electronics

MDPI

Article

Autonomous Maneuver Decision Making of Dual-UAV Cooperative Air Combat Based on Deep Reinforcement Learning

Jinwen Hu [1], Luhe Wang [1], Tianmi Hu [1], Chubing Guo [2,3,*] and Yanxiong Wang [4]

[1] School of Automation, Northwestern Polytechnical University, Xi'an 710072, China; hujinwen@nwpu.edu.cn (J.H.); luhe_wang@foxmail.com (L.W.); 2021202403@mail.nwpu.edu.cn (T.H.)
[2] Key Laboratory of Data Link Technology, The 20th Research Institute of China Electronics Technology Group Corporation, Xi'an 710068, China
[3] School of Artificial Intelligence, Xidian University, Xi'an 710071, China
[4] AVIC Chengdu Aircraft Design and Research Institute, Chengdu 610091, China; wangyanxiongli@163.com
* Correspondence: m18191412336@163.com

Abstract: Autonomous maneuver decision making is the core of intelligent warfare, which has become the main research direction to enable unmanned aerial vehicles (UAVs) to independently generate control commands and complete air combat tasks according to environmental situation information. In this paper, an autonomous maneuver decision making method is proposed for air combat by two cooperative UAVs, which is showcased by using the typical olive formation strategy as a practical example. First, a UAV situation assessment model based on the relative situation is proposed, which uses the real-time target and UAV location information to assess the current situation or threat. Second, the continuous air combat state space is discretized into a 13 dimensional space for dimension reduction and quantitative description, and 15 typical action commands instead of a continuous control space are designed to reduce the difficulty of UAV training. Third, a reward function is designed based on the situation assessment which includes the real-time gain due to maneuver and the final combat winning/losing gain. Fourth, an improved training data sampling strategy is proposed, which samples the data in the experience pool based on priority to accelerate the training convergence. Fifth, a hybrid autonomous maneuver decision strategy for dual-UAV olive formation air combat is proposed which realizes the UAV capability of obstacle avoidance, formation and confrontation. Finally, the air combat task of dual-UAV olive formation is simulated and the results show that the proposed method can help the UAVs defeat the enemy effectively and outperforms the deep Q network (DQN) method without priority sampling in terms of the convergence speed.

Keywords: air combat; maneuver decision; reinforcement learning; priority sampling; situation assessment

check for updates

Citation: 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. https://doi.org/10.3390/electronics11030467

Academic Editor: Arturo de la Escalera Hueso

Received: 13 January 2022
Accepted: 31 January 2022
Published: 5 February 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

With the progress of combat mode and the expansion of combat scale, modern air combat gradually extends from the within visual range (WVR) air combat to the beyond visual range (BVR) air combat [1]. Unmanned aerial vehicles (UAVs) are more and more widely used in military tasks such as investigation, monitoring and target attack [2] because of their low cost, strong mobility and high concealment. Due to the limitations of a single UAV's mission and combat capability, autonomous multi-UAV cooperative air combats have become a research hotspot in recent years [3].

Autonomous air combat maneuver decision making refers to the process of automatically generating the maneuver control decisions of UAVs based on mathematical optimization and artificial intelligence [4], which requires that UAVs have independent capabilities for autonomous sensing, information processing and decision-making abilities [5]. At present, there are many autonomous decision making methods for UAV air combat maneuver control, which can be roughly divided into two categories: Analytical

solution methods and intelligent optimization methods. Analytical solutions include matrix games, influence diagrams, differential games, etc. Matrix game method [6] uses a linear program to find the optimal solution in a short decision time, which cannot guarantee global optimization. In addition, this method needs to introduce expert experience to design the income matrix in line with the actual air combat, which is time consuming and laborious. The maneuver decision making method based on influence graph [7,8] can intuitively express the air combat model of key factors such as threat, situation and pilot's subjective preference through influence graph, but it is difficult to obtain the analytical solution by this method, and the calculation time is long. So the influence graph cannot meet the real-time performance of air combat decision making. In [9,10], the knowledge of game theory is introduced into air combat to realize the one-to-one autonomous maneuver decision of UAV, and the method in [9] solved the curse of dimension problem by using fuzzy theory. However, the method in [9] does not take into account the current state of the enemy aircraft when designing the state, and lacks confrontation simulation results. The method used in [10] has complex calculation and poor real-time performance, and is not suitable for the high dynamic environment. The differential game method [11] is the most practical decision making model for studying air combat games. However, due to the difficulty in setting the performance function, the huge amount of calculation and the ill condition after the model is simplified, although the differential game theory has been developed for many years, it has not produced a more reasonable description of actual air combat. In [12], researchers propose an air-to-air confrontation method based on uncertain interval information, but only analyze the influence of different factors on air combat effect, and do not consider the maneuver model of UAV. In addition, the method in [12] needs to calculate the revenue and expenditure matrix, which is cumbersome and has low real-time performance. In short, the analytical solution method needs to accurately model and describe the decision model, which cannot be applied to the air combat scene without model or incomplete environment information, and cannot meet the requirements of intelligent air combat.

Intelligent optimization methods mainly include expert system method [13], neural network method [14] and some other optimization algorithms such as fuzzy tree, particle swarm optimization [15] and reinforcement learning. The maneuver decision making method based on the expert system has mature technology and is easy to implement, but its disadvantage is that the establishment of the knowledge base is complex and it is difficult to fully cover all air combat situations. The maneuver decision making based on the neural network has strong robustness and learning ability, but it is a supervised learning method, which cannot be applied without a training set. While the application of neural networks in air combat decision making has practical value, it is worth further exploration and improvement. A maneuver decision making method based on the fuzzy tree is proposed in [16], which can guide UAVs to make more targeted maneuver decisions according to a real-time combat situation. However, the design of fuzzy tree is difficult and the hyper-parameters are complex and diverse, and expert experience needs to be introduced. In [17], researchers use dynamic game and particle swarm optimization to realize multi-agent task allocation and confrontation. This method will make the payment matrix of both parties become huge with the increase of the number of agents, and the income matrix needs to be designed manually. Therefore, subjective factors have a great impact on the experimental results. In addition, the simulation result is a two-dimensional plane without considering the maneuver control model of UAV, which is very different from the real maneuver.

Reinforcement learning [18] is an intelligent optimization method that uses the "trial and error" method to interact with the environment, learns from the environment and improves the performance with time [19]. It overcomes the shortcomings of complex modeling, difficult sample marking and cumbersome solutions of other methods, and can produce a series of decision sequences considering long-term effects through self-interactive training without manual intervention. It is a feasible modeling method for autonomous

decision making of air combat maneuvers in artificial intelligence [20,21]. The autonomous maneuver decision making problem of air combat based on deep reinforcement learning is studied in [22–26]. In [22,23], researchers verify the performance of the algorithm by building a high simulation combat platform, and has good experimental results. However, the reward function in [23] is sparse, and the reward is 0 in most states of each round, which is not conducive to network training. The robust multi-agent reinforcement learning (MARL) algorithm framework is used in [24] to solve the problem that the reinforcement learning algorithm cannot converge due to the unstable environment in the training process. However, the simulation environment in [24] is a two-dimensional plane and the simulation test initialization is fixed, which makes it hard to be applied in the dynamic confrontation scenarios. Many aspects of UAV situation assessment is considered in [25], but UAV uses absolute coordinates as the state input, which is highly dependent on spatial characteristics. In [26], researchers use Monte Carlo reinforcement learning to carry out research. The biggest problem is that the agent needs to complete a complete air combat process to evaluate the reward. Moreover, the above references consider the one-to-one air combat scenario, which has limited reference value for the research of multi-aircraft cooperative autonomous control. There are few studies on multi-agent confrontation using reinforcement learning algorithms. In [27], a novel autonomous aerial combat maneuver strategy generation algorithm based on state-adversarial deep deterministic policy gradient algorithm (SA-DDPG) is proposed, which considers the error of the airborne sensor and uses a reward shaping method based on maximum entropy inverse reinforcement learning algorithm. However, the reliance on expert knowledge in the design of reward functions in this paper is not conducive to extension to more complex air combat environments. In [28] researchers propose an air combat decision-making model based on reinforcement learning framework, and use long short-term memory (LSTM) to generate a new displacement prediction. However, the simulation experiments in [28] rely on an off-the-shelf game environment, which is not conducive to the extension of the study and it studies the air combat problem of searching for observation station in a non-threatening environment, which differs significantly from the air combat mission of this paper. Based on the MARL method, the simulation in [29] of multiple UAVs arriving at their destinations from any departure points in a large-scale complex environment is realized. However, the modeling environment is planar, and a sparse reward function is used, and only distance penalty is considered. The method for maneuver decision making of multi-UAV formation air combat in [30] is robust. However, there are no simulation results, and there are only three maneuver behaviors. The deep deterministic policy gradient (DDPG) algorithm is used in [31] to realize the maneuver decision of the dynamic change of UAV quantity in the process of swarm air combat. The algorithm has robustness and expansibility, but the waypoint model is used in this paper, which cannot describe the maneuver characteristics of UAV. Other researches on air combat based on reinforcement learning, the intelligent decision making technology for multi-UAV prevention and control proposed in [21,32]. The control method of UAV autonomous avoiding missile threat based on deep reinforcement learning introduced in [33,34]. Deep reinforcement learning is used in [35] to build an intelligent command framework and so on. These studies focus on the feasibility of reinforcement learning methods in solving some air combat problems, which has little correlation with our autonomous maneuver decision making problem, but provides some ideas for our research. In addition, uniform sampling is used in [21–26,29,30,32], which means that the probability of all experiences in the experience pool being extracted and utilized is the same, thus ignoring the different importance of each experience, resulting in long training time and extremely unstable.

Generally speaking, at present, the research on air combat maneuver decision making based on reinforcement learning mainly focuses on single UAV confrontation tasks, and the research on multi-UAV confrontation and multi-UAV cooperation are in the initial exploration stage. These studies have one or more of the following problems: Dimension explosion, rewards

are sparse and delayed, simple simulation environment, lack of maneuver model, incomplete situation assessment and random uniform sampling leads to slow training.

