process of learning optimal strategies with a reward function. A novel maneuver strategy generation algorithm based on a state-adversarial deep deterministic policy gradient algorithm was proposed in [21], but the experimental results in the paper only proved the effectiveness of the algorithm in the 2D plane.

There are two main inadequacies in the existing research on air combat maneuver decision based on deep reinforcement learning: one is that the training environment of the agent is different from actual air combat, which limits the application of the research results; the other is that factors affecting air combat are not fully considered, which affects the learning efficiency of the agent. In this paper, a novel UCAV air combat maneuver decision method based on a proximal policy optimization algorithm (PPO) [22] is proposed. Firstly, a general UAV short-range air combat confrontation framework was established including a motion model of a fixed-wing aircraft and a situation assessment model of air combat. In order to verify the effectiveness of the method, an enemy maneuver strategy based on a greedy algorithm was designed. Then, the action space based on a basic maneuver library and the state observation space of the agent were constructed. The maneuver policy was generated by training with the PPO algorithm. A reward function, based on situation reward shaping, was designed for accelerating the convergence rate of the algorithm. Finally, simulations of one-to-one, short-range air combat in stochastic situations were carried out to verify the effectiveness of the PPO algorithm. The winning rate of the PPO algorithm reached 62%, and the corresponding losing rate was only 11%. The results of the simulations showed that the maneuver decision method proposed in this paper can enable UCAVs to learn the maneuver policy autonomously and defeat the enemy.

The paper is arranged as follows: the general UCAV short-range air combat confrontation framework is constructed in Section 2; the agent training method based on deep reinforcement learning is designed in Section 3; the simulation analysis is performed in Section 4; Section 5 summarizes the research.

2. Air Combat Confrontation Framework

In this section, the simulation environment of general UCAV short-range air combat is constructed, which mainly included a UCAV motion model, air combat situation assessment model, and enemy maneuver strategy based on the situation assessment. The UCAV motion model provides a research object for air combat maneuver decision. The air combat situation assessment model is the basis for evaluating the advantages and disadvantages of the air combat maneuver decision, and the enemy maneuver strategy based on the situation assessment adds antagonism to the air combat simulation environment and serves as a training opponent to verify the effectiveness of the air combat maneuver decision based on deep reinforcement learning.

2.1. Unmanned Combat Air Vehicle Motion Model

The velocities and positions of UCAVs are the main factors that affect short-range air combat. Therefore, a three degrees of freedom model for a fixed-wing aircraft was employed in this paper, which not only reflected the air combat situation but also improved the simulation's efficiency. Considering that the angle of attack and the angle of sideslip are so small that the two angles can be ignored, a dynamic model of a UCAV in the NED (north-east-down) coordinate system can be expressed as [23]:

$$\begin{cases} \dot{x} = v\cos\gamma\cos\psi \\ \dot{y} = v\cos\gamma\sin\psi \\ \dot{z} = -v\sin\gamma \\ \dot{v} = g(n_x - \sin\gamma) \\ \dot{\gamma} = \frac{g}{v}(n_z\cos\phi - \cos\gamma) \\ \dot{\psi} = \frac{g\sin\phi}{v\cos\gamma}n_z \end{cases}, \tag{1}$$

where $\boldsymbol{p} = [x, y, z]$ represents the position, and the flight altitude of the UCAV is $h = -z$. $\dot{x}$, $\dot{y}$ and $\dot{z}$ denote the north velocity, the east velocity and the vertical velocity, respectively; v represents the airspeed; γ and ψ denote the angle of climb and the azimuth angle of the flight path; respectively. The state variable $\boldsymbol{s}$ is:

$$\boldsymbol{s} = \left[x, y, z, v, \gamma, \psi, \dot{x}, \dot{y}, \dot{z}, \dot{v}, \dot{\gamma}, \dot{\psi}\right]. \tag{2}$$

where $\boldsymbol{u} = [n_x, n_z, \phi]$ is the control input of a UCAV; n_x is the longitudinal overload, and its direction is consistent with the direction of the velocity vector; n_z is the normal overload, and its direction is consistent with the direction of the lift; ϕ is the roll angle with the velocity vector as the axis, and its definition conforms to the right-hand rule. The velocity magnitude can be changed by n_x, and the velocity direction can be changed by n_z and ϕ. Therefore, the desired maneuvers of a UCAV can be achieved with an appropriate $\boldsymbol{u}$ in theory. Considering the real conditions of the short-range air combat environment and the limited maneuverability of a UCAV, several limitations were set, $[x, y] \in \left[-2\times10^4, 2\times10^4\right]$ m and $z \in \left[-2\times10^4, -1\times10^2\right]$ m; $v \in [100, 600]$ m/s; $n_x \in [-5, 5]$ and $n_z \in [-5, 5]$. The parameters of the UCAV motion model are shown in Figure 1.

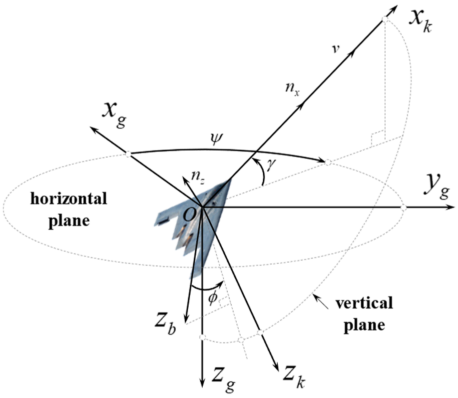


Figure 1. UCAV motion model.

2.2. Air Combat Situation Assessment Model

An air combat situation can be understood as qualitative or quantitative descriptions of various indicators of UCAVs including the position, velocity, firepower and relative position and velocity difference between UCAVs. The air combat situation is dynamic so that the descriptions in different evaluation systems may be different.

The geometry between two UCAVs can be determined by the relative angle and distance in the short-range air combat. The geometry of air combat is shown in Figure 2.

The red aircraft represents the UCAV trained by deep reinforcement learning, and the blue aircraft represents the enemy UCAV. $\boldsymbol{LOS}$ is line of sight, which is defined as the vector from the center of mass of the red UCAV to that of the blue UCAV, and its magnitude is the distance between the two UCAVs. ATA is the antenna train angle, which is defined as the angle between the velocity vector of the red UCAV and the $\boldsymbol{LOS}$, $ATA \in [0, \pi]$. AA is the aspect angle, which is defined as the angle between the velocity vector of the blue UCAV and the $\boldsymbol{LOS}$, $AA \in [0, \pi]$. HCA is the heading crossing angle, which is defined as the angle between the velocity vectors of the two UCAVs, $HCA \in [0, \pi]$. The three-ninths line is the line between the 3 and 9 o'clock directions of an aircraft, which is used to determine whether the other aircraft is located in the front or rear hemisphere of the aircraft.

