Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 1477078, 13 pages
https://doi.org/10.1155/2022/1477078

Research Article

Research on UCAV Maneuvering Decision Method Based on Heuristic Reinforcement Learning

Wang Yuan,[1] **Zhang Xiwen**,[1] **Zhou Rong**,[2] **Tang Shangqin**,[1] **Zhou Huan**,[1] **and Ding Wei**[1]

[1]*Air Force Engineering University, Xi'an, China*
[2]*Southeast University, Nanjing, China*

Correspondence should be addressed to Zhang Xiwen; xwxxn1023@163.com

Received 16 December 2021; Accepted 4 February 2022; Published 3 March 2022

Academic Editor: Ahmed Mostafa Khalil

With the rapid development of unmanned combat aerial vehicle (UCAV)-related technologies, UCAVs are playing an increasingly important role in military operations. It has become an inevitable trend in the development of future air combat battlefields that UCAVs complete air combat tasks independently to acquire air superiority. In this paper, the UCAV maneuver decision problem in continuous action space is studied based on the deep reinforcement learning strategy optimization method. The UCAV platform model of continuous action space was established. Focusing on the problem of insufficient exploration ability of Ornstein–Uhlenbeck (OU) exploration strategy in the deep deterministic policy gradient (DDPG) algorithm, a heuristic DDPG algorithm was proposed by introducing heuristic exploration strategy, and then a UCAV air combat maneuver decision method based on a heuristic DDPG algorithm is proposed. The superior performance of the algorithm is verified by comparison with different algorithms in the test environment, and the effectiveness of the decision method is verified by simulation of air combat tasks with different difficulty and attack modes.

1. Introduction

From a macro point of view, air combat decision making refers to one party in air combat providing corresponding control instructions to fighter jets after analyzing and judging battlefield information so that it can complete the dominant attack position occupying the enemy. Decision making is the core of air combat, and its rationality will determine the final outcome of air combat [1].

In recent years, with the continuous improvement and development of deep learning (DL) theory, the deep reinforcement learning (DRL) algorithm combined with deep learning and reinforcement learning has become a research hotspot in artificial intelligence. With no training samples, not limited by specific models, and able to take into account the long-term impact of actions and other advantages, deep reinforcement learning methods have gradually received attention in the research of air combat maneuver decision making. Deep reinforcement learning can be divided into two types: value-based reinforcement learning algorithms and policy-based reinforcement learning algorithm. [2–4].

Watkins proposed Q learning on the basis of dynamic programming, which forms the evaluation value of each state action through repeated experiments and iterations. However, due to the limitation of the look-up table method, its algorithm is only applicable to the applications of finite state space and action space. Subsequently, with the increasing dimension of the state space of the research object, DNNs, CNNs, or RNNs were used to replace the action value function Q, forming the deep Q network algorithm (DQN) [5, 6] and introducing the experience replay target q-value network. In reference [7], the DQN algorithm is used to construct autonomous obstacle avoidance decisions for UAVs. By transforming the obstacle avoidance process of UAVs into a Markov decision problem and introducing neural networks for the decision model and improving the replay process, random dynamic obstacle avoidance of UCAVs in a 3D environment is realized, which effectively improves the efficiency of task execution. The DeepMind team realized autonomous learning in the Openai Gym simulation platform based on the DQN algorithm [8] and won the battle with professional players with absolute results, which again proved that DQN has obvious advantages over traditional algorithms and humans in decision-making ability. Subsequently, the AlphaGO System and the AlphaGO Master were developed and used to defeat all the world champions, which caused a sensation and made people reunderstand artificial intelligence technology. In 2017, AlphaGo Zero realized a self-game, started training without task samples, and further improved both speed and effect. Silver et al. [9] and Liu and Ma [10] constructed a discrete UAV maneuvering action library and realized the autonomous attack of a low-dynamic UAV by using a DQN. In reference [11], the DQN algorithm is used in UAV air combat confrontation, and the min-max algorithm is used to solve value functions in different states. The simulation result verifies that this method has good effects.

Value-based reinforcement learning methods cannot deal with the problem of continuous action space [12–15]. Lillicrap combined the deterministic policy gradient algorithm [16] and actor-critic framework, and a deep deterministic policy gradient (DDPG) algorithm is proposed to address continuous state space and continuous action space problems [17].

Wang et al. used the DDPG algorithm to study the pursuit strategy of a car in a plane. [18] Yang used the DDPG algorithm to construct an air combat decision system. Focusing on the problem of low data utilization due to the lack of prior knowledge of air combat in the DDPG algorithm, they proposed adding the sample data of the existing mature maneuvering decision-making system into the replay buffer in the initial training stage to prevent the DDPG algorithm from falling into a local optimum during training. Thus, the convergence speed of the algorithm was accelerated. [19].

At present, deep reinforcement learning has been widely applied in unmanned vehicle control, [20] robot path planning and control, [21] pursuit and avoidance of targets, [22] unmanned driving [23, 24], and real-time strategy games [25, 26]. However, most of the reinforcement learning algorithms used in air combat maneuvering decision making are discrete action space algorithms, which inevitably face the problems of rough flight paths and limited reachable domains. At the same time, model-free deep reinforcement learning algorithms are widely used at present, which are capable of self-learning effective air combat maneuver strategies independent of human air combat expert experience and have a general learning framework. However, model-free deep reinforcement learning algorithms need to interact with the environment to obtain a large number of training samples, and inefficient data utilization and learning efficiency become important bottlenecks in the practical application of model-free reinforcement learning methods. [3, 27–30].

According to the above problems, in this paper, the UCAV maneuvering decision-making problem in continuous action space is studied. By introducing a heuristic exploration strategy, the problem of insufficient exploration strategy exploration ability and low data utilization in the DDPG algorithm is improved, and then a UCAV air combat maneuver decision-making method based on the heuristic DDPG algorithm is proposed.

2. Air Combat Environment Design

2.1. Flight Motion Model. To consider the coupling relationship between the control quantities when continuous control quantities are independently sought, the UCAV platform model based on the angle of attack, engine thrust, and roll angle as control quantities is adopted to fully consider the influence of the aerodynamic characteristics of the platform on the flight state so that the model is closer to reality and the flight trajectory is more realistic, increasing its engineering use value. Its three-degree-of-freedom mass kinematic model is as follows:

$$\begin{cases} \dot{x} = v\cos\gamma\cos\psi, \\ \dot{y} = v\cos\gamma\sin\psi, \\ \dot{z} = v\sin\gamma, \end{cases} \tag{1}$$

where $\dot{x}$, $\dot{y}$, and $\dot{z}$ are the components of the velocity of the UCAV in the inertial coordinate system. γ represents the flight path angle, and ψ represents the yaw angle.

The updated equations for its velocity v, flight path angle γ, and yaw angle ψ, i.e., the particle dynamics model, are as follows:

$$\begin{cases} \dot{v} = \dfrac{T\cos\alpha - D}{m} - g\sin\gamma, \\ \dot{\gamma} = \dfrac{(L + T\sin\alpha)\cos\phi}{mv} - \dfrac{g}{v}\cos\gamma, \\ \dot{\psi} = \dfrac{(L + T\sin\alpha)\sin\phi}{mv\cos\gamma}, \end{cases} \tag{2}$$

where m is the mass of the UCAV, g is the acceleration of gravity, D is the drag parameter, and T, α, and ϕ are the three control quantities of the model, that is, the angle of attack, engine thrust, and roll angle, respectively.

As seen from the above equation, to obtain a direct mapping relationship between the model control quantities $u = [T, \alpha, \phi]$ and the state change, the drag parameter D, the lift parameter L, and the thrust T need to be solved; however, as the drag, lift, and thrust are influenced by various factors, such as altitude, atmospheric density, aerodynamic shape, and flight speed, and are strongly coupled and nonlinear, their parameter expressions are difficult to derive through traditional mechanics. In this paper, the relevant aerodynamic parameters of a publicly available storm shadow UAV [31] are fitted by a BP neural network [32–35] to determine the important aerodynamic and dynamic characteristics, with the aim of establishing a more detailed and realistic model of the UCAV platform, which will provide the basis for subsequent maneuvering decision making in continuous action space.