In this paper, an autonomous maneuver decision making method based on deep reinforcement learning is proposed for dual-UAV cooperative air combat. The main contributions are as follows. First, aiming at the problems of dimension explosion, sparse and delayed rewards and incomplete situation assessment, we discretize the continuous air combat state space into 13 dimensions for dimension reduction and quantitative description of air combat states. Then the situation assessment model is established based on the relative location between the UAV and the target. Second, a reward function is designed according to the situation assessment results which includes the real-time gain due to maneuver and the final combat winning/losing gain. Such a design helps to solve the problem of sparse and delayed reward in the games of long-running time for ending. Third, aiming at the problem of slow convergence caused by random sampling in conventional DQN learning, an improved priority sampling strategy is proposed to accelerate the convergence of the DQN network training. Fourth, we apply and modify the designed autonomous maneuver decision making method for the typical task of dual-UAV olive formation air combat, which enables the UAVs to own the capability of collision avoidance, formation and confrontation. Finally, the proposed method is validated by simulation using practical fixed-wing UAV models and compared with the DQN learning method without priority sampling. The simulation results show that our method can make the two UAVs defeat the enemy effectively and improve the performance in terms of the convergence speed.

The following parts are arranged as follows: Section 2 is the problem formulation. Section 3 is the air combat confrontation algorithm based on deep reinforcement learning. Section 4 is the description of typical air combat scenarios and the design of dual-UAV cooperative autonomous maneuver strategy. Section 5 conducts simulation analysis. Section 6 summarizes the full paper.

2. Problem Formulation

This paper studies the autonomous maneuver control decision of multi-UAV BVR cooperative tracking and close combat. Maneuver control model and situation assessment are the premises of UAV maneuver decision. Therefore, in the following sections, we will elaborate on UAV autonomous maneuver decision making from three aspects: Maneuver control model, situation assessment and maneuver control decision.

2.1. UAV Dynamic Model

As shown in Figure 1, in the ground coordinate system, the ox, oy and oz are the east, north and vertical directions respectively. The motion model of UAV in the coordinate system is given by

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

where x, y and z represent the position of the UAV in the coordinate system, v represents the current speed direction of the UAV, $\dot{x}$, $\dot{y}$ and $\dot{z}$ represent the change rate of v in the three coordinate axis directions, v' is the projection of v on the xoy plane, γ is the angle between v' and v, and γ represents the pitch angle, ψ is the angle between v' and oy axis, and ψ represents the yaw angle. In the same coordinate system, the dynamic model of UAV can be expressed as

$$\begin{cases} \dot{v} = g(n_x - \sin\gamma), \\ \dot{\gamma} = \frac{g}{v}(n_z\cos\mu - \cos\gamma), \\ \dot{\psi} = \frac{gn_z\sin\mu}{v\cos\gamma}, \end{cases} \tag{2}$$

where g is the gravitational acceleration. $n_x \in \mathbb{R}$ and $n_z \in \mathbb{R}$ represent tangential overload and normal overload, and $\mu \in [-\pi, \pi]$ represents the roll angle around v. $[n_x, n_z, \mu] \in \mathbb{R}^3$ are the feasible basic control parameters in the UAV maneuver control model and they

jointly control the direction and magnitude of UAV speed. $[n_x, n_z, \mu]$ are often used as the command for air combat maneuver decision making.

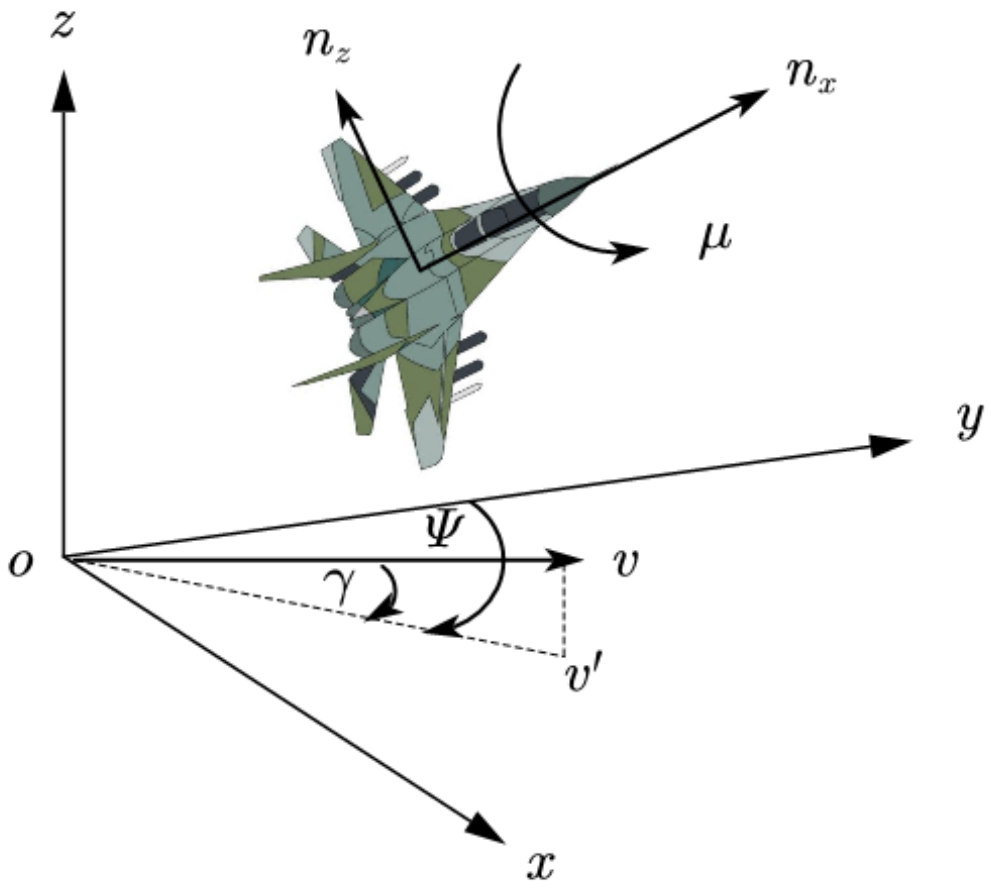

Figure 1. Motion model of UAV.

2.2. Situation Assessment Model

As shown in Figure 2, denote by P_{U_t} and P_{T_t} the position of UAV and target at time t. Denote by φ_{U_t} the angle between the vectors $P_{T_t} - P_{U_t}$ and v_{U_t}, named as the lag angle, and similarly, φ_{T_t} for the angle between the vectors $P_{T_t} - P_{U_t}$ and v_{T_t}, named as lead angle, which are defined as

$$\varphi_{U_t} = \arccos\left(\frac{v_{U_t}(P_{T_t} - P_{U_t})}{\|v_{U_t}\|\|P_{T_t} - P_{U_t}\|}\right), 0 \le \varphi_{U_t} \le \pi, \tag{3}$$

$$\varphi_{T_t} = \arccos\left(\frac{v_{T_t}(P_{T_t} - P_{U_t})}{\|v_{T_t}\|\|P_{T_t} - P_{U_t}\|}\right), 0 \le \varphi_{T_t} \le \pi, \tag{4}$$

where $D_{U_tT_t} = \|P_{T_t} - P_{U_t}\|$ is the distance between UAV and target.

Based on the attack model [36] and evaluation function [31,37], the effective attack range of a UAV in the air combat is a cone with an axis in the direction of v_{U_t} and angle of φ_m, which is truncated by a ball of radius $D_{\max}$ as shown in Figure 2, where $D_{\max}$ represents the attack range of weapons. Similarly, we can define the cone-shape attack range for the target. The UAV should try to follow the target as much as possible. That is, the smaller the φ_{U_t} is, the greater the probability of UAV successfully attacking the target is. On the contrary, the larger the φ_{T_t} is, the greater the probability of the target successfully attacking UAV is. Therefore, we define

$$\eta_{A_t} = \varphi_{U_t} + \varphi_{T_t} \tag{5}$$

to reflect the changes of the angle situation between the target and the UAV in the process of air combat confrontation, and there is $0 \leq \eta_{A_t} \leq 2\pi$. The smaller the η_{A_t} is, the more means that the UAV is in pursuit posture confrontation the target. In addition to angle, distance is also an important factor in air combat. Denote by $D_{\min}$ the minimum distance that a UAV can reach to the target for safety. When $D_{\mathrm{U}_t\mathrm{T}_t} < D_{\min}$, it means that the target is in the blind zone of UAV radar detection, and the UAV has the risk of collision [38]. Thus, the distance situation of UAV in air combat can be defined as

$$\eta_{D_t} = \frac{D_{\max} - D_{\mathrm{U}_t\mathrm{T}_t}}{D_{\max} - D_{\min}}, \tag{6}$$

where $D_{\mathrm{U}_t\mathrm{T}_t}$ is distance which can be as long as 50km in the air combat. The larger the η_{D_t} is, the closer the distance between the UAV and the enemy is. Outside the $D_{\min}$, the smaller the $D_{\mathrm{U}_t\mathrm{T}_t}$ and lag angle are, the greater the probability of success of UAV attack on the target is. Combining angle situation and distance situation, we define

$$\eta_t = w_1\eta_{A_t} + w_2\eta_{D_t}, \tag{7}$$

to evaluate the real-time comprehensive situation during UAV air combat, where w_1 and w_2 are scale factors, which represent the influence of different situation factors on UAV situation assessment. Moreover, since the value range of η_{A_t} and η_{D_t} is very different, we have to balance the effects of η_{A_t} and η_{D_t} by using w_1 and w_2.

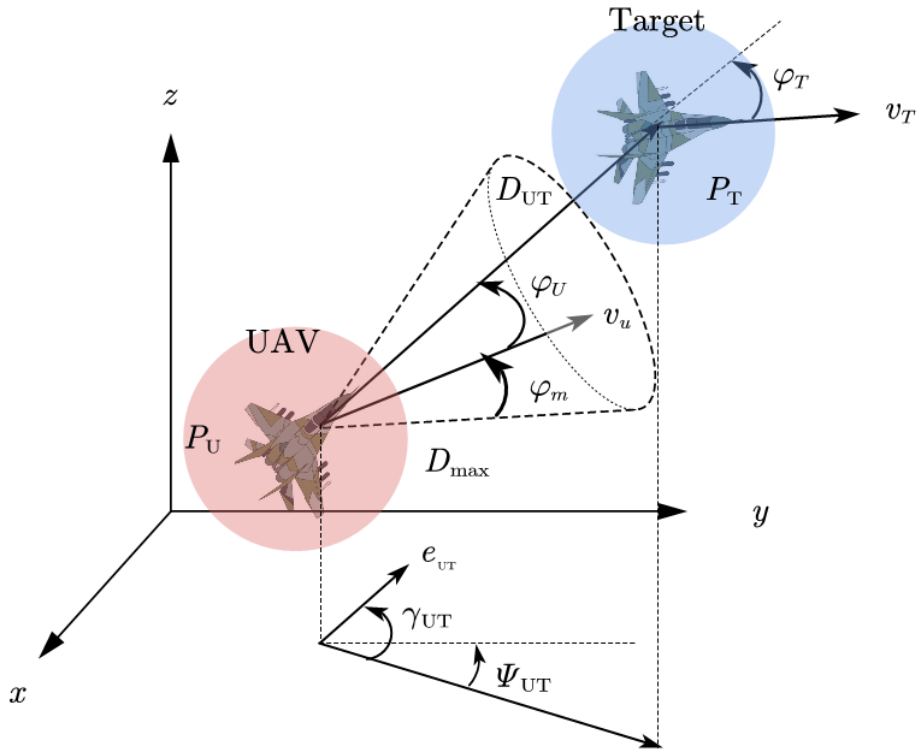

Figure 2. Situation assessment model.