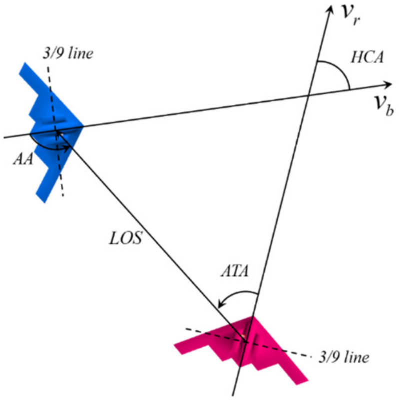


Figure 2. Geometry of air combat.

Let $\boldsymbol{p}_r = [x_r, y_r, z_r]$ and $\boldsymbol{v}_r = \left[v_r^x, v_r^y, v_r^z\right]$ denote the position and the velocity vector of the red UCAV, respectively, and $\boldsymbol{p}_b = [x_b, y_b, z_b]$ and $\boldsymbol{v}_b = \left[v_b^x, v_b^y, v_b^z\right]$ denote the position and the velocity vector of the blue UCAV, respectively. Then, the formula using the above parameters is as follows:

$$\boldsymbol{LOS} = \boldsymbol{p}_b - \boldsymbol{p}_r, \tag{3}$$

$$ATA = \arccos\left(\frac{\boldsymbol{v}_r \cdot \boldsymbol{LOS}}{\|\boldsymbol{v}_r\| \cdot \|\boldsymbol{LOS}\|}\right), \tag{4}$$

$$AA = \arccos\left(\frac{\boldsymbol{v}_b \cdot \boldsymbol{LOS}}{\|\boldsymbol{v}_b\| \cdot \|\boldsymbol{LOS}\|}\right), \tag{5}$$

$$HCA = \arccos\left(\frac{\boldsymbol{v}_r \cdot \boldsymbol{v}_b}{\|\boldsymbol{v}_r\| \cdot \|\boldsymbol{v}_b\|}\right). \tag{6}$$

The situation assessment was divided into two parts: one was the conditions for the end of air combat including the attack zone, altitude limit and velocity limit; the other part was the situation assessment in the process of air combat including the angle, velocity and altitude situations.

2.2.1. Conditions for the End of Air Combat

Short-range air combat, also known as a dogfight, usually covers an operational range of 10 km, and the aim is to shoot down enemy aircraft or make them lose their combat capability. Therefore, whether the air combat is over can be judged by the following conditions:

1. Shooting down the enemy or being shot down;
2. Stalling or crashing.

The attack zone of a UCAV is limited due to the impact of the types and performances of weapons. In this paper, the attack zone was simplified to $\mathbf{Z}_\alpha$, which is the zone surrounded by red lines in Figure 3. α denotes the maximum attack angle of a UCAV, $\alpha \geq 0$. $R_{\min}$ denotes the minimum attack range, and $R_{\max}$ denotes the maximum attack range. If

$ATA < \alpha$ and the distance of two UCAVs is greater than $R_{\min}$ and less than $R_{\max}$, the blue UCAV is in the attack zone of the red UCAV, so that the red UCAV will fire to shoot down the blue UCAV. On the contrary, the red UCAV will be shot down. The attack zone, $\mathbf{Z}_\alpha$, can be expressed as:

$$\mathbf{Z}_\alpha = \{\boldsymbol{p}_b | ATA \leq \alpha, R_{\min} \leq \|\boldsymbol{LOS}\| \leq R_{\max}\}. \tag{7}$$

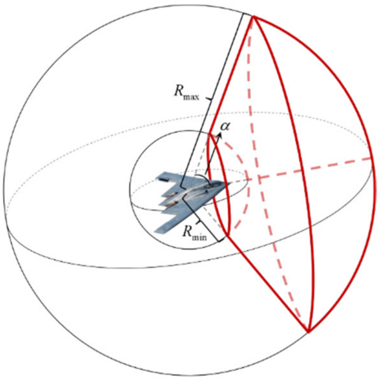


Figure 3. Attack zone of a UCAV.

Aircraft with an extremely low velocity or altitude are not allowed in air combat. Therefore, the other condition for judging whether the air combat is over can be expressed as:

$$\begin{cases} \text{end of air combat}, & \text{if } v \leq v_{\min} \text{ or } h \leq h_{\min}, \\ \text{air combat in progress}, & \text{else}, \end{cases} \tag{8}$$

where $v_{\min}$ and $h_{\min}$ denote the minimum allowable velocity and altitude of a UCAV, respectively.

In addition, it was necessary to limit the number of steps per episode in the simulation. Therefore, another condition for judging whether the air combat is over can be expressed as:

$$\begin{cases} \text{end of air combat}, & \text{if } t \geq t_{\max}, \\ \text{air combat in progress}, & \text{else}, \end{cases} \tag{9}$$

where $t_{\max}$ denotes the maximum steps per episode.

2.2.2. Situation Assessment in the Process of Air Combat

This section takes the red UCAV as an example. In essence, the whole process of air combat can be understood as that the red UCAV avoids entering the attack zone of the blue UCAV and makes the blue UCAV enter that of the red UCAV by maneuvering. Figure 4 shows a top view of the attack zone. The red and the green parts represent the attack zone and the security zone, respectively.

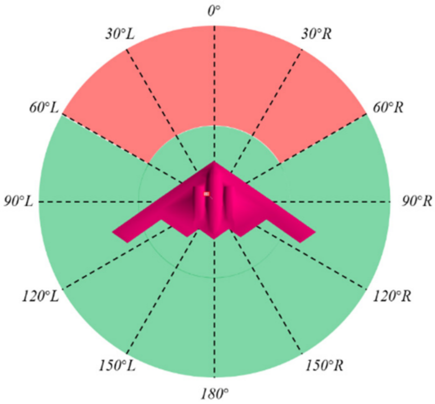


Figure 4. Top view of the attack zone.

According to Figures 2 and 4, the smaller the ATA and AA, the more favorable it is for the red UCAV in the same other conditions. In addition, flight safety is also a key factor affecting the situation according to Section 2.2.1. Therefore, the air combat situation, η, can be divided into four parts, which can be switched according to the state of the UCAV. The switching process is as follows (Algorithm 1).

Algorithm 1 Switching Process of an Air Combat Situation

```
If $v$ or $h$ are less than the threshold
    $\eta = \eta_1(v, h)$
    Else if $AA < 90^\circ$
        $\eta = \eta_2(\|\boldsymbol{LOS}\|)$
        Else if distance is greater than the threshold
            $\eta = \eta_3(AA)$
            Else
                $\eta = \eta_4(\|\boldsymbol{LOS}\|)$
```

The first part of the situation model η_1 takes the velocity and altitude of the UCAV as variables. When the velocity or altitude is less than the threshold, η_1 takes effect, which is to ensure flight safety. The velocity index and altitude index of η_1 are as follows:

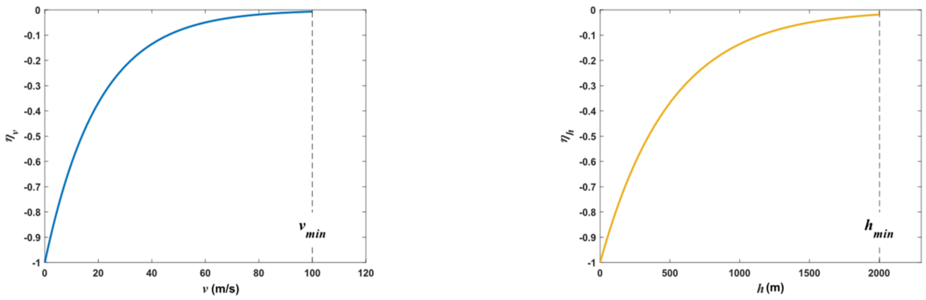
$$\eta_v = -e^{-v/k_v}, \tag{10}$$

$$\eta_h = -e^{-h/k_h}, \tag{11}$$

η_1 is defined as:

$$\eta_1 = \eta_v + \eta_h - 2, \tag{12}$$

where k_v and k_h are constants used to adjust the trend of the index curves, $\eta_1 \in [-4, -2)$, and η_v and η_h are shown in Figure 5.



