IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 13, 2020, accepted April 1, 2020, date of publication April 3, 2020, date of current version April 22, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2985576

Deep Reinforcement Learning With Application to Air Confrontation Intelligent Decision-Making of Manned/Unmanned Aerial Vehicle Cooperative System

YUE LI[1], WEI HAN[1], AND YONGQING WANG[2]
[1]College of Basic Science for Aviation, Naval Aviation University, Yantai 264001, China
[2]Shenyang Aircraft Design and Research Institute, Shenyang 110035, China

Corresponding author: Yue Li (deutchland_li@foxmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61703414.

ABSTRACT With the development of intelligence in air confrontation, the demand for cooperative engagement of manned/unmanned aerial vehicle (MAV/UAV) is becoming more intense. Deep reinforcement learning (DRL), which combines the abstract representation capability of deep learning (DL) and the optimal decision-making and control capability of reinforcement learning (RL), is an appropriate application for dealing with this problem. In the case of continuous action space, the dynamics model of UAV and the basic structure of one of the most popular DRL methods called deep deterministic policy gradient (DDPG) are built firstly. To establish the framework of intelligent decision-making of MAV/UAV, typical intentions including Head-on attack, Fleeing, Pursuing and Energy-storing, corresponding to four optimization models, are introduced secondly. Then the neural network is trained by means of reconstructing the replay buffer of DDPG algorithm. Finally, simulation results show that UAV is able to learn intelligent decision-making throughout the intention guiding of MAV. Compared with original DDPG algorithm, the improved method can achieve a better performance in convergence and stability. Furthermore, the level of intelligent decision-making in air confrontation can be improved by self-learning.

INDEX TERMS Manned/unmanned aerial vehicle, intelligent decision-making, application of deep reinforcement learning, intention guiding, deep deterministic policy gradient, self-learning.

I. INTRODUCTION

As a flourishing air confrontation force, UAV plays a more and more significant role in the modern warfare. Compared with MAV, UAV has the unique advantages of zero casualties, sustained operations, low cost and outstanding maneuverability. By the use of MAV/UAV cooperative platform, the deficiency of current autonomous level of UAV can be made up in some extent, and the survivability in warfare can be enhanced, which is beneficial to the victory of the war [1]. The cooperative mode is one of the development directions of the potential penetration fighter in the sixth generation [2]. In 2016, the University of Cincinnati developed an intelligent pilot named Alpha, using the genetic fuzzy search tree method to implement air confrontation in virtual with the famous pilot Colonel Gene Lee [3]. In this confrontation, Gene Lee was thrashed. That same year, the U.S. military launched the ''Commander's Virtual Staff'' Project'' [4] for reducing the cognitive burden on commanders by means of integrating artificial intelligence technology and information systems. In August of 2019, the U.S. Air Force Research Laboratory and DZYNE Company experimented with the ''ROBOpilot'' program in Dugway [5]. The researchers used cameras to collect aircraft dashboard data and took mechanical transmission equipment to control joystick, pedal and switch, etc. The U.S. put forward ''loyal wingman'' program in 2015 with the purpose of providing reasonable and fast suggestions for pilots [6]. It completed the first flight test of the XQ-58A ''Valkyrie'' UAV in Arizona in March of 2019, which was a typical type of loyal wingman. In addition, Russia announced that a flight test has accomplished with ''Hunter'' UAV and Su-57 MAV in September of 2019 [7].

Airborne communication networks (ACNs) were utilized for MAV/UAV cooperative system [8], [9], where the UAV

The associate editor coordinating the review of this manuscript and approving it for publication was Cong Pu.

receiving commands from MAV was mainly depended on ACNs. Hence, it was one of the most significant guarantees for cooperative engagement of MAV/UAV. Compared to terrestrial wireless networks, ACNs were characterized by frequently changed network topologies and more vulnerable communication connections [10]. The latest research indicated that studying the technologies of control, networking, and transmission would help design scalable, practical, and fault tolerant ACNs [11]. Besides, the DRL method can also be carried out to develop communications coverage, energy consumption and connectivity [12].