Therefore, the air combat maneuver decision making problem of UAV can be regarded as an optimization problem,

$$\max_{[n_{x_t}, n_{z_t}, \mu_t] \in \Lambda} \sum_{t=t_0}^{t_n} \eta_t(n_{x_t}, n_{z_t}, \mu_t). \tag{8}$$

where Λ denotes a set of UAV maneuver control commands. $\eta_t(n_{x_t}, n_{z_t}, \mu_t)$ means η_t is the function of n_{x_t}, n_{z_t}, and μ_t, where $[n_{x_t}, n_{z_t}, \mu_t]$ has the same meaning as (2). Thus (8)

means maximize the sum of UAV situation from the beginning to the end of air combat. It is difficult to get the optimal solution because the objective function is complex high-order nonlinear. Next, we will use deep reinforcement learning to solve this problem.

3. Maneuver Decision Modeling by Deep Q Network

3.1. Reinforcement Learning

Reinforcement learning is a method for the agent to optimize maneuver strategy. The air combat maneuver decision making problem discussed in this paper belongs to the model-free reinforcement learning problem. Markov decision process (MDP) is usually used as the theoretical framework of model-free reinforcement learning, and the final objective of the reinforcement learning is to solve the MDP by deducing an optimum policy [39], which is described by quaternion array $[S, A, R, \gamma]$, where S represents state space, A represents action space, R represents reward function and γ represents discount factor. Reinforcement learning uses the state action value function Q to evaluate the value of action taken in the current state [18], which is defined as

$$\begin{aligned} Q_\pi(s,a) &= E_\pi[G_t|s_t = s, a_t = a] \\ &= E_\pi\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1}|s_t = s, a_t = a\right], \end{aligned} \tag{9}$$

where $s \in S$, $a \in A$, $r \in R$. In addition, in order to facilitate the calculation, the following simplified processing is usually implemented [40],

$$Q_\pi(s,a) = E_\pi[r_{t+1} + \gamma Q(s_{t+1}, a_{t+1})|s_t = s, a_t = a]. \tag{10}$$

Reinforcement learning finds the optimal strategy $\pi_*(s)$ by finding the optimal action value function $Q_*(a|s)$, i.e.,

$$Q_*(s,a) = \max_\pi Q_\pi(s,a). \tag{11}$$

As long as the maximum action value function is found, the corresponding strategy π_* is the solution of the reinforcement learning problem. In order to solve the dimension disaster problem of (11), deep reinforcement learning algorithm [41] is proposed, and transforms table value learning into parameters fitting of the neural network, i.e.,

$$Q_\pi(s,a) = Q_\pi(s,a,\theta), \tag{12}$$

where θ is the parameters of neural network and $Q(s,a;\theta)$ is called online Q network. Therefore, the solution of reinforcement learning problem can be expressed as

$$\pi_*(s) = arg\max_\pi Q_\pi(s,a,\theta). \tag{13}$$

3.2. State Space

The state of UAV can quantitatively reflect the current air combat information. We consider designing the state space of UAV from the following three aspects: The first is the maneuver characteristics of UAV. The second is the relative situation between the UAV and the target. The third is target dynamic embedding prediction of the situation in real-time combat.

In this paper, we use the following 13 variables to form the state space, v_U, γ_T, γ_U, φ_U, φ_T, $v_U - v_T$, ψ_T, ψ_U, D_{UT}, z_U, $z_U - z_T$, ψ_{UT}, γ_{UT}. γ_{UT} represents the angle between $P_{T_t} - P_{U_t}$ and the oxy plane, and ψ_{UT} represents the angle between projection vector of $P_{T_t} - P_{U_t}$ on the oxy plane and ox axis as shown in Figure 2. In order to unify the range of each state variable and improve the efficiency of network learning, each state variable is normalized to a range, as shown in Table 1.

Table 1. The state space for the DQN model.

State	Definition	State	Definition	State	Definition
s_1	$\frac{v_U}{v_{max}}$	s_2	$\frac{\gamma_U}{2\pi}$	s_3	$\frac{\varphi_U}{2\pi}$
s_4	$\frac{v_U - v_T}{v_{max} - v_{min}}$	s_5	$\frac{\gamma_T}{2\pi}$	s_6	$\frac{\varphi_T}{2\pi}$
s_7	$\frac{D_{UT}}{D_{thres}}$	s_8	$\frac{\gamma_{UT}}{2\pi}$	s_9	$\frac{v_U}{v_{max}}$
s_{10}	$\frac{\psi_U}{2\pi}$	s_{11}	$\frac{\psi_U}{2\pi}$	s_{12}	$\frac{z_U}{z_{max}}$
s_{13}	$\frac{z_U - z_T}{z_{max} - z_{min}}$				

v_{max} and v_{min} represent the maximum and minimum speed of UAV respectively. z_{max} and z_{min} represent the maximum and minimum safe altitude of UAV flight respectively. *Dthres* is the distance threshold, and represents the starting distance of close combat. Therefore, the state space can be defined as $s = [s_1, s_2, ..., s_{13}]$.

3.3. Action Space

As mentioned above, $[n_{x_t}, n_{z_t}, \mu_t] \in \Lambda$ constitute the action space of UAV. Different deep reinforcement learning algorithms such as DDPG and DQN have different design methods for the action space. Due to the huge state space, using DDPG and other algorithms to train continuous maneuvering strategies will cause difficulty in neural network convergence, and the maneuver process of UAV can be regarded as a combination of some basic actions [42], thus this paper adds eight action commands on the basis of the basic air combat maneuver [43] divided by NASA, and finally discretizes the UAV action space into 15 actions, as shown in Figure 3 and Table 2. This approach reduces the difficulty of UAV training, and compared with the basic seven actions, this method can make the UAV carry out constant speed, acceleration and deceleration control in each direction, which is closer to the real flight mode of the UAV.

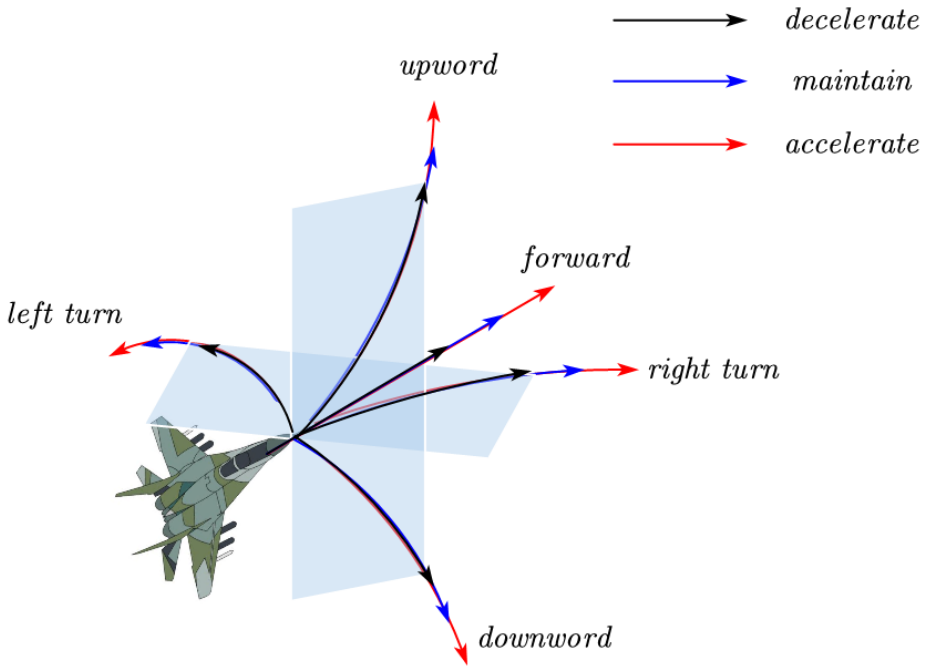

Figure 3. UAV maneuver library.

The UAV selects an action $a \in A$ according to the state s and outputs it to the environment. After format conversion, the UAV is guided to fly according to the command,

$$A = \{a_1, a_2, ..., a_m\} \subseteq \Lambda, m = 15, \tag{14}$$

$$a_i = [n_x, n_z, \mu], i = 1, 2, ..., 15. \tag{15}$$

Table 2. Maneuver library.

No.	Maneuver	Control Values		
		n_x	n_z	μ
a_1	forward maintain	0	1	0
a_2	forward accelerate	2	1	0
a_3	forward decelerate	–1	1	0
a_4	left turn maintain	0	8	$-\arccos(1/8)$
a_5	left turn accelerate	2	8	$-\arccos(1/8)$
a_6	left turn decelerate	–1	8	$-\arccos(1/8)$
a_7	right turn maintain	0	8	$\arccos(1/8)$
a_8	right turn accelerate	2	8	$\arccos(1/8)$
a_9	right turn decelerate	–1	8	$\arccos(1/8)$
a_{10}	upward maintain	0	8	0
a_{11}	upward accelerate	2	8	0
a_{12}	upward decelerate	–1	8	0
a_{13}	downward maintain	0	8	π
a_{14}	downward accelerate	2	8	π
a_{15}	downward decelerate	–1	8	π

3.4. Reward Function

Reward function [44–47] is the feedback signal obtained by the agent in the process of interaction with the environment, which is used to evaluate the effect of the agent executing a certain action strategy. Therefore, reasonable design of reward function can effectively improve the convergence speed of the system [45]. The reward r_t in this paper consists of two parts, which is defined as follow.

$$r_t = R_t + R_{\eta_t}. \tag{16}$$

In (16), R_t is the evaluation of the final result of air combat, which defines as (17).

$$R_t = \begin{cases} C, & \text{if UAV wins,} \\ -C, & \text{if Target wins,} \\ 0, & \text{otherwise.} \end{cases} \tag{17}$$

C is a constant and greater than 0. When the UAV meets the following conditions, it is considered that "UAV wins" is established and a success reward C is given. First, an optimal shooting distance threshold D_{attack} based on expert experience is defined, and there is $D_{attack} \leq D_{\max}$. Then if the distance between UAV and the target meets $D_{\min} < D_{U_tT_t} < D_{attack}$ and the angle meets $\varphi_{U_t} \leq \varphi_m$ and $\varphi_{T_t} \geq \pi - \varphi_m$ at the same time, it is considered that the UAV has the best shooting conditions against the target, and the "UAV wins" condition is established. Similarly, if the distance between UAV and the target satisfies $D_{\min} < D_{U_tT_t} < D_{attack}$ and the angle satisfies $\varphi_{T_t} \leq \varphi_m$ and $\varphi_{U_t} \geq \pi - \varphi_m$ at the same time, the condition "Target wins" is established and we give the failure penalty $-C$.