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

Figure 5. (**a**) Velocity index; (**b**) altitude index.

The second part of the situation model, η_2, takes the distance between UCAVs as a variable. When the UCAV flies at a normal velocity and altitude, $AA \le \frac{\pi}{2}$, and η_2 takes effect, which aims to guide the UCAV to quickly approach the enemy and reach attack conditions. η_2 is defined as:

$$\eta_2 = 0.5\left(e^{-\|\boldsymbol{LOS}\|/k_d} - 1\right), \tag{13}$$

where k_d is a constant used to adjust the trend of η_2, $\eta_2 \in (-0.5, 0]$. The change in η_2 with the distance between two the UCAVs is shown in Figure 6.

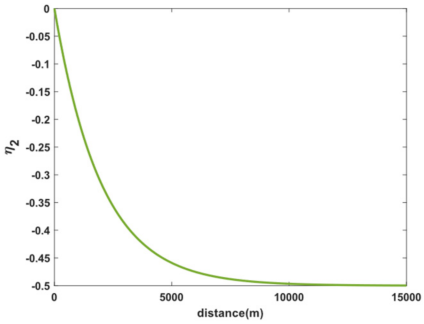


Figure 6. Change in η_2 with the distance between two UCAVs.

The third part of the situation model, η_3, takes AA as a variable. When the UCAV flies at a normal velocity and altitude, $AA > \frac{\pi}{2}$, and the distance between the two UCAVs is greater than the safety threshold, meaning η_3 takes effect, the aim of which is to lay the foundation for attack conditions for the UCAV. η_3 is defined as:

$$\eta_3 = -\frac{AA}{\pi}, \ AA \in [0.5\pi, \pi], \tag{14}$$

$\eta_3 \in [-1, -0.5]$. The change in η_3 with AA is shown in Figure 7.

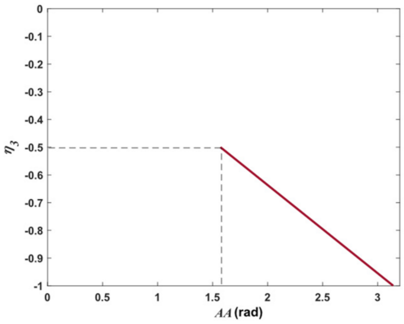


Figure 7. Change in η_3 with AA.

The fourth part of the situation model, η_4, takes the distance between the UCAVs as a variable. When the UCAV flies at a normal velocity and altitude, $AA > \frac{\pi}{2}$, and the distance between the two UCAVs is less than the safety threshold, meaning η_4 takes effect, which is aimed at guiding the UAV to safer states so that the enemy loses the attack conditions. η_4 is defined as:

$$\eta_4 = -\left(\frac{\|\boldsymbol{LOS}\|}{d_{\text{threshold}}} - 1\right)^2 - 1, \tag{15}$$

where $d_{\text{threshold}}$ is a constant, $\eta_4 \in [-2, -1]$. The change in η_4 with the distance between two the UCAVs is shown in Figure 8.

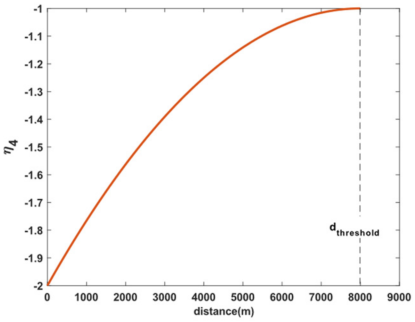
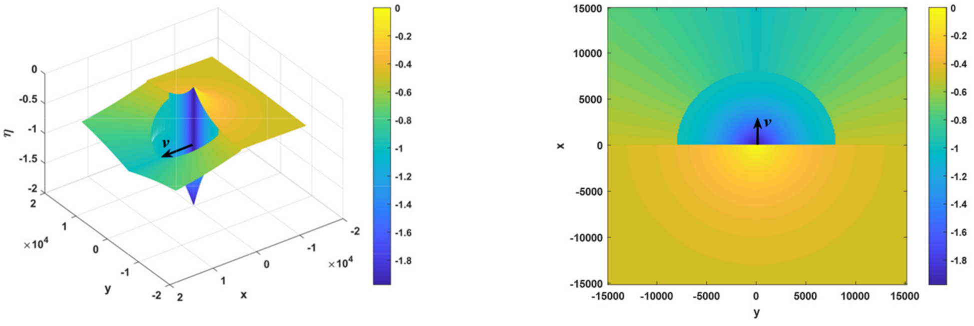


Figure 8. Change in η_4 with the distance between two UCAVs.

Let $k_v = 20$, $k_h = 500$, $k_d = 2000$ and $d_{\text{threshold}} = 8000$ m. According to the velocity of the enemy, $\boldsymbol{v}$, the air combat situation, η, at the same altitude can be drawn with the enemy as the origin as shown in Figure 9.



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

Figure 9. Air combat situation at the same altitude: (**a**) stereoscopic view of the air combat situation at the same altitude; (**b**) plan view of the air combat situation at the same altitude.

It can be seen from Figure 9 that in a position a high air combat situation value, it is difficult for the enemy to form attack conditions, while the UCAV is relatively safe and with conditions that are conducive to the formation of attack conditions against the enemy.

2.3. Enemy Maneuver Policy

A real air combat environment has high-intensity antagonism; that is, the enemy will attack with all its strength. In order to verify effectiveness of the deep reinforcement learning, an enemy maneuver strategy based on a greedy algorithm is designed in this section. Let $\boldsymbol{s}_r$ and $\boldsymbol{s}_b$ denote the states of the red and the blue UCAVs, respectively, at each decision step, t. Firstly, calculating each air combat situation after the blue UCAV executes each available control input, $\boldsymbol{u}_i$, separately, and recording these situations as $\{\eta_i\}$, $i = 1, 2, \cdots$. Then, the $\boldsymbol{u}_i$ corresponding to the maximum η_i as the actual control input of the blue UCAV to maneuver is taken. The maneuver strategy $\pi_{greedy}(\boldsymbol{s}_r, \boldsymbol{s}_a)$ based on a greedy algorithm is defined as:

$$\pi_{greedy}(\boldsymbol{s}_r, \boldsymbol{s}_a) = \operatorname*{argmax}_{\boldsymbol{u}_i}\{\eta_i\}. \tag{16}$$

In order to meet the real-time requirements of air combat confrontation and to improve the exploration efficiency of the greedy algorithm, it was necessary to limit the exploration space of the available control input, $\boldsymbol{u}$. Complex maneuvers can be decomposed into a series of simple basic maneuvers. NASA designed 7 elemental maneuvers, that realize the complex maneuvers of aircraft in 3D space [24]. Therefore, a 13-dimensional basic maneuver library was designed, which is shown in Table 1. Each action, $\boldsymbol{a}_i$, in the library corresponds to a set of control input $\boldsymbol{u}$, $i = 1, 2, \cdots, 13$.