2.2. Geometry of Air Combat. When describing the geometric relationship between aircraft in air combat, the important factors usually considered are the distance between two aircraft, heading crossing angle (HCA), line of sight (LOS), antenna train angle (ATA), and aspect angle (AA). The distance between two aircraft is usually expressed by the calculation $R = \text{norm}\,(x_e - x, y_e - y, z_e - z)$, which is an important factor to evaluate the air combat situation and judge the launching conditions of weapons. HCA refers to the angle formed by two aircraft courses. LOS is the line between the UCAV and the enemy aircraft. AA refers to the included angle between the LOS and the direction of the enemy aircraft, which represents the angle relation between our aircraft and the enemy under the current attitude. When AA = 180, it indicates that our aircraft is on the extension line of the axis direction of the enemy aircraft's body; that is, the nose of the enemy aircraft is facing our aircraft. ATA refers to the included angle between the sighting line vector and the pointing direction of the axis of the aircraft body and represents the angle relation between the enemy aircraft and the current attitude of the aircraft. ATA = 0° when the enemy aircraft is directly in front of the nose of the aircraft. The above geometric relationship between the enemy and us is shown in Figure 1.

ATA and AA can be expressed as

$$\begin{aligned} \text{ATA} &= \arccos \frac{\mathbf{R} \times \mathbf{V}_u}{\|\mathbf{R}\| \times \|\mathbf{V_u}\|}, \text{ATA} \in [0, \pi], \\ \text{AA} &= \pi - \arccos \frac{\mathbf{R} \times \mathbf{V}_e}{\|\mathbf{R}\| \times \|\mathbf{V}_e\|}, \text{AA} \in [0, \pi]. \end{aligned} \tag{3}$$

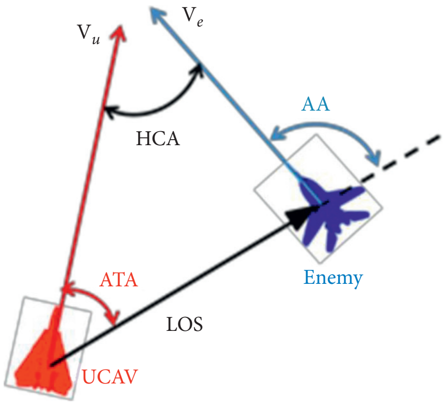


Figure 1: Geometric relation of air combat position.

2.3. Reward Shaping. The objective of maneuver decisions in close air combat based on reinforcement learning is to find an optimal maneuver strategy to enable the UCAV to complete the attack position to maximize the current cumulative reward. Reward is the only quantitative index of strategy evaluation, which determines the final learning strategy of an agent and directly affects the convergence and learning speed of the algorithm. When the UCAV conducts air combat decision making through deep reinforcement learning, except for the reward for completing the task, there is no reward in the middle process, and there is the problem of sparse reward. Therefore, it is not only necessary to design the reward for completing the task but also crucial to design the guiding reward for each step in each round. In this paper, a reward function including angle, height, distance, and speed factors is designed.

2.3.1. Angle Factor. When the maximum firing range of the UCAV weapon is superior to that of the enemy, the UCAV missile firing conditions can be preferentially met in the head-on encounter with the enemy. Due to the omnidirectional attack capability of the fourth-generation short-range air-to-air missile, there is no need to consider the attitude of the enemy at this time. Therefore, under the current weapon advantage, the angle factor is mainly determined by the ATA of the UCAV. As long as the ATA angle is within the range of the maximum off-axis launch angle, the angle reward can be obtained, specifically expressed as

$$\begin{aligned} &\text{if} \quad L^u_{\max} > L^e_{\max}: \\ &r^t_{A\text{-FRO}}(s_t) = \begin{cases} 1, & \text{ATA} \le \theta^u_{\max}, \\ -1, & \text{others}, \end{cases} \end{aligned} \tag{4}$$

where $L^u_{\max}$ is the maximum launching distance of the UCAV airborne weapon, $L^e_{\max}$ is the maximum launching distance of the enemy aircraft weapon, and $\theta^u_{\max}$ is the maximum off-axis launching angle of the UCAV airborne missile.

When the maximum firing distance of the UCAV weapon is weaker than that of enemy aircraft, it is extremely detrimental to UCAV security. At this time, to ensure their own safety, UCAV should be guided to give full play to their maneuverability and always be located beyond the maximum off-axis angle of enemy aircraft and attack enemy aircraft as far as possible with the tactics of tail attack. In this case, the angle factor should consider both the ATA of the UCAV and AA of the enemy aircraft, and the angle factor design is as follows:

$$if \quad L^u_{\max} < L^e_{\max}: r^t_{A\text{-BAC}}(s_t) = \begin{cases} 1, & \begin{array}{l} \text{if ATA} \le \theta^u_{\max}, \\ \dfrac{\pi}{2} \le AA, \end{array} \\ 0, & \text{if AA} \le \dfrac{\pi}{2}, \\ -1, & \text{if others}, \end{cases} \tag{5}$$

where $L^u_{\max}$ is the maximum launching distance of the UCAV airborne weapon, $L^e_{\max}$ is the maximum launching distance of the enemy aircraft weapon, and $\theta^u_{\max}$ is the maximum off-axis launching angle of the UCAV airborne missile.

2.3.2. Height Factor. The height factor not only represents the relationship between the two in the vertical plane in the geometry situation of air combat but also measures the energy advantage of the UCAV. The side that satisfies the height advantage not only has the advantage of energy mobility but can also exert the missile's larger attack range. A high reward factor is achieved when the UCAV is in the desired altitude range relative to the enemy aircraft:

$$r_H^t(s_t) = \begin{cases} 1, & \Delta H_{\text{down}} \le \Delta H \le \Delta H_{\text{up}}, \\ -1, & \text{others}, \end{cases} \tag{6}$$

where $\Delta H = z_u - z_e$ represents the relative height of the UCAV and the enemy aircraft, ΔH_{up} is the upper limit of maintaining the altitude advantage, and ΔH_{down} is the lower limit of maintaining the altitude advantage.

2.3.3. Distance Factor. Distance is an important factor for UCAV platform situation assessment and weapon launch conditions. When the relative distance between two aircraft meets the maximum missile launch distance, the maximum distance factor can be obtained, which is defined as

$$r_R^t(s_t) = \begin{cases} 1, & L_{\min}^u \le R_{\text{LOS}} \le L_{\max}^u, \\ -1, & \text{others}, \end{cases} \tag{7}$$

where $R_{\text{LOS}} = \text{norm}[x_e - x_u, y_e - y_u, z_e - z_u]$ and $L_{\max}^u$ and $L_{\min}^u$ are the maximum and minimum firing ranges of UCAV airborne weapons, respectively.

2.3.4. Speed Factor. When the distance between the two planes reaches the maximum launching distance of the missile, the UCAV sees the speed of the enemy aircraft as the best attack speed. When the distance between the two planes is relatively far, the UCAV should maintain a large flight speed to rapidly form a favorable situation and maintain a kinetic energy advantage with the help of high speed and maneuverability. The speed factor is established as follows:

$$r_V^t(s_t) = \begin{cases} 1, & L_{\min}^u \le R_{\text{LOS}} \le L_{\max}^u \cap |\Delta v| \le \delta_v, \\ 0.5, & R_{\text{LOS}} \le L_{\min}^u \cap v_u > v_e, \\ 0.5, & R_{\text{LOS}} \ge L_{\max}^u \cap v_u > v_e, \\ -1, & \text{others}, \end{cases} \tag{8}$$

where $\Delta v = v_u - v_e$ represents the relative speed of the UCAV and enemy aircraft and δ_v is the allowable relative speed difference from the optimal attack speed.

2.3.5. Environmental Factor. When the UCAV air combat strategy is learned through reinforcement learning, in addition to making the UCAV capable of attacking enemy aircraft, the more important prerequisite is that the UCAV has the ability to adapt to the battlefield environment and maintain a safe flight altitude. Therefore, to train the air combat strategy with both air combat capability and safe flight capability, it is necessary to set negative returns in the form of punishment for dangerous flight maneuvers, so the environmental factor r_{ENV}^t is constructed as follows:

$$r_{\text{ENV}}^t = \begin{cases} -50, & (x_u, y_u, z_u, v_u) \notin (x_{\text{limit}}, y_{\text{limit}}, z_{\text{limit}}, v_{\text{limit}}), \\ 0, & \text{others}, \end{cases} \tag{9}$$

where $x_{\text{limit}} = [x_{\min}, x_{\max}], y_{\text{limit}} = [y_{\min}, y_{\max}]$, and $z_{\text{limit}} = [h_{\min}, h_{\max}]$ represent the range of the operational airspace on the X, Y, and Z axes of the inertial coordinate system, and $v_{\text{limit}} = [v_{\min}, v_{\max}]$ represents the extreme value of the safe flight speed of the UCAV.