R_{η_t} in (16) is used for the real-time evaluation of maneuver decision making, which is defined as

$$R_{\eta_t} = \eta_t - \eta_{t-1} = w_1(\eta_{A_t} - \eta_{A_{t-1}}) + w_2(\eta_{D_t} - \eta_{D_{t-1}}). \tag{18}$$

(18) indicates the change of situation of UAV during air combat. If $R_{\eta_t} > 0$, the situation at time t is better than that at time $t-1$. The maneuver strategy a_{t-1} adopted by UAV from s_{t-1} to s_t is reasonable, and we give a positive reward. On the contrary, if $R_{\eta_t} < 0$, we give a negative penalty. w_1 and w_2 indirectly affect the maneuver decision making of UAV by influencing r_t. Considering the different importance of η_{A_t} and η_{D_t} under different s_t, an evaluation method of maneuver decision making based on w_1 piecewise adjustment is proposed. In this paper, w_1 is set as a piecewise function,

$$w1 = \begin{cases} 0, & \text{if } D_{\mathrm{U}_t\mathrm{T}_t} > D_{\max}, \\ W_1 & \text{if } D_{attack} \le D_{\mathrm{U}_t\mathrm{T}_t} \le D_{\max}, \\ W_2 & \text{otherwise.} \end{cases} \tag{19}$$

where W_1 and W_2 are constants, and $0 < W_1 < W_2$. When $D_{\mathrm{U}_t\mathrm{T}_t} > D_{\max}$ holds, UAV should give priority to adjust $D_{\mathrm{U}_t\mathrm{T}_t}$ to quickly approach the target. When $D_{\mathrm{U}_t\mathrm{T}_t} \le D_{\max}$ holds, UAV should consider adjusting φ_{U_t} and $D_{\mathrm{U}_t\mathrm{T}_t}$ at the same time, and the smaller the $D_{\mathrm{U}_t\mathrm{T}_t}$ is, the larger the w_1 is. w_2 is a constant and there is no need to set a piecewise function for w_2 because the relative importance of w_2 will change with w_1.

Remark 1. *If we change one of w_1 and w_2, the influence of angle advantage and distance advantage on UAV maneuver decision will change relatively. Therefore, this paper discusses the design of w_1. The setting method of w_1 can consider the following three types: Fixed value, piecewise function as (17), or continuous function as $w_1 \propto D_{\mathrm{U}_t\mathrm{T}_t}$. The fixed value represents that the influence of η_{A_t} and η_{D_t} on UAV maneuver decision is fixed. The piecewise function represents the influence of η_{A_t} and η_{D_t} on UAV maneuver decision, which is changed in a limited number of different cases. The continuous function represents the influence of η_{A_t} and η_{D_t} on UAV maneuver decision, which changes in real-time according to the current state of UAV. In this paper, we set w_1 as a piecewise function with the following considerations. r_t is a comprehensive evaluation of UAV at time t on η_{A_t}, η_{D_t} and R_t and these three contents are independent of each other. If $w_1 \propto D_{\mathrm{U}_t\mathrm{T}_t}$, $D_{\mathrm{U}_t\mathrm{T}_t}$ and φ_{U_t} are coupled with each other. Then with the change of $D_{\mathrm{U}_t\mathrm{T}_t}$, the change of w_1 represents the synchronous change of the importance of η_{A_t} and η_{D_t} to UAV maneuver decision making, which cannot reflect the different importance of the two advantage functions in different stages of air combat. In addition to the above theoretical analysis, this paper also gives the comparison results in the experimental stage. That is, w_1 is set as a fixed value, piecewise function and continuous function proportional to distance respectively, and analyzes the changes of the loss function in three cases, and further explains the rationality of w_1 as a piecewise function.*

3.5. Priority Sampling and Network Training

In this paper, the DQN algorithm is used to realize the self-learning of UAV maneuver control strategy [41,48], and an improved priority sampling strategy is proposed to accelerate the training and learning process. Experience replay mainly includes two key steps of "experience storage" and "sampling replay". Here, it is mainly to improve "sampling replay". The basic idea is to assign a priority to each sample in the experience pool. When selecting experience, we prefer to choose the experience with high priority. First, the data in the experience pool is marked according to importance, that is, the greater the value of data to network training is, the more important it is, and the higher the corresponding priority is. Then sample the labeled data, that is, the higher the priority is, the greater the probability that the sample is extracted is. Finally, the extracted samples are used for the weighted training of the network. Next, we will introduce priority sampling in detail from three aspects: Sample labeling, sampling and network training.

p_i is used to indicate the importance of the ith sample (s_i, a_i, r_i, s_{i+1}). A reasonable approach is to use the following TD error δ_i to assign p_i [49],

$$y_i = r_i + \gamma \max_{a'} Q(s_{i+1}, a'; \theta'), \tag{20}$$

$$\delta_i = y_i - Q(s_i, a_i; \theta). \tag{21}$$

where y_i is called the target Q value, and δ_i represents TD error. Since $|\delta_i| \geq 0$, and the larger the $|\delta_i|$ is, the more important the sample i is [49], and the higher the probability of being sampled should be. In order to avoid accidental factors that cause $|\delta_i|$ of some samples to be too large and the sampling probability of some samples with lower priority to be close to 0 resulting in the decrease of sampling diversity, $|\delta_i|$ is limited in $[0-1]$ by using tanh function, i.e.,

$$p_i = \tanh(|\delta_i| + \sigma), \tag{22}$$

where σ is a positive number, so that $p_i = \sigma > 0$ at $\delta_i = 0$. Then the sampling probability of sample i is expressed as

$$P(i) = \frac{p_i^\alpha}{\sum_k p_k^\alpha}, \tag{23}$$

where α is the priority factor, which is used to adjust the priority of the sample. The larger the α is, the larger the $P(i)$ will be. When $\alpha = 0$, the above equation will degenerate into uniform sampling, and k is the number of samples.

Remark 2. *The definition of $P(i)$ is not unique. Two variants are proposed in the [49]. The second variant is $p_i = \frac{1}{rank(i)}$. $rank(i)$ represents the rank of sample i sorted according to $|\delta_i|$. Considering the simplicity of code implementation, we use $p_i = |\delta_i| + \sigma$.*

Prioritized replay introduces bias because it changes distribution of sampled data in an uncontrolled fashion, and therefore changes the solution that the $Q(s, a; \theta)$ will converge to. Generally, the important sampling (IS) weights λ_i can be used to correct this error,

$$\lambda_i = \left(\frac{1}{N} \cdot \frac{1}{P(i)}\right)^\beta, \tag{24}$$

where N represents the experience pool capacity and β represents the compensation degree. If $\beta = 1$, the non-uniform probability is fully compensated. Then update the network parameters by using $\lambda_i \delta_i$ instead of δ_i. For stability reasons [50], we will standardize λ_i as follow,

$$\lambda_i = \frac{(N \cdot P(i))^{-\beta}}{\max_j \lambda_j}. \tag{25}$$

After obtaining the sample data needed for network training through priority sampling, we input it into $Q(s, a, \theta')$ and $Q(s, a, \theta)$ to update θ. DQN adjusts θ through gradient descent method during training, and the loss function after adding importance sampling weight λ_i is

$$L(\theta) = \delta_i \lambda_i, \tag{26}$$

and its gradient is

$$\frac{\partial L(\theta)}{\partial \theta} = \lambda_i \frac{\partial Q(s_i, a_i; \theta)}{\partial \theta}. \tag{27}$$

Finally, in order to collect enough samples for network training, DQN algorithm uses $\varepsilon - greedy$ strategy [41,48] to select actions, i,e.,

$$a_t = \pi(s) = \begin{cases} \arg\max_{a \in A} Q(s_t, a; \theta), & \text{if num} > \varepsilon; \\ \text{random} A, & \text{otherwise}, \end{cases} \tag{28}$$

where num is a random number of 0–1.

The above UAV BVR autonomous maneuver decision algorithm is summarized in the form of pseudo code as shown in Algorithm 1.

Algorithm 1 DQN with proportional prioritization

Initialize online network Q with random parameters θ Initialize target network Q' with random parameters θ' Initialize replay buffer M, Initialize hyper-parameters $D_{\max}$, $D_{\min}$, $V_{\max}$, γ, φ_{m}, w_2, W_1, W_2, β, $z_{\min}$, $z_{\max}$, σ, D_{BVR}, $V_{\min}$, D_{WVR}, α, a, b, C, k, K, D_{attack}.
for $episode = 1$ to N **do**
  Initialize the initial state of air combat
  Receive initial observation state s_1
  for $t = 1$ to T **do**
    With probability ε select a random action a_t
    Otherwise select $a_t = \max_a Q(s_t, a; \theta)$
    UAV executes action a_t, and target executes action according to its policy
    Receive reward r_t and observe new state s_{t+1}
    Store transition (s_t, a_t, r_t, s_{t+1}) in M
    Sample a mini batch of N transition $(s_{t+1}, a_t, r_t, s_{t+1})$ from M with priority $P(i) = p_i^\alpha / \sum_k p_k^\alpha$
    Compute importance-sampling weight $\lambda_i = (N \cdot P(i))^{-\beta} / \max_j \lambda_j$
    Set $y_i = r_i + \gamma \max_{a'} Q(s_{i+1}, a'; \theta')$
    Compute TD-error $\delta_i = y_i - Q(s_i, a_i; \theta)$
    Update transition priority $p_i \leftarrow |\delta_i|$
    Perform a gradient descent step on $\lambda_i (y_i - Q(s_i, a_i; \theta))^2$ with respect to the network parameters θ
    Every K steps reset $\theta' = \theta$
  end for
end for

The current state of the UAV is s_t. The online Q network selects and executes the action a_t based on $\varepsilon - greedy$ to transfer the UAV to the next state s_{t+1}, and obtain the reward r_t. Save (s_t, a_t, r_t, s_{t+1}) to the experience pool, and repeat the above steps until the number of samples in the experience pool meets the requirements. Select samples from the experience pool according to the priority $P(i)$ to train the neural network, and calculate the importance sampling weight λ_i of the selected samples. Use these samples to train the network parameters. That is, first, input s_t into the online Q network, and input s_{t+1} into the target Q network. Second, calculate the weighted mean square error according to (25), and use (27) to update the online Q network's parameters. At the same time, TD error $\delta_i = y_i - Q(s_i, a_i; \theta')$ is obtained. According to δ_i, the priority of the selected samples in the experience pool is updated, and the target network parameters are updated after a certain number of times or rounds of training. In the training process, the ε should be increased slowly, so that the UAV can choose the optimal action according to the value function with greater probability. When the error is close to 0 or there is no obvious change, the training is stopped, and the trained neural network is saved to obtain the air combat maneuver strategy of dual-aircraft formation

$$\pi(a|s)_{\mathrm{U}} = \arg\max_{a \in A} Q(s, a, \theta). \tag{29}$$

4. Olive Formation Air Combat as an Example

4.1. Task Description

This paper takes a typical two UAVs olive formation combat scene as an example. As shown in Figure 4, two UAVs perform tracking, defense and attacking tasks.