Table 1. Basic maneuver library.

Action	n_x	n_z	ϕ
a_1	5	1	0
a_2	0	1	0
a_3	-5	1	0
a_4	5	5	0
a_5	0	5	0
a_6	-5	5	0
a_7	5	5	π
a_8	0	5	π
a_9	-5	5	π
a_{10}	0	5	$\arccos(1/5)$
a_{11}	-5	5	$\arccos(1/5)$
a_{12}	0	5	$-\arccos(1/5)$
a_{13}	-5	5	$-\arccos(1/5)$

The effectiveness of the maneuver strategy based on a greedy algorithm was verified in the simulation in Section 4.

3. Maneuver Decision Method Based on a Proximal Policy Optimization Algorithm

3.1. Proximal Policy Optimization Algorithm

Reinforcement learning is a branch of machine learning [25]. Different from supervised learning and unsupervised learning, the agent in reinforcement learning learns strategy by interaction with the environment. The general model of reinforcement learning is the Markov decision process (MDP), which is shown in Figure 10. The model is composed of an environment, agent, state, reward and action. The environment includes the motion law of the object and some restrictions. It receives the action and updates the state and feeds back the reward to the agent. The agent is the carrier of reinforcement learning algorithm, which outputs the action according to the state of the environment and the reward.

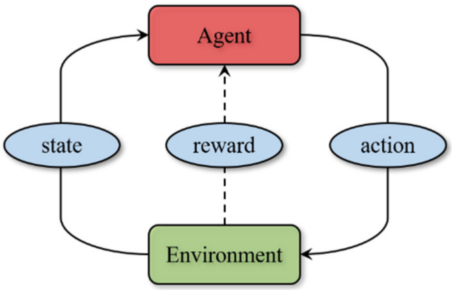


Figure 10. General model of reinforcement learning.

At each step, the agent executes action a_t according to the policy π, and then the state of the agent updates from s_t to s_{t+1} with the state transition probability $p(s_{t+1}|s_t, a_t)$. The environment feedbacks reward $r(s_t, a_t)$ to the agent at the same time. The goal of reinforcement learning is to determine the optimal strategy π^* to maximize the expected discount cumulative return, $V^\pi(s)$, called the value function [25], which is defined as:

$$V^\pi(s) = \mathbb{E}_{s_t,a_t\sim\pi}\left[\sum_{t=0}^{\infty}\tau^t r(s_t, a_t)\right], \tag{17}$$

where $\tau \in (0,1)$ denotes the discount factor. Therefore, the optimal value function, $V^*(s)$ (i.e., the maximum value of the expected discount cumulative return), is defined as:

$$V^*(s) = \max\mathbb{E}_{s_t,a_t\sim\pi^*}\left[\sum_{t=0}^{\infty}\tau^t r(s_t, a_t)\right], \tag{18}$$

Deep reinforcement learning is a combination of deep learning and reinforcement learning, which uses the parameter θ of neural networks to store and update the policy π_θ of reinforcement learning. It has an edge in solving the problems with high task complexity, high dimension of solution space and redundant state information.

The proximal policy optimization algorithm (PPO) is an on-policy algorithm based on policy gradient methods [26] for deep reinforcement learning. Policy gradient methods work by computing an estimator of the policy gradient and plugging it into a stochastic gradient ascent algorithm. The loss function of a PPO is defined as [22]:

$$L_t^{CLIP+VF+S}(\theta) = \hat{\mathbb{E}}_t\Big[L_t^{CLIP}(\theta) - c_1 L_t^{VF}(\theta) + c_2 S[\pi_\theta](s_t)\Big], \tag{19}$$

where c_1 and c_2 are coefficients; $L^{VF}(\theta)$ is the value function of the squared-error loss; $S[\pi_\theta]$ denotes the entropy bonus of policy π_θ [27,28]. The surrogate objective $L^{CLIP}(\theta)$ is the core of the PPO algorithm, which is defined as [22]:

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t\big[\min(r_t(\theta)\hat{A}_t, \operatorname{clip}(r_t(\theta), 1-\varepsilon, 1+\varepsilon)\hat{A}_t)\big], \tag{20}$$

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}, \tag{21}$$

where ε denotes the clip range, which is used to clip the probability ratio $r_t(\theta)$ to avoid an excessively large policy update; $\hat{A}_t$ is an estimator of the advantage function, which can be calculated by generalized advantage estimation (GAE) [29].

$$\hat{A}_t = \sum_{l=0}^{\infty} (\tau\lambda)^l \delta_{t+l}^V, \tag{22}$$

where λ is the GAE factor, and δ_t^V is defined as [29]:

$$\delta_t^V = r_t + \tau V(s_{t+1}) - V(s_t), \tag{23}$$

In the training process, the PPO algorithm updates the parameter θ of the neural network by optimizing the loss function $L_t^{CLIP+VF+S}(\theta)$ to achieve the purpose of updating the policy.

3.2. Action Space

The action space is divided into discrete space and continuous space. The discrete action space, $\boldsymbol{A}_1$, can be directly composed of the basic maneuver library in Table 1, which is expressed as $\boldsymbol{A}_1 = [\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_{13}]$. The discrete action space, based on the maneuver library, is equivalent to providing priori experience for the agent. Compared with the discrete action space, the continuous action space, $\boldsymbol{A}_2$, can make better use of the maneuverability of a UCAV, but it may increase the difficulty of flight control and reduce the learning efficiency. Since the actions of reinforcement learning generally need to be normalized, it was necessary to linearly map the control input $\boldsymbol{u}$. Therefore, the continuous action space $\boldsymbol{A}_2$ can be expressed as:

$$\boldsymbol{A}_2 = \big\{\boldsymbol{a}\big|\boldsymbol{a} = f(\boldsymbol{u}), \boldsymbol{a}_j \in [-1,1],\ j = 1,2,3\big\}, \tag{24}$$

where $f(\cdot)$ denotes linear mapping; $\boldsymbol{a}_j$ denotes the jth dimension element of action $\boldsymbol{a}$.

A comparative experiment between the discrete action space and the continuous action space is carried out in Section 4.

3.3. State Observation Space

The convergence of the algorithm is affected by the state observation space. Design of the state observation space strives to be concise and efficient. On the one hand, incompleteness of information in the state observation space may make the algorithm difficult to converge. On the other hand, redundant information in the state observation may cause overfitting. In addition, refining the original state information properly and designing the information highly relevant to decisions can reduce the training time. Therefore, a 14-dimensional state observation space is designed in this section, and the information was scaled to facilitate feature extraction with a neural network.