2.3.6. End Factor

$$r_{\text{END}}^t = \begin{cases} 100, & \text{if End} = \text{Win}, \\ -100, & \text{if End} = \text{Loss}, \\ 0, & \text{others}, \end{cases} \tag{10}$$

where End is the result of outcome determination. When the angle, height, distance, and speed reward factors of the UCAV are 1 at the same time, the weapon launch condition is reached, and the UCAV is judged to win, where End can be expressed as

$$\text{End}(t) = \begin{cases} \text{Win}, & \text{if } r_A^t = \begin{Bmatrix} r_{A\text{-FRO}}^t(s_t), & \text{if } L_{\max}^u > L_{\max}^e \\ r_{A\text{-BAC}}^t(s_t), & \text{if } L_{\max}^u < L_{\max}^e \end{Bmatrix} = r_R^t = r_H^t = r_V^t = 1, \\ \text{Loss}, & \text{if enemy win}. \end{cases} \tag{11}$$

2.3.7. Total Reward Function. Based on the above analysis, the total reward function is

$$r_t(s_t) = \begin{Bmatrix} r_{A\text{-FRO}}^t(s_t), & \text{if} L_{\max}^u > L_{\max}^e \\ r_{A\text{-BAC}}^t(s_t), & \text{if} L_{\max}^u < L_{\max}^e \end{Bmatrix} + r_H^t + r_R^t + r_V^t + r_{\text{ENV}}^t + r_{\text{END}}^t. \tag{12}$$

3. Heuristic DDPG Algorithm

This section constructs an exploration strategy that is more effective than traditional Gaussian noise or OU noise. At present, the traditional exploration strategy such as OU usually acts directly on the actions of the strategy network output and makes the actions randomly disturbed in the form of addition to realize the exploration of unknown space. In an air combat environment, unmanned combat aircraft control the amount of high dimensionality and large amplitude range; therefore, the DDPG algorithm is based on the strategy of OU explores noise and is likely to create many blind spots in the search. The serious influence training effect, at the same time, is based on the limited performance and flight safety, UCAV variation and volume control of each dimension has a strict limit. When the output action of the policy network is close to the boundary of its scope, it is blind and ineffective to implement exploration by adding noise directly.

A large number of current research exploration methods that potentially set the exploration strategy π_e and action generation strategy π are highly similar and can be improved by applying various forms of random noise to the action generation strategy. However, this setting is not conducive to the expansion of state exploration. In contrast, the space that has not been explored in the past will be explored only when there is a significant difference between exploration strategy π_e and action generation strategy π. The amplitude of Gaussian noise is an important parameter that has been discussed for a long time, even in the method of using Gaussian noise as the exploration mode. Therefore, it is of great theoretical value and engineering significance to construct an adaptive exploration system method to replace the traditional probabilistic method.

As the DDPG algorithm is a typical off-policy learning method, its exploration process and learning process are independent from each other, so the exploration strategy π_e can be decoupled from the action generation strategy π. The specific idea is to construct a more efficient heuristic exploration strategy acting on the experience replay space to have a more positive role in the training of the action generation strategy π.

3.1. Algorithm Design

3.1. Algorithm Design. The framework proposed in this section can be regarded as a heuristic learning framework, in which the exploration strategy π_e acts as the heurist and generates a set of heuristic information D_0 during each iteration, and the action generation strategy π learned by the DDPG algorithm acts as the heurist and receives the D_0 heuristic strategy π_e and carries out training and learning. Therefore, the decisive factor is changed to adaptively improve the exploration strategy π_e to generate optimal value heuristic information D_0 according to the learning efficiency of the DDPG algorithm so that DDPG can learn as quickly and effectively as possible.

The generation of heuristic information D_0 can be considered as the action a_e performed by the exploration strategy π_e, and its related reward function should be defined as the improvement of the DDPG algorithm through heuristic information D_0:

$$\begin{aligned} J(\pi_e) &= E_{D_0 \sim \pi_e}\left[R(\pi, D_0)\right] \\ &= E_{D_0 \sim \pi_e}\left[R_{\pi'} - R_\pi\right], \end{aligned} \tag{13}$$

where $\pi' = \mathrm{DDPG}(\pi, D_0)$ represents the new strategy obtained by one or more updated steps of the DDPG algorithm on the basis of heuristic information D_0, $R_{\pi'}$, and R_π represent the cumulative rewards of the DDPG algorithm obtained by strategy π' and strategy π interaction with the environment, respectively, which have no relationship with exploration strategy π_e. $R(\pi, D_0)$ represents the extent to which the heurist (exploration strategy π_e) helps the heurist (DDPG algorithm) in the learning process.

Referring to the parameterized representation of the policy network in traditional DDPG, the policy π_e can be parameterized by parameters θ^{π_e}. Similar to the traditional reinforcement learning method, the gradient $J(\pi_e)$ of parameters θ^{π_e} can be calculated as follows:

$$\nabla_{\theta^{\pi_e}} J(\pi_e) = E_{D_0 \sim \pi_e}\left[R(\pi, D_0) \nabla_{\theta^{\pi_e}} \log P(D_0 \mid \pi_e)\right], \tag{14}$$

where $P(D_0 \mid \pi_e)$ represents the probability of generating heuristic information $D_0 := \{s_t, a_t, r_t\}_{t=1}^{T}$ in the search strategy π_e, and its distribution can be decomposed into

$$P(D_0 \mid \pi_e) = p(s_1) \prod_{t=1}^{T} \pi_e(a_t \mid s_t) p(s_{t+1} \mid s_t, a_t), \tag{15}$$

where $p(s_{t+1} \mid s_t, a_t)$ is the state transition probability and $p(s_1)$ is the initial distribution. Since the probability $p(s_{t+1} \mid s_t, a_t)$ has no relationship with the exploration strategy parameters θ^{π_e}, a gradient of approximately θ^{π_e} can be obtained as follows:

$$\nabla_{\theta^{\pi_e}} \log P(D_0 \mid \pi_e) = \sum_{t=1}^{T} \nabla_{\theta^{\pi_e}} \log \pi_e(a_t \mid s_t). \tag{16}$$

This value can be calculated from the data that the DDPG algorithm interacts with the environment.

To estimate the difference reward value $R(\pi, D_0)$, a heuristic strategy is adopted in this paper. The heuristic is realized by calling the DDPG algorithm one step or n steps in advance. First, a new action strategy is obtained by calling DDPG based on heuristic information D_0, and then heuristic information is obtained by using the newly obtained action strategy $\pi' = \mathrm{DDPG}(\pi, D_0)$. The cumulative reward value $\widehat{R}_{\pi'}$ of action strategy π' can be estimated through heuristic information D_1 so that the reward of travel value can be estimated as follows:

$$\widehat{R}(\pi, D_0) = \widehat{R}_{\pi'} - \widehat{R}_\pi, \tag{17}$$

where $\widehat{R}_\pi$ is the estimation of the reward function value of action strategy π, which is obtained by the previous iteration of the cycle.

After the difference reward $R(\pi, D_0)$ is obtained, the following parameters of the exploration strategy π_e are updated along the gradient direction of (16) by referring to the parameter θ^{π_e} updating idea of the DDPG algorithm:

```
Initialize exploration strategy $\pi_e$ and DDPG action strategy $\pi$;
Action strategy $\pi$ generates $D_1$ to estimate the reward $\widehat{R}_\pi$ of the strategy $\pi$.
Initialize the replay buffer. $B = D_1$
for $episode = 1, M$ do
    Heuristic strategy $\pi_e$ generate heuristic information $D_0$;
Call DDPG: $\pi' \longleftarrow \text{DDPG}(\pi, D_0)$
    Action strategies $\pi'$ generate heuristic information $D_1$ and calculate rewards $\widehat{R}_{\pi'}$
Calculate the reward of the exploration strategy $\widehat{R}(\pi, D_0) = \widehat{R}_{\pi'} - \widehat{R}_\pi$
    Update network parameters according to the gradient of exploration strategy: $\theta^{\pi_e} \longleftarrow \theta^{\pi_e} + \eta \nabla_{\theta^{\pi_e}} \log P(D_0 \mid \pi_e) \widehat{R}(\pi, D_0)$
Add heuristic information $D_0$ and $D_1$ together to the replay buffer: $B \longleftarrow B \cup D_0 \cup D_1$
Update action strategy $\pi$ based on heuristic information in replay buffer, calculate new $\widehat{R}_\pi$
End for
```

Algorithm 1: Heuristic DDPG pseudocode.

$$\theta^{\pi_e} \longleftarrow \theta^{\pi_e} + \eta \widehat{R}(D_0) \sum_{t=1}^{T} \nabla_{\theta^{\pi_e}} \log \pi_e (a_t \mid s_t). \tag{18}$$

After the exploration strategy π_e is updated, the heuristic information D_0 and D_1 are added to the experience replay space, that is, $B \longleftarrow B \cup D_0 \cup D_1$. The action strategy π is updated through the DDPG algorithm after sampling from the experience replay space, that is, $\pi \longleftarrow \text{DDPG}(\pi, B)$. The specific Algorithm 1 process is as follows:

Although the improvement of the DDPG algorithm in this section increases the amount of computation in the calculation of heuristic data D_1, the high efficiency D_1 brought to the overall algorithm can compensate for this shortcoming. The construction of DDPG can not only be used to evaluate the improvement of learning performance but also participate in the update of action strategy π.