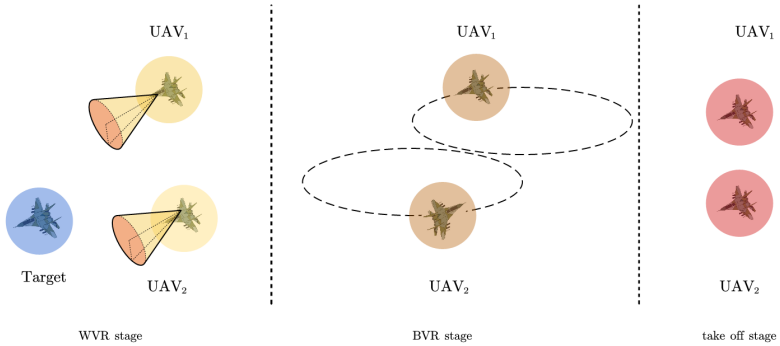

Figure 4. Task description.

UAV air combat can be divided into three stages: Take-off stage, BVR tracking stage and WVR attack stage. During the take-off stage, the two UAVs took the enemy aircraft as the target and continuously accelerated to the target direction. In the BVR tracking stage, the nose of one UAV faces the target and the nose of the other UAV backs the target. Two UAVs fly in olive formation to maintain the continuity of attack and defense. In this stage, the trajectory formed by the UAV from flying towards the target to flying back to the target and then flying towards the target is a circle or ellipse, so it is called olive formation. The process of WVR combat is also called dog fight. The two UAVs find the best angle and distance to attack the enemy and avoid entering the attack range of the target at the same time. We assume that the UAVs can accurately obtain any information they want, and then control the speed, yaw and roll through autonomous maneuver decision making, so as to track, defend and attack the target.

There are three problems to be considered for dual-aircraft autonomous maneuver control decisions: First, how to conduct inter aircraft collision avoidance? Second, how to make the two UAVs form an olive formation to maintain the continuity of attack and defense? Third, how to make the two UAVs maneuver independently to realize BVR tracking and short range attack? Next, we solve the above three problems by designing a hybrid autonomous maneuver strategy of obstacle avoidance, formation and confrontation.

4.2. Collision Avoidance and Formation Strategy

As described in Section 4.1, $\mathfrak{A} = \{\mathrm{U}_1, \mathrm{U}_2\}$ is used to represent our UAV set. We use U to represent any UAV in $\mathfrak{A}$, and use $\tilde{\mathrm{U}}$ to represent U's friendly aircraft. Denote by $D_{\mathrm{U}_t\tilde{\mathrm{U}}_t}$ the distance between U and $\tilde{\mathrm{U}}$ at time t. First, if $D_{\mathrm{U}_t\tilde{\mathrm{U}}_t} < D_{\min}$, no matter whether the UAV meets the firing conditions or not, it must avoid collision between UAVs. When $D_{\mathrm{U}_t\tilde{\mathrm{U}}_t} < D_{\min}$, the fast and effective obstacle avoidance method between UAVs is to change the flight altitude between itself and friendly aircraft. Therefore, the collision avoidance strategy is as follows,

$$O(a|s)_{\mathrm{U}} = \begin{cases} a_{11}, & \text{if} \quad D_{\mathrm{U}_t\tilde{\mathrm{U}}_t} < D_{\min}, \quad \mathrm{U}^z \geq \tilde{\mathrm{U}}^z, \\ a_{14}, & \text{if} \quad D_{\mathrm{U}_t\tilde{\mathrm{U}}_t} < D_{\min}, \quad \mathrm{U}^z < \tilde{\mathrm{U}}^z; \end{cases} \tag{30}$$

U^z and $\tilde{\mathrm{U}}^z$ represent the heights of U and $\tilde{\mathrm{U}}$ respectively. The UAV with a higher altitude adopts the accelerated ascent strategy, and the UAV with a lower altitude adopts the accelerated descent strategy. Second, in order to ensure that the UAVs can realize olive formation flight, and considering the flight characteristics and ease of control of the UAV, we use continuous uniform left turn to realize circling flight,

$$H(a|s)_{\mathrm{U}} = a_4. \tag{31}$$

However, strategy (31) is not enough. We also need to let the two UAVs know when to start flying in olive formation, who starts first and how to switch. It is assumed that U_1 is the leader and U_2 is the wingman. In the BVR tracking stage, the leader first uses the

strategy (31). It can be seen from (2) that the decision making time required by the UAV for a circle is

$$T_{olive} = \frac{2\pi v \cos\gamma}{g n_z \sin\mu}. \tag{32}$$

Finally, the maneuver control in takeoff stage, BVR tracking stage and WVR attack stage is realized by $\pi(a|s)_{\mathrm{U}}$.

Therefore, the maneuver strategy of UAV at time t can be expressed by the following equation.

$$\Pi(a|s)_{\mathrm{U}} = \begin{cases} O(a|s)_{\mathrm{U}}, & \text{if } D_{\mathrm{U}_t\tilde{\mathrm{U}}_t} < D_{\min}; \\ H(a|s)_{\mathrm{U}}, & \text{if } D_{\mathrm{U}_t\tilde{\mathrm{U}}_t} \geq D_{\min}, \\ & D_{\mathrm{WVR}} \leq D_{\mathrm{U}_t\mathrm{T}_t} \leq D_{\mathrm{BVR}}, \\ & \Pi(a|s)_{\tilde{\mathrm{U}}} \neq H(a|s)_{\tilde{\mathrm{U}}}; \\ \pi(a|s)_{\mathrm{U}}, & \text{otherwise.} \end{cases} \tag{33}$$

D_{BVR} represents the distance threshold of BVR air combat, and D_{WVR} represents the distance threshold of WVR air combat. If $D_{\mathrm{U}_t\mathrm{T}_t} \geq D_{\mathrm{BVR}}$, our UAV belongs to take-off stage. If $D_{\mathrm{WVR}} \leq D_{\mathrm{U}_t\mathrm{T}_t} \leq D_{\mathrm{BVR}}$, our UAV belongs to BVR tracking stage. If $D_{\mathrm{U}_t\mathrm{T}_t} < D_{\mathrm{WVR}}$, our UAV belongs to the WVR air combat stage.

The autonomous maneuver decision algorithm of the dual-UAV olive formation is sketched in Algorithm 2.

Algorithm 2 Maneuver strategy of two UAVs olive formation in air combat

Load trained neural network $Q(s, a, \theta)$.
Initialize the state of the leader and wingman $(s_{\mathrm{U1},0}, s_{\mathrm{U2},0}, s_{\mathrm{T},0})$,
Initialize target maneuver strategy $\pi(a|s)_{\mathrm{T}}$.
for $step = 1$ to $maxstep$ **do**
    for U in $\mathfrak{A}$ **do**
        Calculate D_{UT}
        Calculate $D_{\mathrm{U}\tilde{\mathrm{U}}}$
        Execute $a_t = \Pi(a|s)_{\mathrm{U}}$
        $[\Delta v, \Delta\gamma, \Delta\psi]$ is obtained according to (2)
        $[\Delta x, \Delta y, \Delta z]$ is obtained according to (1)
        Get the next state $s_{\mathrm{U,t+1}}$
        $s_{\mathrm{U,t}} = s_{\mathrm{U,t+1}}$
    end for
    The target moves to the next state $s_{\mathrm{T},t+1}$ according to the strategy $\pi(a|s)_{\mathrm{T}}$
    if $D_{\mathrm{U}_t\mathrm{T}_t} < D_{attack}, \varphi_{\mathrm{U}_t} \leq \varphi_m, \varphi_{\mathrm{T}_t} \geq \pi - \varphi_m$ **then**
        UAVs win
        break
    end if
    if $D_{\mathrm{U}_t\mathrm{T}_t} < D_{attack}, \varphi_{\mathrm{T}_t} < \varphi_m, \varphi_{\mathrm{U}_t} > \pi - \varphi_m$ **then**
        target win
        break
    end if
    $s_{\mathrm{T,t}} = s_{\mathrm{T,t+1}}$
end for

First, the trained neural network $Q(s, a, \theta)$ and the maneuver strategy of the target $\pi(a|s)_{\mathrm{T}}$ are loaded. Initialize the state $(s_{\mathrm{U}_1,0}, s_{\mathrm{U}_2,0}, s_{\mathrm{T},0})$ of UAVs and target, where $s_{\mathrm{U}_1,0}$ and $s_{\mathrm{U}_2,0}$ are the initial state of our two UAVs respectively, and $s_{\mathrm{T},0}$ is the initial state of target. For each UAV, the distance D_{UT} and the distance $D_{\mathrm{U}\tilde{\mathrm{U}}}$ are calculated, and then the maneuver strategy a_t is obtained according to (33). If the distance between UAV and the enemy is less than $D_{\min}$, the collision avoidance strategy (30) is implemented. If the UAV is

in the takeoff stage, the UAV selects the maneuver strategy according to (29). If both UAVs enter the BVR tracking state, the leader first executes the strategy (31), and the number of execution steps is obtained according to (32), while the wingman continues to select a_t according to (29). When the leader completes a circle according to (32) and (31), the flight strategies of the leader and the wingman are exchanged. In the BVR tracking stage, our two UAVs constantly change their maneuver strategies to maintain the continuity of attack and defense. In the WVR attack stage, both UAVs use (29) to complete the short-range combat. If either of the enemy and our UAVs satisfies (17), the air combat ends.

5. Simulation

This paper uses Python language to establish the air combat environment model of dual-aircraft olive formation tracking and attacking, and establishes the DQN network model based on the PyTorch module.

5.1. Simulation Setup

The air combat environment parameters are set as shown in Table 3.

Table 3. Design of the simulation parameters.