Taking the red UCAV as an example, the state observation space $\boldsymbol{S} = [s_1, s_2, \cdots, s_{14}]$ is shown in Table 2.

Table 2. State observation space.

State	Value	State	Value
s_1	$x_r/x_{\max}$	s_8	$x_b/x_{\max}$
s_2	$y_r/y_{\max}$	s_9	$y_b/y_{\max}$
s_3	$z_r/z_{\max}$	s_{10}	$z_b/z_{\max}$
s_4	$\|\boldsymbol{v}_r\|/v_{\max}$	s_{11}	$\|\boldsymbol{v}_b\|/v_{\max}$
s_5	$v_r^x/v_{\max}$	s_{12}	$v_b^x/v_{\max}$
s_6	$v_r^y/v_{\max}$	s_{13}	$v_b^y/v_{\max}$
s_7	$v_r^z/v_{\max}$	s_{14}	$v_b^z/v_{\max}$

3.4. Reward Function with Situation Reward Shaping

The design of the reward function is one of the keys to deciding whether the algorithm is convergent. Only when the UCAV of one side dies can victory or defeat be determined in air combat, which can be abstracted as a sparse reward problem in reinforcement learning. The sparse reward problem means that the agent finds it difficult to learn tasks with high exploration complexity due to the lack of a feedback signal [30]. In order to make the algorithm converge within a limited time, a reward function with situation reward shaping based on air combat situation assessment is designed in this section. The design using the air combat situation assessment can make the reward function continuous. The idea of situation reward shaping is to design the reward by the state situation of the agent, so that the agent will explore the terminal state faster to experience less punishment. The situation guidance reward is defined as:

$$B_{situ} = \frac{\|\boldsymbol{v}_r\| \cos(ATA)}{v_{\max}}, \tag{25}$$

The situation guidance reward reflects the current state of the UCAV. When the reward is positive, the UCAV is in an offensive situation, and when the reward is negative, the UCAV is in a defensive situation.

4. Simulation Results

4.1. Simulation Platform

The simulation environment in this paper was built with Python, and the algorithm was implemented based on PyTorch. The real-time confrontation display was based on Tacview. The experimental equipment was a desktop computer configured with an Intel(R) Xeon(R) Gold 6254 CPU @ 3.10 GHz and a single NVIDIA GeForce RTX 3090 GPU.

In order to ensure the fairness of the confrontation, the blue UCAV adopted the same parameters as the red UCAV. The relevant parameter settings are shown in Table 3.

Table 3. Relevant parameters of air combat.

Parameter	Value	Parameter	Value
α	$\pi/3$	$t_{\max}$	3×10^3
$R_{\min}$	500	k_v	20
$R_{\max}$	2×10^3	k_h	500
$v_{\min}$	100	k_d	2×10^3
$h_{\min}$	2×10^3	$d_{\text{threshold}}$	8×10^3

After debugging, the relevant parameters of deep reinforcement learning, including neural network parameters, were set as follows. The policy network and value network were constructed by a fully connected network. Both of the networks had two hidden layers with 64 and 64 units, respectively. The activation function was a tanh function. The PPO algorithm used eight environments to collect data in parallel in the simulation. The number of steps to run for each environment per update was 4096, the minibatch size was 128 and the number of epoch when optimizing the surrogate loss was 10. The discount factor $\tau = 0.99$, the value function squared-error loss coefficient $c_1 = 0.5$, the entropy bonus coefficient $c_2 = 0.001$, the clip range $\varepsilon = 0.2$ and the GAE factor $\lambda = 0.95$.

4.2. Results and Analysis

Figures 11 and 12 show that the different stages of the blue UCAV with the greedy algorithm chasing the green target moving in the horizontal plane and vertical plane, respectively. It can be seen from Figures 11 and 12 that the blue UCAV shot down the green target successfully through a series of maneuvers such as turn and climb. Moreover, the blue UCAV also had the ability to avoid entering the attack zone of the target, which verifies the effectiveness of the maneuver strategy based on the greedy algorithm.

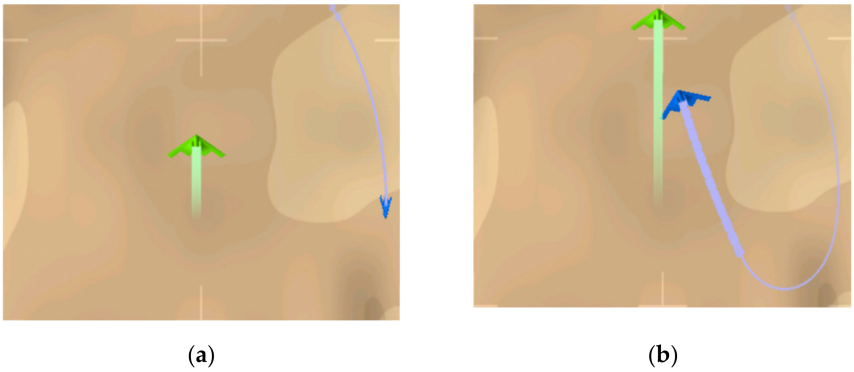


Figure 11. Blue UCAV chasing the green target moving in the horizontal plane: (**a**) early stage of air combat; (**b**) later stage of air combat.

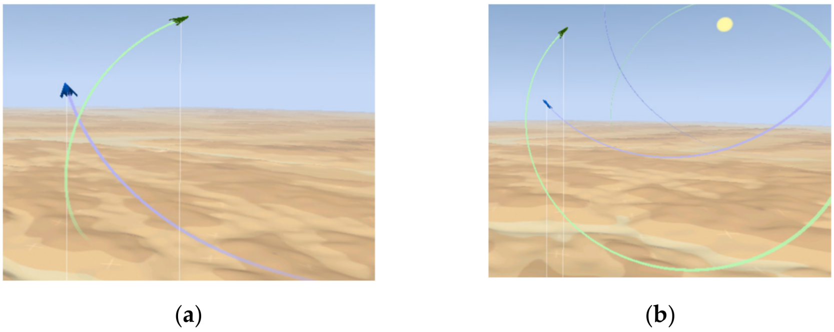


Figure 12. Blue UCAV chasing the green target moving in the vertical plane: (**a**) early stage of air combat; (**b**) later stage of air combat.

In order to ensure the fairness of the confrontation and to be consistent with an actual combat environment, the position, velocity and attitude of the UCAVs were initialized stochastically within a certain range in the training process. The initial states settings of the UCAVs are shown in Table 4.

Table 4. Initial states setting.

State	Value	State	Value
x	$[-10, 10]$ km	v	$[100, 600]$ m/s
y	$[-10, 10]$ km	γ	$[-\pi/2, \pi/2]$ rad
z	$[-12, -8]$ km	ψ	$[-\pi, \pi]$ rad

Figure 13 shows the training curve of the agents with different action spaces. The horizontal axis is the number of training time steps, and the vertical axis is the average reward value of each episode in the rollout of the PPO algorithm, which reflects the convergence of the training. As can be seen from Figure 13, the agent using discrete action space quickly improved the average reward per episode and stabilized at approximately 300. However, the agent using continuous action space had no prior experience and needed to learn flight control first. Therefore, the average reward increased slowly and only stabilized at approximately 50 in the later stage. The subsequent experiments in this paper used the discrete action space.