3.2. Performance Test of Algorithm. To test the performance of the improved algorithm proposed in this paper, Half Cheetah-v1, a Mujoco robot control environment in the OpenAI Gym toolkit, is selected as the test environment. Considering that the UCAV air combat in this paper is a decision-making process with air combat status information as input, without considering the image input, the RAM version of the environment is chosen and the state information is obtained directly, rather than the RGB version with the game graphics as input. To reflect the performance of the algorithm, 20 Monte Carlo simulations were performed for each algorithm. The $Q(s_0)$ curves of the three algorithms are shown in Figure 2.

In Figure 2, the ordinate ‘performance’ is the cumulative reward value of completing a task. The areas covered by red, dark blue, and light blue are the heuristic DDPG algorithm proposed in this paper, the PPO algorithm and the traditional DDPG algorithm, respectively, after 20 Monte Carlo simulations of the $Q(s_0)$ distribution. The solid lines of the three colors are the average values of their distribution data. Through comparison, it can be seen that the heuristic DDPG algorithm has a stronger scoring ability after strengthening the exploration performance, which reflects a stronger ability to explore the optimal solution. Simulation comparison tests verify the effectiveness and superiority of the proposed algorithm.

4. Maneuver Decision Scheme Design

To increase the generalization ability of strategic networks, this paper considers the relative relationship between the enemy and the UCAV in the selection of state variables and takes the three-dimensional relative position coordinates of two aircraft, the relative flight speed, AA, and ATA as state variables; that is, the state variables are

$$s = [\Delta x, \Delta y, \Delta z, V, V_e, \Delta V, \gamma, \gamma_e, \psi, \psi_e \text{AA}, \text{ATA}], \tag{19}$$

where $\Delta x, \Delta y, \Delta z$ and ΔV are the relative position coordinates and relative flight speed of the two aircraft, respectively.

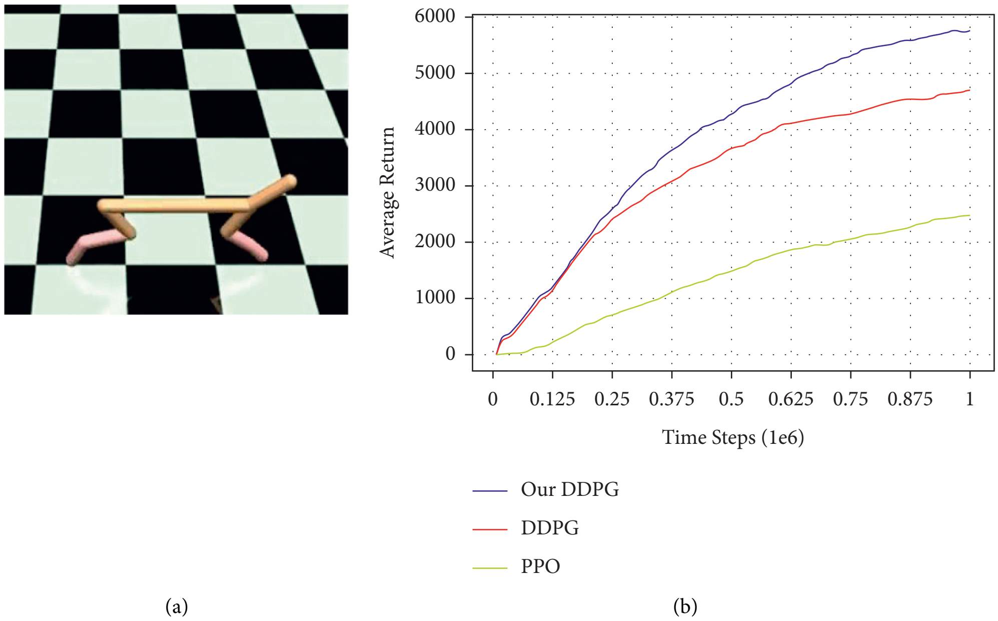
In the selection of the action, it is designed as the variation of the model control variable $u = [\kappa, \alpha, \phi]$ of the UCAV platform in consideration of generating the smoothness of the maneuver trajectory.

$$a_i = [\Delta\kappa_i, \Delta\alpha_i, \Delta\phi_i], \tag{20}$$

where $\Delta\kappa_i, \Delta\alpha_i, \Delta\phi_i$ represents the change in throttle lever, change in the angle of attack, and change in roll angle, respectively. The control variable a_t of the strategy network output at time t acts on the environment to produce the state s_{t+1} at the next step. Together with the state s_t at time t, the reward r_t constitutes the state transfer information $[s_t, a_t, r_t, s_{t+1}]$.

5. Simulation and Analysis

5.1. Network and Parameter. Combined with the maneuver decision scheme, the actor network and critic network structures in our algorithm are designed. The structures of the actor network and actor target network are the same, and the input value is the state input $s = [\Delta x, \Delta y, \Delta z, V, V_e, \Delta V, \gamma, \gamma_e, \psi, \psi_e \text{AA}, \text{ATA}]$, so the input layer with 12 nodes is set. The output is the maneuvering action control variable $a_i = [\Delta\kappa_i, \Delta\alpha_i, \Delta\phi_i]$ in the current state; therefore, the number of nodes in the output layer is 3. The structure of actor network is shown in Table 1.



(a)

(b)

Figure 2: Comparison of the algorithm cumulative reward curve in the half Cheetah environment. (a) Half Cheetah environment interface. (b) Comparison of the algorithm cumulative reward curve.

Table 1: Actor/Actor target network structure.

Layer	Input	Activation function	Output
Input layer	1 × 12	None	128
Full connection layer 1	128	tanh	128
Full connection layer 2	128	tanh	128
Output layer	128	Linear	1 × 3

The critic network has the same structure as the critic target network. The input value is the combination of the state input $s = [\Delta x, \Delta y, \Delta z, V, V_e, \Delta V, \gamma, \gamma_e, \psi, \psi_e \mathrm{AA}, \mathrm{ATA}]$ and the change rate of the control value $a_i = [\Delta \kappa_i, \Delta \alpha_i, \Delta \phi_i]$ generated by the current actor network. Therefore, the input layer of 15 nodes is constructed, and the network output is the action value function Q. The structure of critic network is shown in Table 2.

The neural network training platform is a TensorFlow open-source deep learning computing platform based on an NVIDIA GeForce GTX 1080Ti GPU in an Ubuntu 16.04 system. The specific hyperparameter settings of the H-DDPG algorithm are shown in Table 3.

5.2. Initial Situation Setting. To verify the effectiveness of the algorithm, it is assumed that the enemy fighter and the UCAV adopt the same platform model and the same maneuverability constraints. The decision method of enemy adopts the rolling time-domain maneuver decision method proposed in reference [36]. In order to reflect the antagonism of air combat, we suppose the two sides enter the battle in a head-on encounter and set the UCAV height slightly lower than the enemy aircraft at a disadvantage. The simulation initialization state is shown in Table 4.

Table 2: Critic/Critic target network structure.

Layer	Input	Activation function	Output
Input layer	1 × 15	None	128
Full connection layer 1	128	tanh	128
Full connection layer 2	128	tanh	128
Output layer	128	Linear	1 × 1

Table 3: Hyperparameter setting of heuristic DDPG algorithm.