Variable	Value	Variable	Value	Variable	Value
$D_{\max}$	1000	$D_{\min}$	200	D_{attack}	500
C	10	$V_{\max}$	300	b	10
γ	0.9	φ_{m}	45	w_2	20
β	0.4	$V_{\min}$	90	$z_{\min}$	1000
$z_{\max}$	12,000	$D_{threshold}$	10,000	a	5
W_1	30	W_2	40	α	0.6
k	5000	K	300	D_{BVR}	20,000
σ	0.01	D_{WVR}	10,000		

The parameters in the DQN model are set as follows. According to the definition of state space and maneuver library, it is obvious that DQN has 13 input states and 15 output Q values. The online Q network and the target Q network are constructed using a fully connected network. The network has three hidden layers, 512, 1024 and 512 units respectively. The output layer has no activation function, and the other layers are tanh layers. The learning rate is 0.001 and the discount coefficient is 0.9. The size of the experience pool is 5000, the number of samples taken in batch during training is 64, and the target network is updated every 300 steps. In the process of air combat simulation, the decision cycle T is set to 1s, and one set contains 100 decision steps. If any of the following conditions are met: The height of UAV is greater than $z_{\max}$ or less than $z_{\min}$, or any UAV meets (17). The round of training is completed.

In order to verify the effectiveness of the DQN algorithm based on priority sampling and dynamic adjustment of scale factor proposed in this paper, we compare the loss values in the training process of the DQN network under the following four conditions while ensuring the same initial conditions of the simulation: w_1 is set as a fixed value, piecewise function and continuous function proportional to distance respectively, and introduce the priority sampling. The loss value represents the difference between the online network and the target network. The larger the loss value is, the larger the network optimization space at this stage is. The greater the fluctuation of loss value is, the worse the convergence performance of the neural network is.

5.2. Simulation Results

Figure 5 shows the results of two UAVs formation air combat. Red and green are our two UAVs, and blue is the target. The initial positions of the three UAVs are fixed, namely (0,0,3000), (20,000,20,000,4000), (500,500,3000), and the heading angle and pitch angle are initialized randomly.

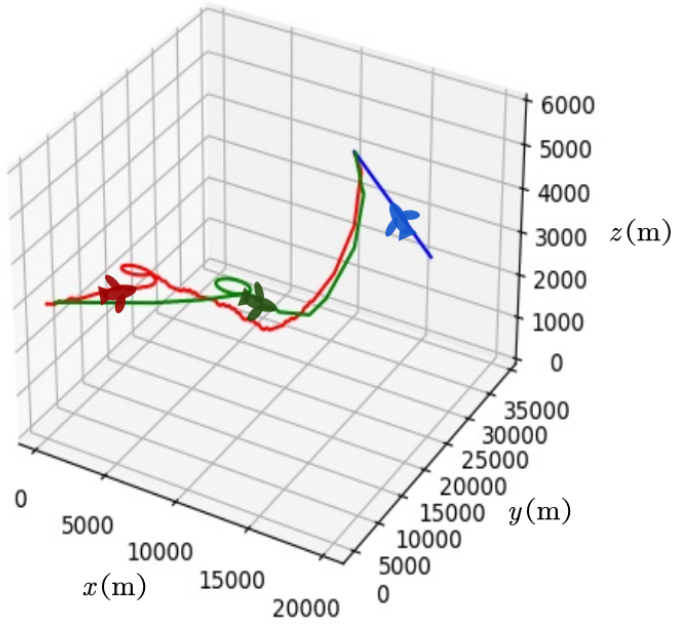

Figure 5. Combat result.

The target moved in a straight line at a constant speed. Our two UAVs take different maneuver control decisions in different air combat stages, and the two UAVs cooperate to complete the tracking, defense and attack tasks of the target. During the takeoff stage, our two UAVs tracked the target from a distance of about 30,000 m. In the BVR stage, the leader (red) first executes the strategy $H(a|s)_{\mathrm{U}}$, while the wingman (green) continues to track the target. When the leader hovers for one circle, its strategy is changed to track the target, and the wingman switches the flight strategy to $H(a|s)_{\mathrm{U}}$. The two UAVs cooperate to maintain the continuity of attack and defense. When the distance between UAV and target is less than D_{BVR}, two UAVs enter the close attack phase. Figure 5 shows that our two UAVs attack the target from the rear of the target. Figures 6–9 more clearly show the changes of various parameters of both sides in the process of air combat in Figure 5. The abscissas of Figures 5–9 are the decision time, and the ordinates are different air combat parameters.

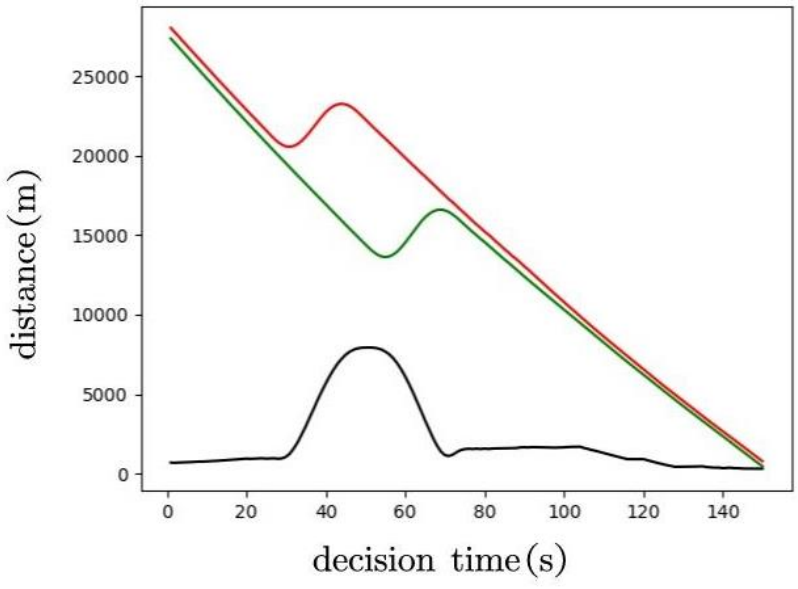

Figure 6. Change of the distance in the process of air combat.

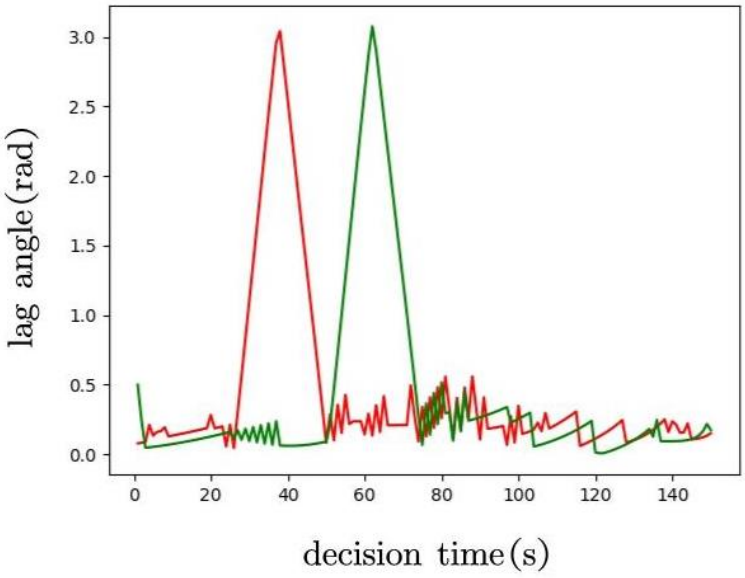

Figure 7. Change of the lag angle in the process of air combat.

Figure 6 shows the change of the distance between the two sides in the process of air combat, where red is the change curve of the distance between the leader and the target, and green is the change curve of the distance between the wingman and the target, and black is the change of the distance between our two UAVs. The ordinate is the distance in meters. It can be seen that the distance between the UAV and the target will increase when flying in circles, and the distance between the UAV and the target will continue to shorten in the stages of BVR tracking and WVR air combat. Figure 7 shows the change of the lag angle of the UAVs of our two UAVs, where red is the change of the lag angle between the leader UAV and the target, and green is the change of the lag angle between the wingman UAV and the target. The ordinate is the lag angle in radians. It can be seen that the lag angle of our UAV changes from 0 to π when flying in circles, and the lag angle remains at a low level during the pursuit and close combat.

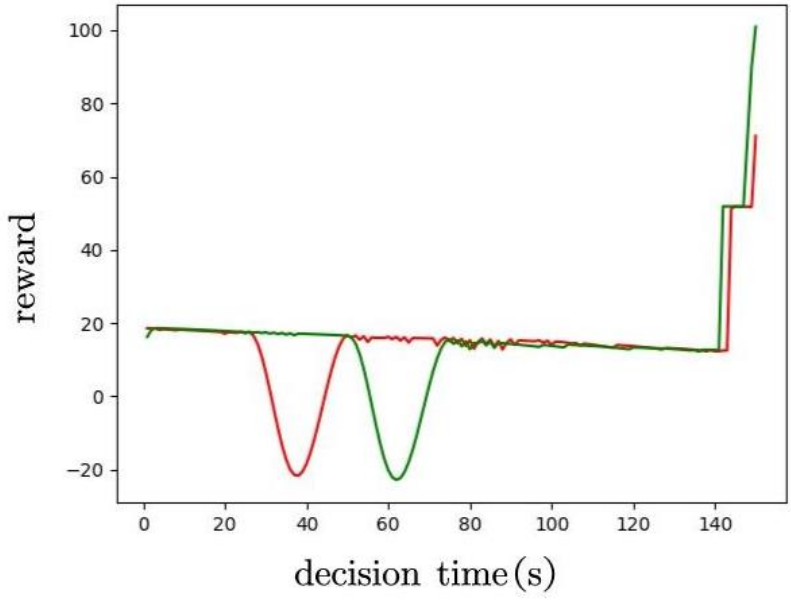

Figure 8. Change of the single step reward in the process of air combat.