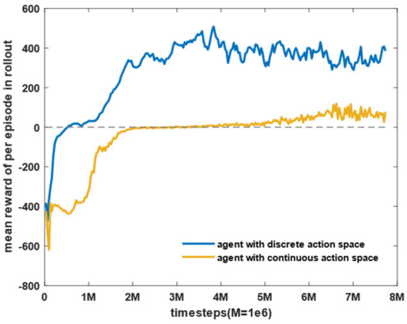


Figure 13. Training curve of the agents using different action spaces.

Figure 14 shows the average steps of the agent in each episode during the training process. It can be seen from the figure that the average length of the episode increased gradually in the early stage of the training but decreased gradually when the number of training time steps was greater than 2 M, and it finally stabilized at approximately 700 steps. In addition, the average reward value of per episode in Figure 13 tended to rise first and then stabilize, which shows that the average reward received by the agent at each step gradually increased.

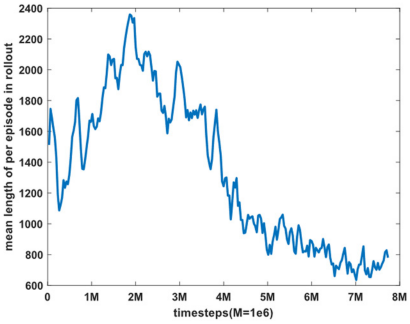


Figure 14. Mean length per episode in the training process.

Figures 15 and 16 show the entropy loss and the explained variance of the PPO algorithm during the training process, respectively. The entropy loss reflects the randomness of the actions of the agent. As can be seen from Figure 15, the entropy loss gradually dropped to a lower level after the beginning of training, which indicates that the policy of the agent was relatively stable in the later stage of training. The explained variance reflects the fitting accuracy of the value network of the PPO algorithm. The closer it is to one, the higher the accuracy. It can be seen from Figure 16 that the explained variance rose rapidly in the beginning of training and increased with the training time steps, and it finally stabilized at approximately one. Figures 15 and 16 show that the performance of the policy network of the agent converged to a high level after the training.

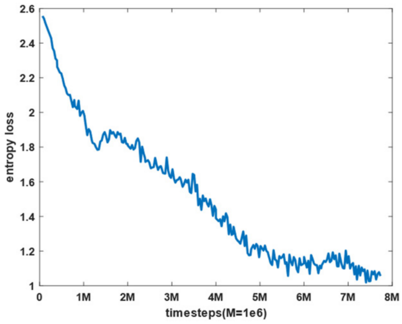


Figure 15. Entropy loss of the PPO algorithm in the training process.

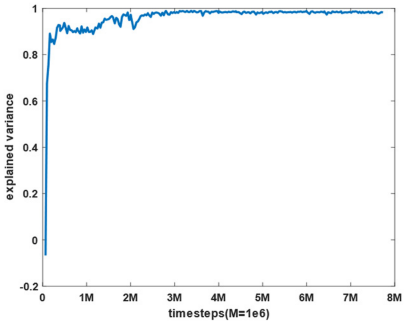


Figure 16. Explained variance of the PPO algorithm in the training process.

Figure 17 shows the results of 100 episodes of air combat confrontation between the agent and the blue UCAV based on the greedy policy. The confrontation results are divided into four types including winning, losing, drawing and reaching the maximum number of steps in the current episode. Winning means that the UCAV shot down the enemy, or the enemy went beyond the margins. Losing means that the enemy shot down the UCAV, or the UCAV went beyond the margins. Drawing means that the two UCAVs met the firing conditions at the same time and were shot down by the other side, or the two the UCAVs went beyond the margins. The horizontal axis is the number of training steps, and the vertical axis is the percentage of the results. Green represents winning, red represents losing, yellow represents a tie, and grey represents that the current episode reached the maximum number of steps. As can be seen from Figure 17, the winning rate of the agent gradually rose with the increase in the number of training time steps and tended to be stable. After training, the winning rate can reach 62%, and the corresponding losing rate is only 11%. In addition, the average decision time per step of the PPO algorithm was 0.98 ms and that of the greedy algorithm was 1.62 ms, both of which were less than the simulation step size of 0.1 s, thus meeting the real-time requirements. It should be pointed out that even in the later stage of training, the greedy algorithm could still maintain a certain unbeaten rate for the blue UCAV, which shows that it also had a strong autonomous decision ability.

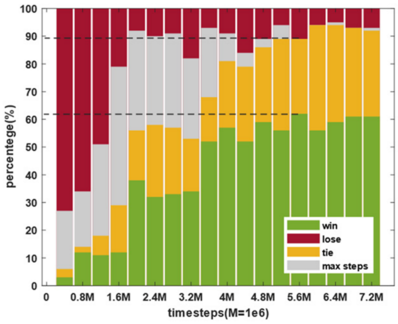


Figure 17. Air combat confrontation results.

Figures 14 and 17 show that the agent lacked attack skills in the early stage of training, but learned to avoid attack from the blue UCAV so as to increase the cumulative reward by prolonging the survival steps. As a result, the agent in the early stage of training often reached the maximum number of steps. When the training reached a certain level, the agent gradually learned the attack skills, which could not only defeat the blue UCAV but also reduce the maneuver time steps. The above conditions show that the maneuver decision method proposed in this paper can enable a UCAV to learn the maneuver policy autonomously and defeat the enemy.

The experiments in this paper realized real-time situation display through a TCP communication protocol and Tacview software. Figure 18 is an example of the real-time display. Figure 19 displays the recorded trajectory curves of the two UCAVs for the example in Figure 18. As can be seen from Figures 18 and 19, the agent had learned to combine a series of basic maneuvers, such as diving, climbing and circling, into tactical maneuvers and finally defeated the enemy.