Parameter	Parameter value
Size of replay buffer D	50000
Size of minibatch N_T	64
Actor learning rate α	0.0001
Critic learning rate β	0.001
Discount rate γ	0.99

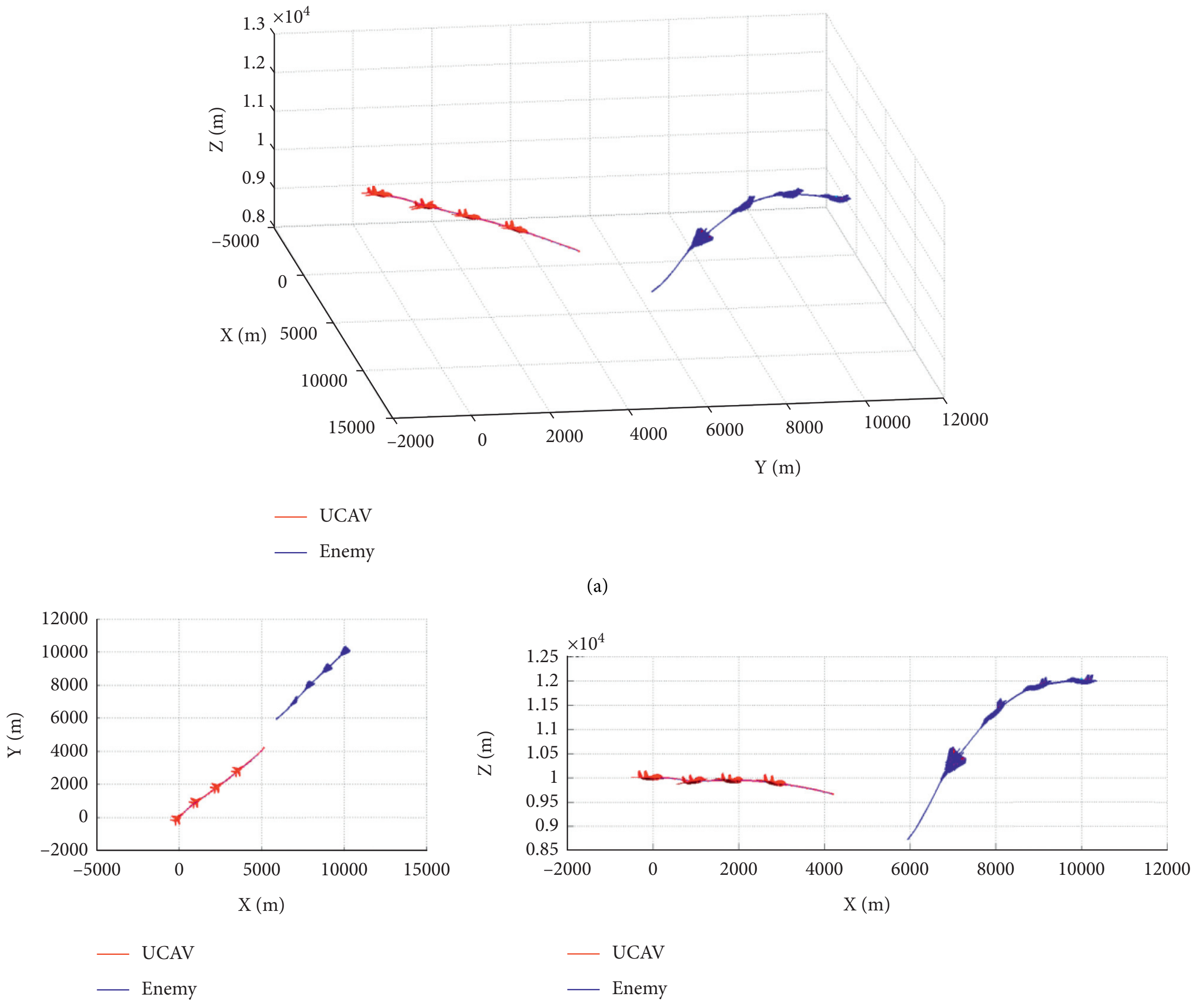
Table 4: Initial state of UCAV and enemy.

	x (m)	y (m)	z (m)	v (m/s)	γ (°)	ψ (°)	Max step (s)
UCAV	0	0	10000	250	0	45	200
Enemy	10000	10000	12000	200	0	−135.	

5.3. Enemy Making Random Maneuvers

Case 1. The UCAV weapon is stronger in the head-on situation, and the launching distance of the UCAV weapon is superior. The winning conditions of the UCAV are as follows: $\mathrm{ATA} \leq 30^\circ \quad \&200\,\mathrm{m} \leq D \leq 2500\,\mathrm{m} \quad \&0\,\mathrm{m} \leq h_r - h_b \leq 1000\,\mathrm{m}$. The air battle trajectory is shown in Figure 3.

As shown in Figure 3, the enemy aircraft chooses to dive downward through a random maneuver. The UCAV approaches the enemy aircraft in flat flight and then dives



(a)

(b)

(c)

FIGURE 3: Air combat trajectory. (a) 3D view of air combat trajectory. (b) Aerial view of the air combat trajectory. (c) Horizontal view of the air combat trajectory.

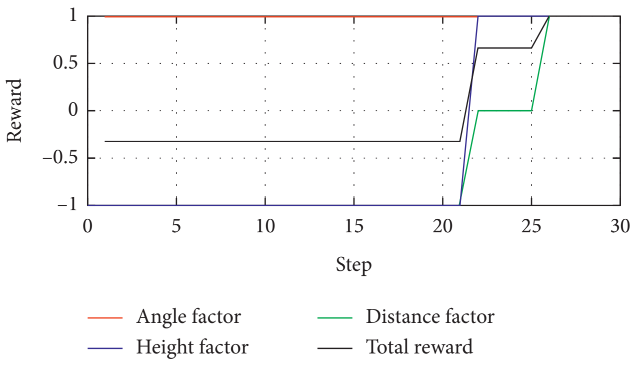


FIGURE 4: Curves of reward factors.

downward to gain altitude superiority. It finally gives priority to meet the weapon firing conditions and launches missiles to win air battles. It can be seen from the changes of reward factors in Figure 4 that at the beginning of the battle, the UCAV had already met the maximum angle reward factor, approached the enemy aircraft through flat flight and dove at 21 s to obtain the height advantage, and reached the weapon launch range at 26 s. At this time, all the reward functions achieved 1, meeting the winning conditions in the air battle. Figure 5 shows the curve of the average cumulative reward function value of training for this air combat mission. Each epoch on the horizontal axis contains 200 training missions, and the ordinate axis is the average cumulative reward value obtained for every 200 missions.

Case 2. The enemy weapon is stronger when the firing distance of the enemy weapon is dominant; the UCAV winning conditions are as follows: $\mathrm{ATA} \leq 30^\circ$ $\&\mathrm{AA} \geq 90^\circ$ $\&200\,\mathrm{m} \leq D \leq 2500\,\mathrm{m}$ $\&0\,\mathrm{m} \leq h_r - h_b \leq 1000\,\mathrm{m}$. Air battle trajectory is shown in Figure 6.