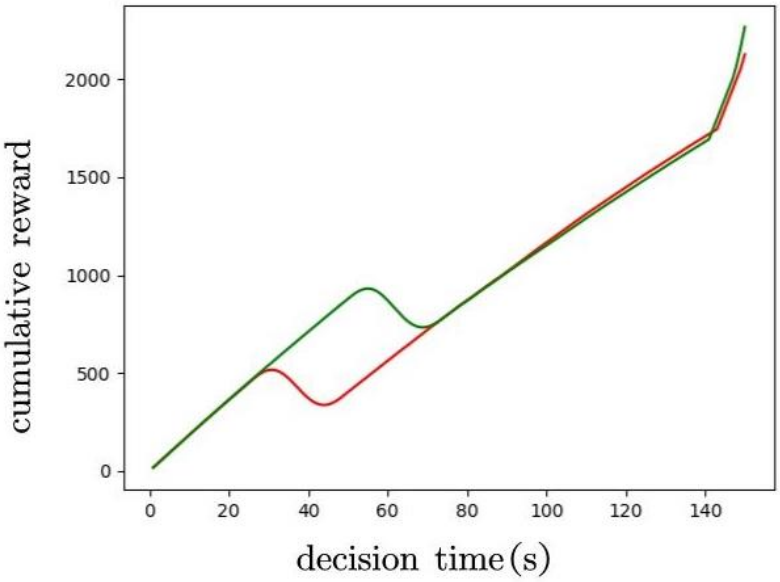

Figure 9. Change of the cumulative reward in the process of air combat.

Figure 8 shows the change of the reward function of our two UAVs in the whole process of air combat, in which red is the leader and green is the wingman. The abscissa is r_t. It can be seen that our UAV is flying in circles r_t fluctuated obviously, and the situation of pursuit and close combat remained positive. When the target enters the attack range of UAV, the angle advantage function is added, and our situation rises obviously. Figure 9 shows the change of cumulative r_t, which also shows an upward trend as a whole. Figure 10 shows the variation diagram of loss values during training when $w1$ are $w1 = -20$ and piecewise function (19), in which red is $w1 = -20$. Figure 11 shows the variation diagram of loss values during training when $w1$ are $w_1 \propto D_{\mathrm{U}_t\mathrm{T}_t}$ and piecewise function (19), in which red is $w_1 \propto D_{\mathrm{U}_t\mathrm{T}_t}$.

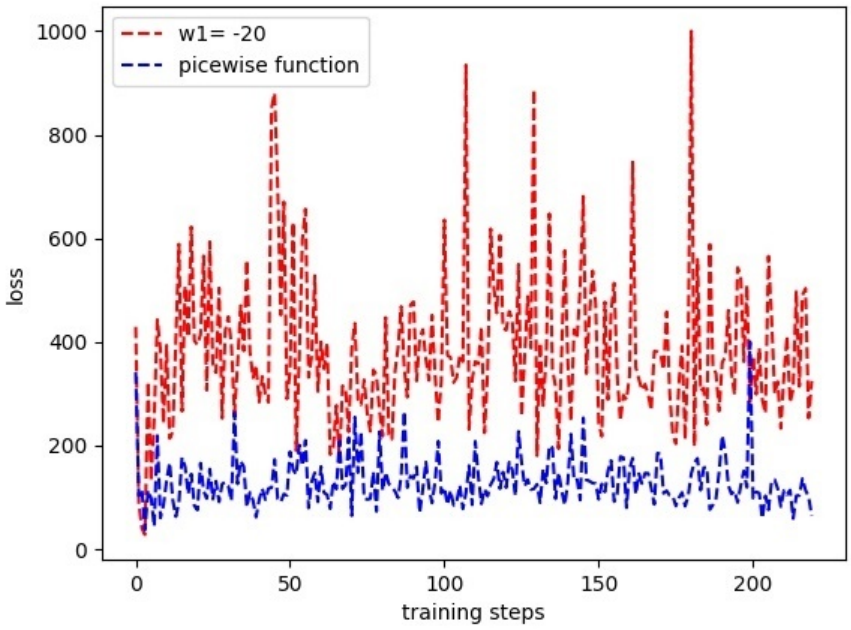

Figure 10. Comparison of loss with different forms of scale factors 1.

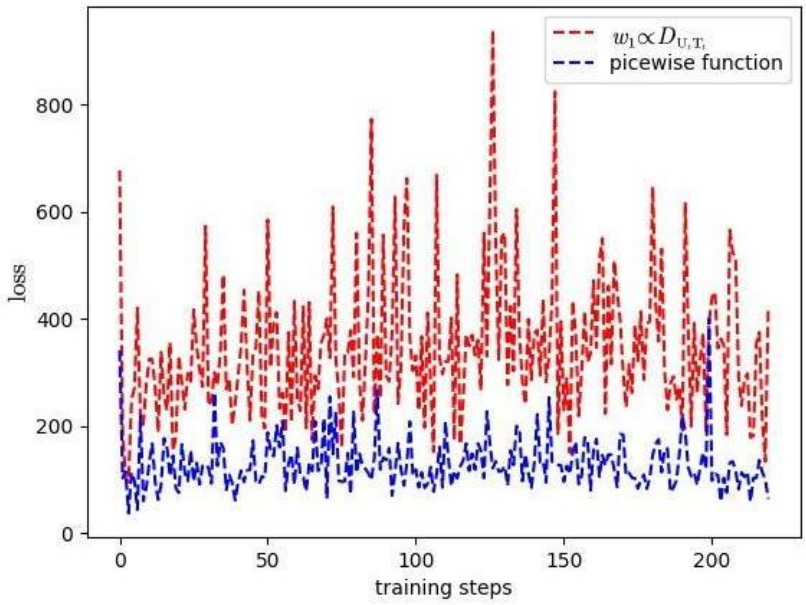

Figure 11. Comparison of loss with different forms of scale factors 2.

Figure 12 shows the error comparison diagram of introducing priority sampling and not introducing priority sampling on the premise that w_1 is a piecewise function (19). The blue curve represents the DQN algorithm considering priority sampling, and the red curve represents the DQN algorithm without priority sampling. The abscissas of the above three figures represent training steps, and the ordinates represent loss values, and the comparison shows the changing trend of loss in the whole training process. In addition, because the amount of data to be presented is too large, in order to avoid affecting the clarity of the display results, and on the premise of not affecting the loss change trend, Figures 10–12 choose to record the loss value every 300 training times. It is not difficult to see from the above figures that the fluctuation amplitude and range of the red curve are much larger than that of the blue curve. The network converges better when w_1 is a piecewise function. The priority sampling can effectively improve the learning efficiency of agents and accelerate the convergence speed of the neural network.

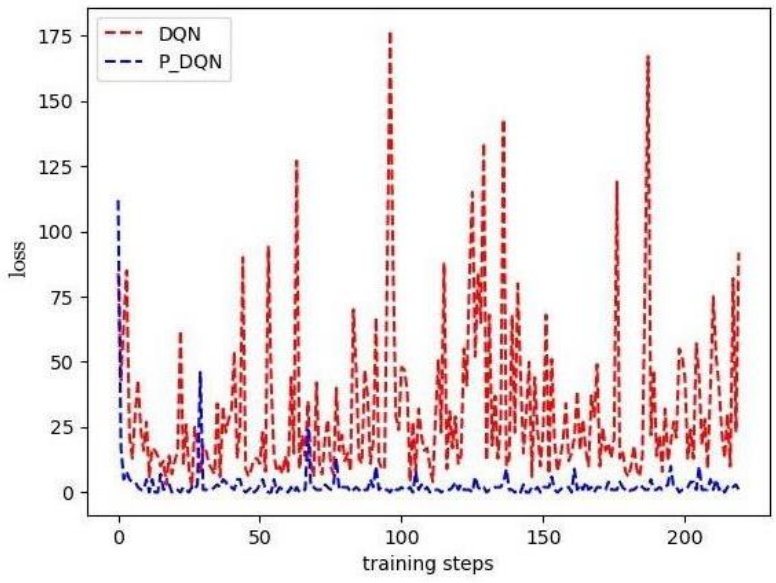

Figure 12. Comparison of convergence results before and after priority sampling.

In summary, by modeling and simulation, this paper solves the following problems that exist in the state of the art: (i) Dimension explosion solved by discretizing the air combat state space to be finite and using neural network to learn the decision making model; (ii) Sparse and delayed reward solved by designing a real-time reward function based on situation assessment; (iii) Slow convergence solved by using improved priority sampling strategy; and (iv) Inadaptation to the real air combat maneuver control solved by incorporating the real UAV dynamic model and the comprehensive situation assessment model which is verified in the classical two agents olive formation scenario.

6. Conclusions

Based on reinforcement learning theory, an improved maneuver decision algorithm for UAV autonomous air combat is proposed in this paper. First, the UAV dynamic model and situation assessment model are established, and the UAV state space and action space are improved to solve the dimension explosion problem and make the UAV maneuver more flexible. Second, aiming at the problems of delayed reward and poor guidance ability, a reward function design method based on adaptive adjustment of the relative situation and the scale factor is proposed. Third, an improved priority sampling strategy is proposed to speed up the learning rate. Fourth, based on the dual-UAV olive formation task, a hybrid maneuver strategy of collision avoidance, formation and confrontation is proposed to realize dual-UAV cooperative autonomous air combat decision making. The simulation results show that the improved method can effectively improve the efficiency of the UAV learning confrontation maneuver strategy, and the UAV air combat maneuver decision model based on deep reinforcement learning can realize strategy with self-learning. The improved deep reinforcement learning method has a faster training speed and a more stable effect.

Author Contributions: Conceptualization, J.H., L.W., C.G. and Y.W.; methodology, J.H., L.W. and T.H.; software, J.H., L.W. and T.H., validation, C.G. and Y.W.; formal analysis, J.H., L.W. and T.H.; writing, J.H., L.W. and T.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China (61803309), Key Research and Development Project of Shaanxi Province (2020ZDLGY06-02, 2021ZDLGY07-03), Aeronautical Science Foundation of China (2019ZA053008, 20185553034), CETC Key Laboratory of Data Link Technology Open Project Fund (CLDL − 20202101_2).