Since the classical observe-orient-decide-act (OODA) operational theory [13], it is critical to research on the intelligent decision-making of MAV/UAV, which is of significance to improve the level of air confrontation [14]. The current air confrontation decision-making methods can be divided into two main categories [15]: one is traditional strategy contains differential game and expert system, etc. The other is intelligent strategy, such as genetic algorithm, influence diagram, ant colony algorithm, artificial immune system, etc. The core of the decision-making problem is to predict the possible states of future. However, the above methods can not realize the long-term prediction in real time due to the influence of the computational complexity and other factors. Recent years, artificial intelligence (AI) has drawn much attention and researchers have implemented the technology for decision-making [16], [17], but three main deficiencies are presented. First is the limitation of deviation from reality caused by discretization of action space or simplification of reward function, etc. [18], [19]. In [20], an autonomous maneuver decision model was proposed for the UAV short-range air combat based on RL. According to the requirement of discrete control, deep Q network (DQN), a traditional DRL method for solving discrete space was applied. Second, the efficiency of RL should still be further improved although numerous researchers have focused on the issue [21], [22]. Typically, DDPG, an algorithm for solving continuous controlling models was proposed by DeepMind in 2016 [17]. After that, in July of 2017, OpenAI introduced proximal policy optimization (PPO) [23], a family of policy optimization methods. The method had some of the benefits of trust region policy optimization, but they were much simpler to implement and higher efficiency. Then Google DeepMind proposed Distributed PPO on this basis [24]. Third, the application and implementation of decision-making are relative monotonous, such as for UAV, attacking implementation is in dominant [25], [26] and for unmanned vehicle, achieving the goal of safely driving is in majority [27], which ignores the directive effect of commander in cooperative system. To alleviate the third trouble, in [28], the mapping from state to motivation in standard Q learning is transformed into three layer mapping, i.e., state-motivation-action, and simulated annealing algorithm is adopted to improve the RL process. Nevertheless, if-then rule is used during the mapping of the motivation layer to the action layer, which brings great subjectivity and should be elaborately considered in future.

To the best of our knowledge, combining DRL with air confrontation decision-making of MAV/UAV as well as researching the feasibility and applicability is a novel application and has not been deep investigated. Due to the core of this paper is the application of DRL, so which detailed method of DRL for solving the problem of air confrontation decision-making of MAV/UAV is not vital. The method for continuous controlling, such as DDPG, PPO, DPPO, are all suited. To this end, the improved DDPG is investigated in this work.

The main contributions and innovations of this paper can be summarized as follows:

i) We combine DRL with air confrontation decision-making of MAV/UAV, where the innovation is reflected in proposing the four typical intentions including Head-on attack, Fleeing, Pursuing and Energy-storing, corresponding to four optimization models.

ii) We improve the traditional DDPG algorithm by means of reconstructing the replay buffer. The mechanism of experience judgment and improvement of sampling strategy are put forward, which result in the superiority in the aspect of convergence and stability of training.

iii) We utilize self-learning of different intentions to improve the degree of intelligent decision-making, which is of significance to achieve real intelligent air confrontation in future.

This paper is organized as follows. In Section II, the dynamics model of UAV and the basic structure of DDPG are built. The framework of decision-making of MAV/UAV containing four typical intentions and the reconstruction of replay buffer in DDPG are presented in Section III. The simulation is executed in Sections IV. Finally, we conclude with suggestions for future research in Section V.

II. PROBLEM FORMULATION

A. DYNAMICS EQUATION OF UAV

In this paper, it is assumed that the tasks of MAV/UAV cooperative system are mainly performed by UAV. For MAV, in the rear of UAV, is relatively safe. Its work is to commanding in the light of battlefield situation. Hence, only the dynamics equation of UAV needs to be discussed.

In the decision-making process of air confrontation, the main focus is on real-time position and speed information of the two sides. Therefore, the model of UAV can be described commendably by simplified point-mass equations. In the inertial frame, the dynamics equation of UAV is given as

$$\begin{cases} \dot{x} = v_{\mathrm{u}} \cos\gamma \cos\psi \\ \dot{y} = v_{\mathrm{u}} \cos\gamma \sin\psi \\ \dot{z} = v_{\mathrm{u}} \sin\gamma \\ \dot{v}_{\mathrm{u}} = \dfrac{T\cos\alpha - D}{m} - g\sin\gamma \\ \dot{\gamma} = \dfrac{(L + T\sin\alpha)\cos\mu}{m v_{\mathrm{u}}} - \dfrac{g}{v_{\mathrm{u}}}\cos\gamma \\ \dot{\psi} = \dfrac{(L + T\sin\alpha)\sin\mu}{m v_{\mathrm{u}} \cos\gamma} \end{cases} \tag{1}$$

where v_u represents the velocity of UAV; γ, ψ and μ indicate the UAV's flight-path angle, heading angle, and flight-path bank angle respectively; α is attack angle, m is the mass of UAV, regarding as a constant in the paper; T is the engine thrust, g is acceleration of gravity;Air resistance D and lift L can be donated as

$$\begin{cases} L = \frac{1}{2}\rho v_u^2 S C_L \\ D = \frac{1}{2}\rho v_u^2 S C_D \end{cases} \tag{2}$$

in which S is reference cross-sectional area for UAV; C_L and C_D are coefficient of lift and air resistance, respectively; ρ is atmospheric density. When UAV is in the troposphere, ρ varies with the altitude h, as given in (3).

$$\rho = 1.225^* \left[(288.15 - 0.0065^* h)/288.15\right]^{4.25588} \tag{3}$$

The engine thrust T of UAV can be defined as

$$T = \delta T_{\max} \tag{4}$$

in which $T_{\max}$ is the maximum thrust of the engine, δ is throttle with the range of [0, 1].