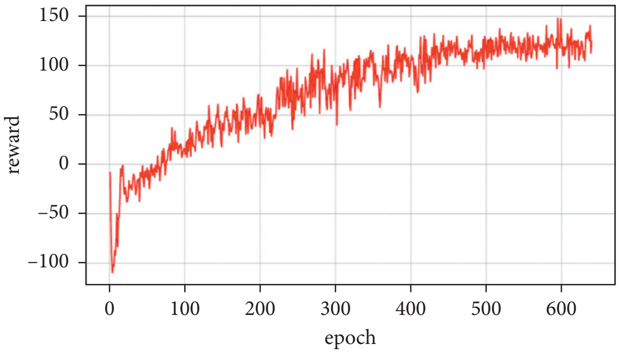
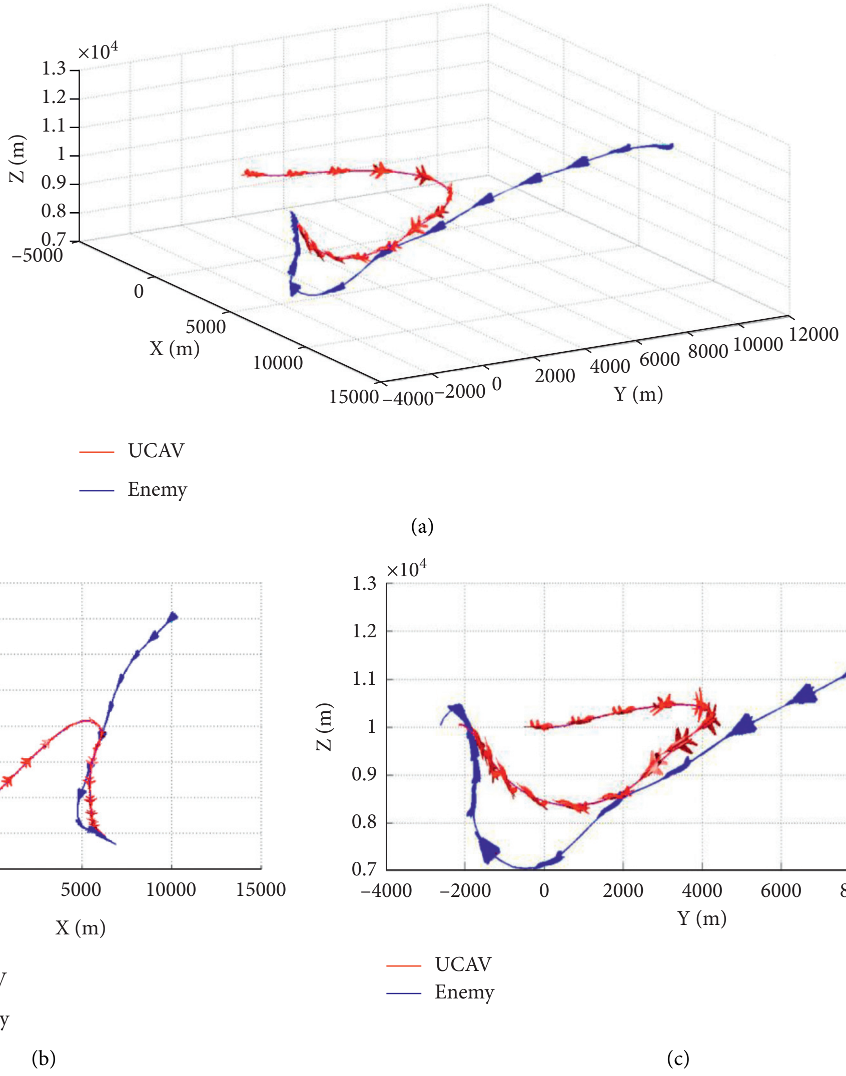


Figure 5: Cumulative reward curve.



(a)

(b)

(c)

Figure 6: Air combat trajectory. (a) 3D view of air combat trajectory. (b) Aerial view of the air combat trajectory. (c) Horizontal view of the air combat trajectory.

As seen from Figure 6, the enemy swooped down to the left through random maneuvers and then climbed to the left. Due to the low altitude at the beginning, the UCAV first shortened the distance with the enemy and improved the height advantage by climbing. Before entering the enemy attack range, it made a sharp right turn. The UCAV achieves a height advantage by successfully diving behind the enemy's tail and by turning to the right with a small overload. Finally,

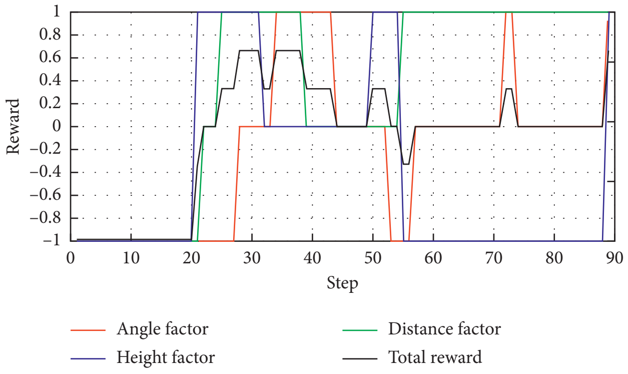


Figure 7: Curves of reward factors.

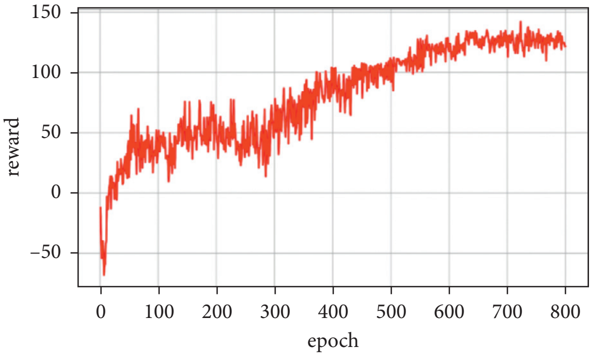
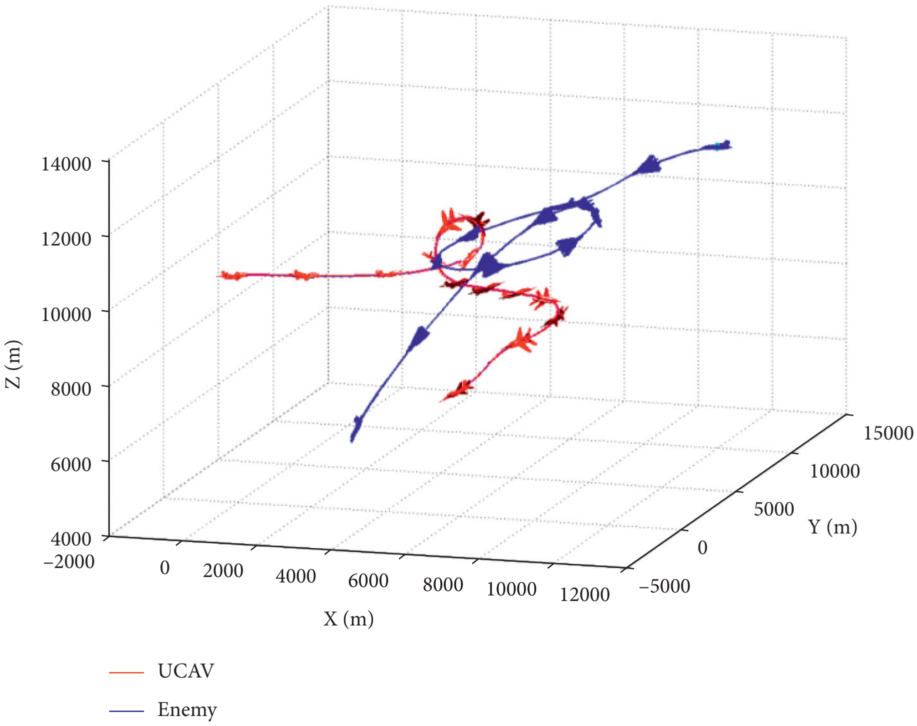
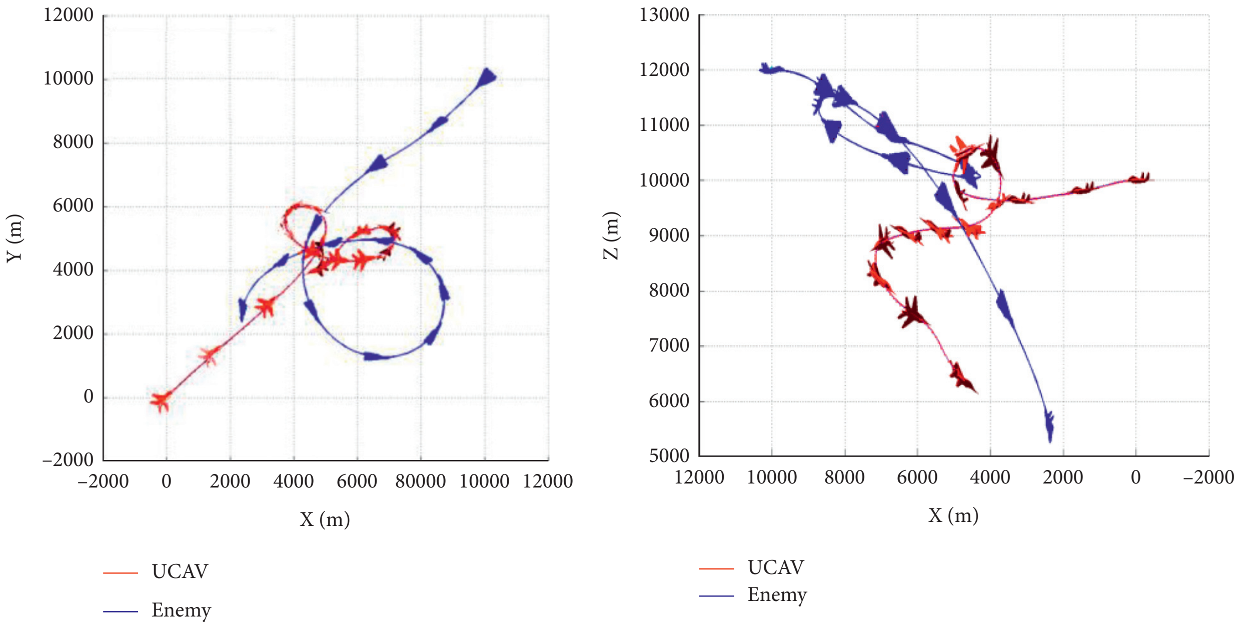


Figure 8: Cumulative reward curve.