Conflicts of Interest: The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

DDPG	Deep Deterministic Policy Gradient
DQN	Deep Q Network
BVR	Beyond Visual Range
UAV	Unmanned Aerial Vehicle
MDP	Markov Decision Process
MARL	Multi-Agent Reinforcement Learning
IS	Important Sampling
TD error	Time Difference error
WVR	Within Visual Range

References

1. Ma, Y.; Wang, G.; Hu, X.; Luo, H.; Lei, X. Cooperative occupancy decision making of Multi-UAV in Beyond-Visual-Range air combat: A game theory approach. *IEEE Access* **2019**, *8*, 11624–11634. [CrossRef]
2. Azar, A.T.; Koubaa, A.; Ali Mohamed, N.; Ibrahim, H.A.; Ibrahim, Z.F.; Kazim, M.; Ammar, A.; Benjdira, B.; Khamis, A.M.; Hameed, I.A.; et al. Drone Deep Reinforcement Learning: A Review. *Electronics* **2021**, *10*, 999. [CrossRef]
3. Skorobogatov, G.; Barrado, C.; Salamí, E. Multiple UAV systems: A survey. *Unmanned Syst.* **2020**, *8*, 149–169. [CrossRef]

4. Fu, L.; Xie, F.; Meng, G.; Wang, D. An UAV air-combat decision expert system based on receding horizon control. *J. Beijing Univ. Aeronaut. Astronaut.* **2015**, *41*, 1994.
5. Zhang, Y.; Luo, D. Editorial of Special Issue on UAV Autonomous, Intelligent and Safe Control. *Guid. Navig. Control* **2021**, *1*, 2102001. [CrossRef]
6. Austin, F.; Carbone, G.; Falco, M.; Hinz, H.; Lewis, M. Automated maneuvering decisions for air-to-air combat. *AIAA J.* **1987**, *87*, 659–669.
7. Virtanen, K.; Raivio, T.; Hämäläinen, R.P. Decision Theoretical Approach to Pilot Simulation. *J. Aircr.* **1999**, *36*, 632. [CrossRef]
8. Pan, Q.; Zhou, D.; Huang, J.; Lv, X.; Yang, Z.; Zhang, K.; Li, X. Maneuver decision for cooperative close-range air combat based on state predicted influence diagram. In Proceedings of the 2017 IEEE International Conference on Information and Automation (ICIA), Macau, China, 18–20 July 2017; pp. 726–731.
9. An, X.; Yingxin, K.; Lei, Y.; Baowei, X.; Yue, L. Engagement maneuvering strategy of air combat based on fuzzy markov game theory. In Proceedings of the 2011 IEEE 2nd International Conference on Computing, Control and Industrial Engineering, Wuhan, China, 20–21 August 2011; Volume 2, pp. 126–129. [CrossRef]
10. Chae, H.J.; Choi, H.L. Tactics games for multiple UCAVs Within-Visual-Range air combat. In Proceedings of the AIAA Information Systems-AIAA Infotech@ Aerospace, Kissimmee, FL, USA, 8–12 January 2018; p. 0645.
11. Horie, K.; Conway, B.A. Optimal Fighter Pursuit-Evasion Maneuvers Found Via Two-Sided Optimization. *J. Guid. Control. Dyn.* **2006**, *29*, 105–112. [CrossRef]
12. Qiuni, L.; Rennong, Y.; Chao, F.; Zongcheng, L. Approach for air-to-air confrontment based on uncertain interval information conditions. *J. Syst. Eng. Electron.* **2019**, *30*, 100–109. [CrossRef]
13. Belkin, B.; Stengel, R. Systematic methods for knowledge acquisition and expert system development (for combat aircraft). *IEEE Aerosp. Electron. Syst. Mag.* **1991**, *6*, 3–11. [CrossRef]
14. Schvaneveldt, R.W.; Goldsmith, T.E.; Benson, A.E.; Waag, W.L. *Neural Network Models of Air Combat Maneuvering*; Technical Report; New Mexico State University: Las Cruces, NM, USA, 1992.
15. Huang, C.; Fei, J. UAV Path Planning Based on Particle Swarm Optimization with Global Best Path Competition. *Int. J. Pattern Recognit. Artif. Intell.* **2017**, *32*, 1859008. [CrossRef]
16. Wu, A.; Yang, R.; Liang, X.; Zhang, J.; Qi, D.; Wang, N. Visual Range Maneuver Decision of Unmanned Combat Aerial Vehicle Based on Fuzzy Reasoning. *Int. J. Fuzzy Syst.* **2021**, 1–18. [CrossRef]
17. Duan, H.; Li, P.; Yu, Y. A predator-prey particle swarm optimization approach to multiple UCAV air combat modeled by dynamic game theory. *IEEE/CAA J. Autom. Sin.* **2015**, *2*, 11–18.
18. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; MIT Press: Cambridge, MA, USA, 2018.
19. Gopi, S.P.; Magarini, M. Reinforcement Learning Aided UAV Base Station Location Optimization for Rate Maximization. *Electronics* **2021**, *10*, 2953. [CrossRef]
20. Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Huang, A.; Guez, A.; Hubert, T.; Baker, L.; Lai, M.; Bolton, A.; et al. Mastering the game of go without human knowledge. *Nature* **2017**, *550*, 354–359. [CrossRef]
21. Liu, P.; Ma, Y. A deep reinforcement learning based intelligent decision method for UCAV air combat. In Proceedings of the Asian Simulation Conference, Melaka, Malaysia, 27–29 August 2017; Springer: Berlin/Heidelberg, Germany, 2017; pp. 274–286.
22. You, S.; Diao, M.; Gao, L. Deep reinforcement learning for target searching in cognitive electronic warfare. *IEEE Access* **2019**, *7*, 37432–37447. [CrossRef]
23. Piao, H.; Sun, Z.; Meng, G.; Chen, H.; Qu, B.; Lang, K.; Sun, Y.; Yang, S.; Peng, X. Beyond-Visual-Range Air Combat Tactics Auto-Generation by Reinforcement Learning. In Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 19–24 July 2020; pp. 1–8.
24. Weiren, K.; Deyun, Z.; Zhang, K.; Zhen, Y. Air combat autonomous maneuver decision for one-on-one within visual range engagement base on robust multi-agent reinforcement learning. In Proceedings of the 2020 IEEE 16th International Conference on Control & Automation (ICCA), Sapporo, Hokkaido, 9–11 October 2020; pp. 506–512.
25. Zhang, Y.; Zu, W.; Gao, Y.; Chang, H. Research on autonomous maneuvering decision of UCAV based on deep reinforcement learning. In Proceedings of the 2018 Chinese Control and Decision Conference (CCDC), Shenyang, China, 9–11 June 2018; pp. 230–235.
26. Minglang, C.; Haiwen, D.; Zhenglei, W.; QingPeng, S. Maneuvering decision in short range air combat for unmanned combat aerial vehicles. In Proceedings of the 2018 Chinese Control and Decision Conference (CCDC), Shenyang, China, 9–11 June 2018; pp. 1783–1788.
27. 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]
28. You, S.; Diao, M.; Gao, L. Completing Explorer Games with a Deep Reinforcement Learning Framework Based on Behavior Angle Navigation. *Electronics* **2019**, *8*, 576. [CrossRef]
29. Wei, X.; Yang, L.; Cao, G.; Lu, T.; Wang, B. Recurrent MADDPG for Object Detection and Assignment in Combat Tasks. *IEEE Access* **2020**, *8*, 163334–163343. [CrossRef]

30. Kong, W.; Zhou, D.; Zhang, K.; Yang, Z.; Yang, W. Multi-UCAV Air Combat in Short-Range Maneuver Strategy Generation using Reinforcement Learning and Curriculum Learning. In Proceedings of the 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA), Miami, FL, USA, 14–17 December 2020; pp. 1174–1181.
31. Wang, L.; Hu, J.; Xu, Z.; Zhao, C. Autonomous maneuver strategy of swarm air combat based on DDPG. *Auton. Intell. Syst.* **2021**, *1*, 1–15. [CrossRef]
32. 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]
33. Hu, D.; Yang, R.; Zuo, J.; Zhang, Z.; Wu, J.; Wang, Y. Application of Deep Reinforcement Learning in Maneuver Planning of Beyond-Visual-Range Air Combat. *IEEE Access* **2021**, *9*, 32282–32297. [CrossRef]
34. Lee, G.T.; Kim, C.O. Autonomous Control of Combat Unmanned Aerial Vehicles to Evade Surface-to-Air Missiles Using Deep Reinforcement Learning. *IEEE Access* **2020**, *8*, 226724–226736. [CrossRef]
35. Fu, Q.; Fan, C.L.; Song, Y.; Guo, X.K. Alpha C2–An intelligent air defense commander independent of human decision-making. *IEEE Access* **2020**, *8*, 87504–87516. [CrossRef]
36. Zhang, S.; Duan, H. Multiple UCAVs target assignment via bloch quantum-behaved pigeon-inspired optimization. In Proceedings of the 2015 34th Chinese Control Conference (CCC), Hangzhou, China, 28–30 July 2015; pp. 6936–6941.
37. Chen, X.; Wei, X. Method of firepower allocation in multi-UCAV cooperative combat for multi-target attacking. In Proceedings of the 2012 Fifth International Symposium on Computational Intelligence and Design, Hangzhou, China, 28–29 October 2012; Volume 1, pp. 452–455.
38. Wang, G.; Li, Q.; He, L.; Yang, Z. Reacher on Calculation Model for Blind Zone of an Airborne Warning Rader. *Rader Sci. Technol.* **2010**, *8*, 8.
39. Vithayathil Varghese, N.; Mahmoud, Q.H. A survey of multi-task deep reinforcement learning. *Electronics* **2020**, *9*, 1363. [CrossRef]
40. Watkins, C.J.; Dayan, P. Q-learning. *Mach. Learn.* **1992**, *8*, 279–292. [CrossRef]
41. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [CrossRef] [PubMed]
42. Shaw, R.L. *Fighter Combat-Tactics and Maneuvering*; US Naval Institute Press: Annapolis, MD, USA, 1985.
43. Breitner, M.H.; Pesch, H.J.; Grimm, W. Complex differential games of pursuit-evasion type with state constraints, part 2: Numerical computation of optimal open-loop strategies. *J. Optim. Theory Appl.* **1993**, *78*, 443–463. [CrossRef]
44. Ng, A.Y.; Harada, D.; Russell, S. Policy invariance under reward transformations: Theory and application to reward shaping. In Proceedings of the Sixteenth International Conference on Machine Learning (ICML 1999), Bled, Slovenia, 27–30 June 1999; Volume 99, pp. 278–287.
45. Wiewiora, E. Potential-based shaping and Q-value initialization are equivalent. *J. Artif. Intell. Res.* **2003**, *19*, 205–208. [CrossRef]
46. Brys, T.; Harutyunyan, A.; Suay, H.B.; Chernova, S.; Taylor, M.E.; Nowé, A. Reinforcement learning from demonstration through shaping. In Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, Buenos Aires, Argentina, 25–31 July 2015.
47. Tan, C.Y.; Huang, S.; Tan, K.K.; Teo, R.S.H.; Liu, W.Q.; Lin, F. Collision avoidance design on unmanned aerial vehicle in 3D space. *Unmanned Syst.* **2018**, *6*, 277–295. [CrossRef]
48. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; Riedmiller, M. Playing atari with deep reinforcement learning. *arXiv* **2013**, arXiv:1312.5602.
49. Schaul, T.; Quan, J.; Antonoglou, I.; Silver, D. Prioritized experience replay. *arXiv* **2015**, arXiv:1511.05952.
50. Mahmood, A.R.; Van Hasselt, H.; Sutton, R.S. Weighted importance sampling for off-policy learning with linear function approximation. In Proceedings of the NIPS, Montreal, Canada, 8–13 December 2014; pp. 3014–3022.