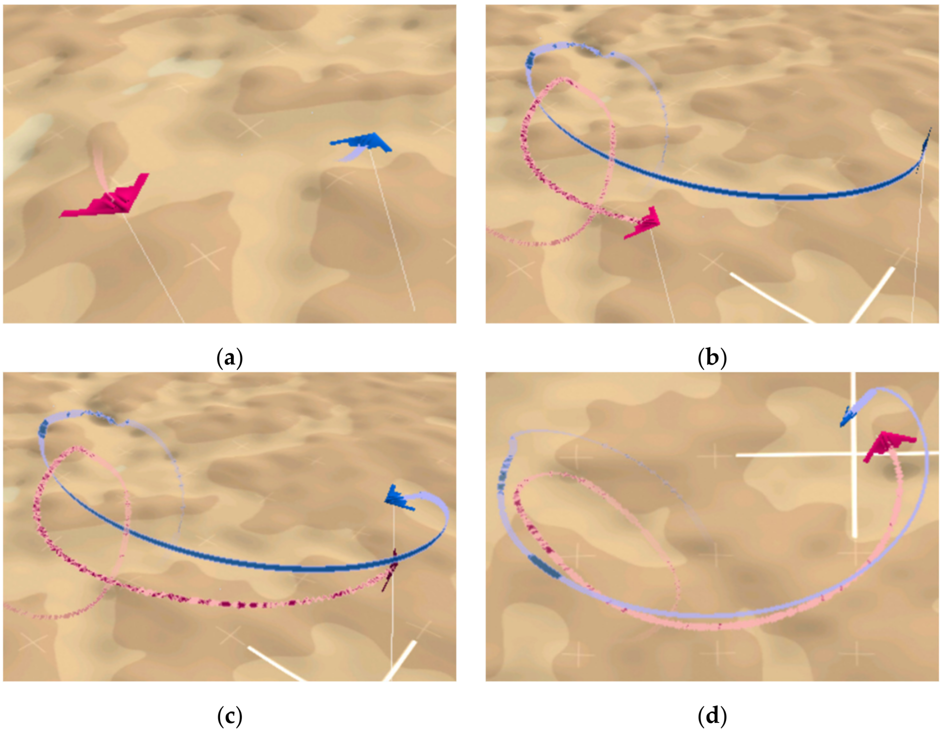
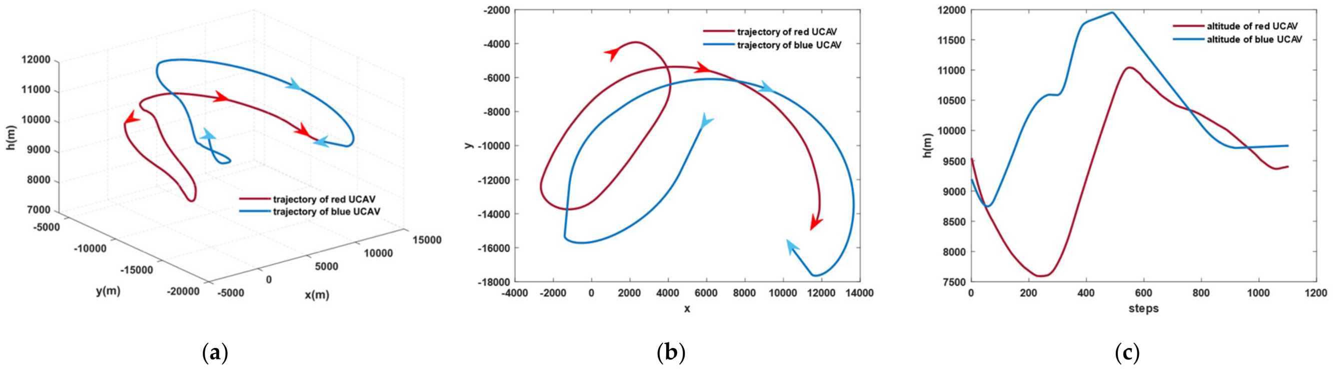


Figure 18. Real-time display example of air combat confrontation: (**a**) early stage of air combat; (**b**) medium stage of air combat; (**c**) later stage of air combat; (**d**) top view of the air combat.



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

Figure 19. Trajectories for the air combat confrontation example: (**a**) two UCAVs; (**b**) top view of the two UCAVs; (**c**) altitudes of the two UCAVs.

5. Conclusions

In this paper, a UCAV air combat maneuver decision method based on a PPO algorithm was proposed. Firstly, a general UAV short-range air combat confrontation framework was established including a motion model of a fixed-wing aircraft and a situation assessment model of air combat. In order to verify the effectiveness of the method, an enemy maneuver strategy based on a greedy algorithm was designed. Then, the action space based on a basic maneuver library and a state observation space for the agent were constructed. A maneuver policy was generated by training with the PPO algorithm. A reward function based on situation reward shaping was designed for accelerating the convergence rate of the algorithm. Finally, simulations of one-to-one, short-range air combat in stochastic situations were carried out to verify the effectiveness of the PPO algorithm. The experiment with agents using different action spaces showed that the discrete action space may reduce the difficulty of flight control and increase the learning efficiency. The experiment with the blue UCAV using a greedy algorithm to chase a moving target resulting in 100 episodes of air combat confrontation showed that the greedy algorithm also has a strong autonomous decision ability, but the performance of the PPO algorithm was better. The winning rate of the PPO algorithm reached 62%, and the corresponding losing rate was only 11%. In addition, the average decision time per step of the PPO algorithm was less than the simulation step size of 0.1 s, meeting real-time requirements. As can be seen from the real-time display example, the agent learned to combine a series of basic maneuvers, such as diving, climbing and circling, into tactical maneuvers. The results of the simulations show that the maneuver decision method proposed in this paper can enable a UCAV to learn the maneuver policy autonomously and defeat the enemy.

In addition, the proposed UCAV maneuver decision method is essentially motion control of a UAV. Therefore, it also has application potential in other fields including monitoring, organizing communication, search and transportation [31–33].

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

Funding: This research was funded by the Experience-Based Reinforcement Learning Approach for UAV Control (grant number: 61806217).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: Thank you to our partners for their trust and collaboration and for facilitating our work's completion. Thank you to the editors for their patience, guidance and corrections. Thank you to Wenhan Dong, who supported us in this work the entire time.

Conflicts of Interest: The authors declare no conflict of interest.

Nomenclature

$\boldsymbol{p}$	position vector of a UCAV
n_x	longitudinal overload my
n_z	normal overload
$\boldsymbol{u}$	control input
$\boldsymbol{LOS}$	line of sight
ATA	antenna train angle
AA	aspect angle
HCA	heading crossing angle
α	maximum attack angle of a UCAV
R	attack range of a UCAV
η	air combat situation
$\pi(\cdot)$	maneuver policy of a UCAV
$\boldsymbol{a}$	Action vector of a UCAV
$V^{\pi}(s)$	value function of reinforcement learning
$L(\theta)$	loss function of a neural network
$r_t(\theta)$	probability ratio of a policy update
$\boldsymbol{A}$	action space of reinforcement learning
$\boldsymbol{S}$	state observation space of reinforcement learning
τ	discount factor of reinforcement learning
c_1	value function squared-error loss coefficient
c_2	entropy bonus coefficient
ε	clip range of the PPO
λ	generalized advantage estimation factor