(a)

Figure 9: Continued.



(b)

(c)

Figure 9: Air combat trajectory. (a) 3D view of air combat trajectory. (b) Aerial view of the air combat trajectory. (c) Horizontal view of the air combat trajectory.

the UCAV achieves a height advantage by continuously following the enemy with a small overload deceleration and pulling up to the left to meet the rear attack conditions and win the air battle. As seen from the changes in reward factors in Figure 7, at the early stage of air combat, due to the long distance, low altitude, and enemy meeting the attack angle, all reward factors are −1. With the implementation of a large overload maneuver, the UCAV gradually obtains each situation advantage and finally meets the weapon launch conditions at 89 s by tracking the enemy aircraft. At the beginning of the battle, the UCAV has already met the maximum angle reward. It approaches the enemy through flat flight and dives at 21 s to gain an altitude advantage. At 26 s, the UCAV reaches the weapon launch range. Figure 8 shows the curve of the cumulative reward value during the task training process in this section.

5.4. *Enemy Making Intelligent Maneuvers.*

Under this task, the enemy makes intelligent maneuvers using the rolling time-domain maneuver decision method proposed in reference [36], which is adopted to traverse 216 trial maneuvers generated by the discrete variation of control variables, and the maneuvers corresponding to the optimal membership function value are selected and executed through the membership function of the air combat situation.

Mission setting: the enemy weapon is stronger. Under this mission, the UCAV adopts the rear attack mode to attack the enemy aircraft. The enemy does not need to go around the rear but adopts an omnidirectional attack strategy. The UCAV winning conditions are as follows: $\text{ATA} \leq 30^\circ \quad \&\text{AA} \geq 90^\circ \quad \&200\,\text{m} \leq D \leq 2500\,\text{m} \quad \&0\,\text{m} \leq h_r - h_b \leq 1000\,\text{m}$. After training, the maneuvering UCAV strategy gradually converges. Under this strategy, the air battle trajectory is shown in Figure 9.

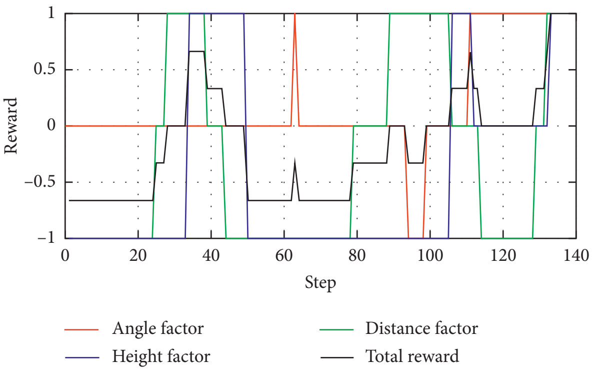


Figure 10: Curve of reward factors.

As shown in Figure 9, after the two sides entered the air combat airspace in a head-on encounter, the enemy aircraft adopted an accelerated dive maneuver at a high altitude to quickly approach the UCAV to meet the priority conditions of weapon launch. The UCAV first adopted an accelerated flat flight to quickly shorten the distance between the two sides and pulled off to the upper left and right of the enemy aircraft before entering the enemy missile attack range. The enemy aircraft lost altitude advantage due to the rapid speed of the dive and then leveled out and pulled up to the left, regained altitude advantage and turned, but due to the climb maneuver reduced speed resulting in a larger turning radius. At this point, the UCAV performs a loop to increase its speed and power advantage and finally wins by following the enemy aircraft to reach the weapon firing conditions.

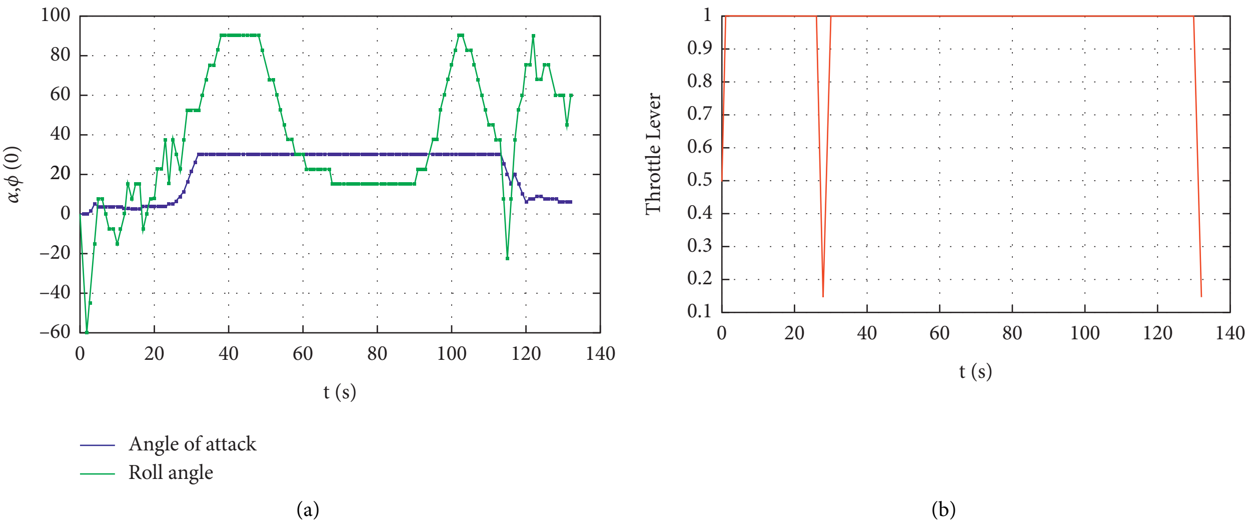


Figure 11: Curve of control variable. (a) Curves of the angle of attack and roll angle. (b) Curve of the throttle lever.

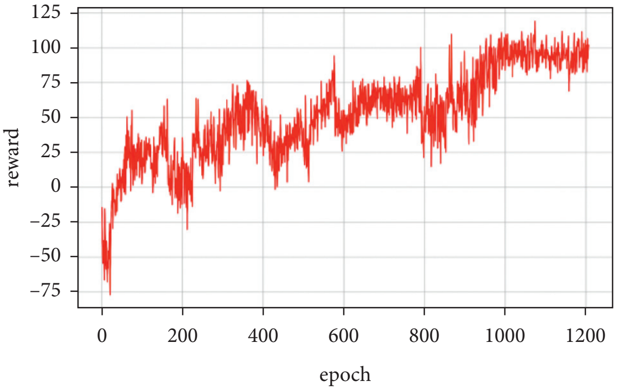


Figure 12: Curve of average cumulative reward.

Figure 10 shows the reward function curve. It can be seen from the figure that in the 30 s–50 s range, the UCAV gained a temporary altitude advantage through a loop and then repositioned below the enemy aircraft until 105 s the UCAV remained level and dived, adjusting the angle by sacrificing its altitude advantage. After that, the UCAV succeeded in placing itself behind the enemy aircraft at 110 s gaining altitude and angle advantages. After that, the UCAV closed the distance by adjusting its attitude and defeated the enemy aircraft at 133 s. Figure 11 shows the curve of control variable of UCAV. Figure 12 shows the curve of the average cumulative reward value of training for this air combat mission. Each epoch on the horizontal axis contains 200 training missions, and the ordinate axis is the average cumulative reward value obtained for every 200 missions.

6. Conclusions

In this paper, a continuous action space air combat decision-making technology for UCAVs based on reinforcement learning is studied. Starting with the UCAV continuous action space model, a continuous action space air combat model is constructed based on the aerodynamic parameters of the unmanned stealth fighter. Focusing on the problems of weak exploration ability and low data utilization rate of the DDPG algorithm, a heuristic exploration strategy was introduced to propose a heuristic DDPG algorithm to improve the exploration ability of the original algorithm. The effectiveness and superiority of the proposed algorithm are verified by the Monte Carlo simulation in a typical continuous motion control environment (Half Cheetah). In the simulation verification stage, two subtasks with increasing difficulty, random maneuvers, and intelligent attack maneuvers are adopted for enemy aircraft, and the results show that the method presented in this paper can accomplish maneuver decisions under various tasks as well.

Data Availability

All data included in this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] L. Fu, F. Xie, D. Wang, and G. Meng, “The overview for UAV air-combat decision method,” in *Proceedings of the 26th Chinese Control and Decision Conference (2014 CCDC)*, IEEE, Changsha, China, May 2014.

[2] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, MIT press, Cambridge, MA, USA, 2018.

[3] B. R. Kiran, I. Sobh, V. Talpaert et al., “Deep reinforcement learning for autonomous driving: a survey,” *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–18, 2021.

[4] Y. Li, “Deep reinforcement learning: an overview,” 2017, https://arxiv.org/abs/1701.07274.

[5] V. Mnih, K. Kavukcuoglu, D. Silver et al., “Playing atari with deep reinforcement learning,” 2013, https://arxiv.org/abs/1312.5602.

[6] F. Agostinelli, G. Hocquet, S. Singh, and P. Baldi, "From reinforcement learning to deep reinforcement learning: an overview," *Braverman Readings in Machine Learning. Key Ideas from Inception to Current State*, pp. 298–328, Springer, New York, NY, USA, 2018.

[7] X. Han, J. Wang, J. Xue, and Q. Zhang, "Intelligent decision-making for 3-dimensional dynamic obstacle avoidance of UAV based on deep reinforcement learning," in *Proceedings of the 2019 11th International Conference on Wireless Communications and Signal Processing (WCSP)*, IEEE, Xi'an, China, October 2019.

[8] V. Mnih, K. Kavukcuoglu, D. Silver et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[9] D. Silver, J. Schrittwieser, K. Simonyan et al., "Mastering the game of Go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[10] P. Liu and Y. Ma, "A deep reinforcement learning based intelligent decision method for UCAV air combat," in *Proceedings of the Asian Simulation Conference*, pp. 274–286, Springer, Melaka, Malaysiaa, October 2017.

[11] W. Ma, H. Li, Z. Wang, Z. Huang, Z. Wu, and X. Chen, "Close air combat maneuver decision based on deep stochastic game," *Systems Engineering and Electronics*, vol. 9, p. 14, 2020.

[12] X. Zhang, G. Liu, C. Yang, and J. Wu, "Research on air combat maneuver decision-making method based on reinforcement learning," *Electronics*, vol. 7, no. 11, p. 279, 2018.

[13] Q. Yang, J. Zhang, G. Shi, J. Hu, and Y. Wu, "Maneuver decision of UAV in short-range air combat based on deep reinforcement learning," *IEEE Access*, vol. 8, pp. 363–378, 2019.

[14] X. Xu, D. Hu, and X. Lu, "Kernel-based least squares policy iteration for reinforcement learning," *IEEE Transactions on Neural Networks*, vol. 18, no. 4, pp. 973–992, 2007.

[15] Z. Wang, H. Li, H. Wu, and Z. Wu, "Improving maneuver strategy in air combat by alternate freeze games with a deep reinforcement learning algorithm," *Mathematical Problems in Engineering*, vol. 2020, Article ID 7180639, 7 pages, 2020.

[16] T. P. Lillicrap, J. J. Hunt, A. Pritzel et al., "Continuous control with deep reinforcement learning," 2015, https://arxiv.org/abs/1509.02971.

[17] D. Silver, G. Lever, N. Heess, D. Thomas, W. Daan, and R. Martin, "Deterministic policy gradient algorithms," in *Proceedings of the International conference on machine learning*, pp. 387–395, PMLR, Beijing China, June 2014.

[18] M. Wang, L. Wang, T. Yue, and H. Liu, "Influence of unmanned combat aerial vehicle agility on short-range aerial combat effectiveness," *Aerospace Science and Technology*, vol. 96, Article ID 105534, 2020.

[19] M. Wang, L. Wang, and T. Yue, "An application of continuous deep reinforcement learning approach to pursuit-evasion differential game," in *Proceedings of the 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, pp. 1150–1156, IEEE, Chengdu, China, March 2019.

[20] Q. Yang, Y. Zhu, J. Zhang, S. Qiao, and J. Liu, "UAV air combat autonomous maneuver decision based on DDPG algorithm," in *Proceedings of the 2019 IEEE 15th International Conference on Control and Automation (ICCA)*, pp. 37–42, IEEE, Edinburgh, UK, July 2019.

[21] B. Li and Y. Wu, "Path planning for UAV ground target tracking via deep reinforcement learning," *IEEE Access*, vol. 8, pp. 29064–29074, 2020.

[22] S. You, M. Diao, L. Gao, F. Zhang, and H. Wang, "Target tracking strategy using deep deterministic policy gradient," *Applied Soft Computing*, vol. 95, Article ID 106490, 2020.

[23] A. E. Sallab, M. Abdou, E. Perot, and S. Yogamani, "Deep reinforcement learning framework for autonomous driving," *Electronic Imaging*, vol. 2017, no. 19, pp. 70–76, 2017.

[24] S. Wang, D. Jia, and X. Weng, "Deep reinforcement learning for autonomous driving," 2018, https://arxiv.org/abs/1811.11329.

[25] O. Vinyals, T. Ewalds, S. Bartunov et al., "Starcraft ii: a new challenge for reinforcement learning," 2017, https://arxiv.org/pdf/1710.03131.pdf.

[26] P. Peng, Y. Wen, Y. Yang et al., "Multiagent bidirectionally-coordinated nets: emergence of human-level coordination in learning to play starcraft combat games," 2017, https://arxiv.org/abs/1703.10069.

[27] T. Zhao, L. Kong, Y. Han, D. Ren, and Y. Chen, "Review of model-based reinforcement learning," *Journal of Frontiers of Computer Science and Technology*, vol. 14, no. 06, pp. 918–927, 2020.

[28] Q. Zhihui, L. Ning, L. Xiaotong, L. Xiulei, and Q. Dong, "OverviewofResearchonModel-freeReinforcementLearning," *Computer Science*, vol. 48, no. 03, pp. 180–187, 2021.

[29] H. Liu, Y. Pan, S. Li, and Y. Chen, "Synchronization for fractional-order neural networks with full/under-actuation using fractional-order sliding mode control," *International Journal of Machine Learning and Cybernetics*, vol. 9, no. 7, pp. 1219–1232, 2018.

[30] H. Liu, S. Li, G. Li, and H. Wang, "Robust adaptive control for fractional-order financial chaotic systems with system uncertainties and external disturbances," *Information Technology and Control*, vol. 46, no. 2, pp. 246–259, 2017.

[31] Storm Shadow UCAV performance, "Storm Shadow UCAV performance," 1994, http://www.aerospaceweb.org/design/ucav/.

[32] T. Xu, Y. Wang, and C. Kang, "Tailings saturation line prediction based on genetic algorithm and BP neural network," *Journal of Intelligent and Fuzzy Systems*, vol. 30, no. 4, pp. 1947–1955, 2016.

[33] Z. Zhao, Q. Xu, and M. Jia, "Improved shuffled frog leaping algorithm-based BP neural network and its application in bearing early fault diagnosis," *Neural Computing & Applications*, vol. 27, no. 2, pp. 375–385, 2016.

[34] Y. C. Lin, D. D. Chen, M. S. Chen, X. M. Chen, and J. Li, "A precise BP neural network-based online model predictive control strategy for die forging hydraulic press machine," *Neural Computing & Applications*, vol. 29, pp. 1–12, 2016.

[35] H. Wang, Y. Wang, and K. E. Wen-Long, "An intrusion detection method based on spark and BP neural network," *Computer Knowledge & Technology*, vol. 13, no. 6, pp. 157–160, 2017.

[36] K. Dong and C. Huang, "Autonomous air combat maneuver decision using Bayesian inference and moving horizon optimization," *Journal of Systems Engineering and Electronics*, vol. 29, no. 1, pp. 86–97, 2018.