References

1. McManus, J.W.; Chappell, A.R.; Arbuckle, P.D. Situation Assessment in the Paladin Tactical Decision Generation System. In *AGARD Conference AGARD-CP-504: Air Vehicle Mission Control and Management*; NATO: Amsterdam, The Netherlands, 1992.
2. Burgin, G.H. *Improvements to the Adaptive Maneuvering Logic Program*; NASA CR-3985; NASA: Washington, DC, USA, 1986.
3. 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. Def. Manag.* **2016**, *6*, 1–7. [CrossRef]
4. DARPA. Air Combat Evolution. Available online: https://www.darpa.mil/program/air-combat-evolution (accessed on 24 June 2022).
5. Vajda, S. Differential Games. A Mathematical Theory with Applications to Warfare and Pursuit, Control and Optimization. By Rufus Isaacs. Pp. xxii, 384. 113s. 1965. (Wiley). *Math. Gaz.* **1967**, *51*, 80–81. [CrossRef]
6. Mendoza, L. Qualitative Differential Equations. *Dict. Bioinform. Comput. Biol.* **2004**, *68*, 421–430. [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. Bullock, H.E. ACE: The Airborne Combat Expert Systems: An Exposition in Two Parts. Master's Thesis, Defense Technical Information Center, Fort Belvoir, VA, USA, 1986.
9. Chin, H.H. Knowledge-based system of supermaneuver selection for pilot aiding. *J. Aircr.* **1989**, *26*, 1111–1117. [CrossRef]
10. Wang, R.; Gao, Z. Research on Decision System in Air Combat Simulation Using Maneuver Library. *Flight Dyn.* **2009**, *27*, 72–75.
11. Xuan, W.; Weijia, W.; Kepu, S.; Minwen, W. UAV Air Combat Decision Based on Evolutionary Expert System Tree. *Ordnance Ind. Autom.* **2019**, *38*, 42–47.
12. Huang, C.; Dong, K.; Huang, H.; Tang, S.; Zhang, Z. Autonomous air combat maneuver decision using Bayesian inference and moving horizon optimization. *J. Syst. Eng. Electron.* **2018**, *29*, 86–97. [CrossRef]
13. Cao, Y.; Kou, Y.-X.; Xu, A.; Xi, Z.-F. Target Threat Assessment in Air Combat Based on Improved Glowworm Swarm Optimization and ELM Neural Network. *Int. J. Aerosp. Eng.* **2021**, *2021*, 4687167. [CrossRef]
14. Kaneshige, J.; Krishnakumar, K. Artificial immune system approach for air combat maneuvering. In Proceedings of the SPIE 6560, Intelligent Computing: Theory and Applications V, Orlando, FL, USA, 9–13 April 2007; Volume 6560, p. 656009. [CrossRef]
15. Başpinar, B.; Koyuncu, E. Assessment of Aerial Combat Game via Optimization-Based Receding Horizon Control. *IEEE Access* **2020**, *8*, 35853–35863. [CrossRef]

16. François-Lavet, V.; Henderson, P.; Islam, R.; Bellemare, M.G.; Pineau, J. An Introduction to Deep Reinforcement Learning. *Found. Trends Mach. Learn.* **2018**, *11*, 219–354. [CrossRef]
17. McGrew, J.S.; How, J.; Williams, B.C.; Roy, N. Air-Combat Strategy Using Approximate Dynamic Programming. *J. Guid. Control Dyn.* **2010**, *33*, 1641–1654. [CrossRef]
18. Liu, P.; Ma, Y. A Deep Reinforcement Learning Based Intelligent Decision Method for UCAV Air Combat. In *Modeling, Design and Simulation of Systems, Proceedings of the 17th Asia Simulation Conference, AsiaSim 2017, Malacca, Malaysia, 27–29 August 2017*; Communications in Computer and Information Science; Springer: Singapore, 2017; pp. 274–286. [CrossRef]
19. Zhang, X.; Liu, G.; Yang, C.; Wu, J. Research on Air Confrontation Maneuver Decision-Making Method Based on Reinforcement Learning. *Electronics* **2018**, *7*, 279. [CrossRef]
20. Yang, Q.; Zhang, J.; Shi, G.; Hu, J.; Wu, Y. Maneuver Decision of UAV in Short-Range Air Combat Based on Deep Reinforcement Learning. *IEEE Access* **2019**, *8*, 363–378. [CrossRef]
21. Kong, W.; Zhou, D.; Yang, Z.; Zhao, Y.; Zhang, K. UAV Autonomous Aerial Combat Maneuver Strategy Generation with Observation Error Based on State-Adversarial Deep Deterministic Policy Gradient and Inverse Reinforcement Learning. *Electronics* **2020**, *9*, 1121. [CrossRef]
22. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal Policy Optimization Algorithms. *arXiv* **2017**, arXiv:1707.06347.
23. Hu, J.; Wang, L.; Hu, T.; Guo, C.; Wang, Y. Autonomous Maneuver Decision Making of Dual-UAV Cooperative Air Combat Based on Deep Reinforcement Learning. *Electronics* **2022**, *11*, 467. [CrossRef]
24. Austin, F.; Carbone, G.; Falco, M.; Hinz, H.; Lewis, M. Automated maneuvering decisions for air-to-air combat. In Proceedings of the Guidance, Navigation and Control Conference, Monterey, CA, USA, 17–19 August 1987. [CrossRef]
25. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; MIT Press: Cambridge, MA, USA, 2018.
26. Paczkowski, M. Low-Friction Composite Creping Blades Improve Tissue Properties. In Proceedings of the Pulp and Paper, Stockholm, Sweden, 9 October 1996; Volume 70.
27. Williams, R.J. Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. *Mach. Learn.* **1992**, *8*, 229–256. [CrossRef]
28. Mnih, V.; Badia, A.P.; Mirza, L.; Graves, A.; Lillicrap, T.; Harley, T.; Silver, D.; Kavukcuoglu, K. Asynchronous Methods for Deep Reinforcement Learning. In Proceedings of the 33rd International Conference on Machine Learning, ICML 2016, New York, NY, USA, 19–24 June 2016; Volume 48, pp. 1928–1937.
29. Schulman, J.; Moritz, P.; Levine, S.; Jordan, M.; Abbeel, P. High-Dimensional Continuous Control Using Generalized Advantage Estimation. In Proceedings of the 4th International Conference on Learning Representations, ICLR 2016–Conference Track Proceedings, San Juan, Puerto Rico, 2–4 May 2016; pp. 1–14.
30. Riedmiller, M.; Hafner, R.; Lampe, T.; Neunert, M.; Degrave, J.; Van De Wiele, T.; Mnih, V.; Heess, N.; Springenberg, J.T. Learning by Playing Solving Sparse Reward Tasks from Scratch. In Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholm, Sweden, 10–15 July 2018; Volume 80, pp. 4344–4353.
31. Mukhamediev, R.I.; Symagulov, A.; Kuchin, Y.; Zaitseva, E.; Bekbotayeva, A.; Yakunin, K.; Assanov, I.; Levashenko, V.; Popova, Y.; Akzhalova, A.; et al. Review of Some Applications of Unmanned Aerial Vehicles Technology in the Resource-Rich Country. *Appl. Sci.* **2021**, *11*, 10171. [CrossRef]
32. Agarwal, A.; Kumar, S.; Singh, D. Development of Neural Network Based Adaptive Change Detection Technique for Land Terrain Monitoring with Satellite and Drone Images. *Def. Sci. J.* **2019**, *69*, 474–480. [CrossRef]
33. Smith, M.L.; Smith, L.N.; Hansen, M.F. The quiet revolution in machine vision–A state-of-the-art survey paper, including historical review, perspectives, and future directions. *Comput. Ind.* **2021**, *130*, 103472. [CrossRef]