As discussed above, given the input [δ α μ] and the time step of decision-making Δt, we can easily obtain the next states of UAV accordance with Runge-Kutta method [26].

B. DDPG ALOGORITHM

The goal of RL is to find the policy that maximizes the expected return which is defined by a reward function. The interaction between agent and environment is modeled as a Markov Decision Process (MDP) [29], which contains the state space, action space, reward function, discount coefficient and probability of transition. Specially, in the case of model-free RL, only the former four elements need to be analyzed.

Classical reinforcement learning is difficult or impossible to traverse all cases in the face of high dimension of state and action space, which may result in slow convergence of the algorithm or the inability to learn reasonable strategies. An effective way to solve the above problems is to use the method of function approximation to express the value function or strategy explicitly [21]. For the complex nonlinear function, the deep neural network has a better approximate effect, so it has become a trend to introduce the deep neural network as a tool into RL for approximating the value function or strategy function in recent years [20].

Faced with the model of intelligent decision-making of MAV/UAV, whose state space is multi-dimensional, one of the most popular DRL algorithms called DDPG is adopted. For the algorithm, the idea of Deterministic Policy-Gradient algorithms, Actor-Critic structure, and DQN are combined. The Policy-Gradient is a kind of RL based on the probabilistic theory, which represents the optimal decision of MDP by a probability distribution function. For the algorithm, the whole action space should be integrated in each step of the decision-making process, so the demand of computational power is extremely high. In this context, Siliver [30] proves the existence of Deterministic Policy-Gradient and integrates it into the Actor-Critic framework, which consists of critic network and actor network.

To make the learning stable and robust, similar to DQN [16], DDPG also deploys experience replay and evaluate/target network. We use $Q(s, a|\theta^Q)$ and $\mu(s|\theta^\mu)$ to denote the evaluate network of critic network and actor network, respectively, corresponding to the parameter θ^Q and θ^μ, while Q' and μ' are represented as target network with the parameter $\theta^{Q'}$ and $\theta^{\mu'}$.

For critic network, the input of which are action a and state s, output is Q value, i.e. $Q(s, a)$. The update of the network parameter is mainly dependent on minimizing the loss between Q value of evaluate network and target network. The former Q value is estimated by evaluate network, while the latter is obtained by adding the Q' and reward r_i. The process of update can be expressed as (5)-(6)

$$y_i = r_i + \gamma_d Q'\left(s_{i+1}, \mu'\left(s_{i+1}|\theta^{\mu'}\right)|\theta^{Q'}\right) \tag{5}$$

$$\min_{\theta^Q} L = \min_{\theta^Q} \frac{1}{N}\sum_{i=1}^{N}\left(y_i - Q\left(s_i, a_i|\theta^Q\right)\right)^2 \tag{6}$$

where γ_d is discount coefficient of the reward function.

For actor network, the input is s, output is a, and using the sampled policy gradient to update, which can be described as

$$\nabla_{\theta^\mu} J \approx \frac{1}{N}\sum_{i=1}^{N} \nabla_a Q\left(s, a|\theta^Q\right)_{|s=s_i, a=\mu(s_i)} \nabla_{\theta^\mu}\mu\left(s|\theta^\mu\right)|_{s_i} \tag{7}$$

Target network is updated according to:

$$\begin{aligned} \theta^{Q'} &\leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'} \\ \theta^{\mu'} &\leftarrow \tau\theta^\mu + (1-\tau)\theta^{\mu'} \end{aligned} \tag{8}$$

in which τ is the parameter of update rate.

In summary, the flow-chart of DDPG algorithm is shown in Figure 1. The number in the figure is the sequence the algorithm runs.

III. ALGORITHM DESIGN

Based on the DQN algorithm, the DDPG continues the idea of mode-free RL, and state space, action space, reward function and discount coefficient should be studied respectively. Among of them, the action space is derived from the dynamics equation of UAV (Section II.A), and the state space is mainly determined accordance with the model of air confrontation (Section III.A), which is also based on the dynamics equation. Besides, the reward function in four typical intentions is introduced in Section III.B, which is a key to realize the intelligent decision-making of MAV/UAV. The discount coefficient is designed as universal rules [20]. In addition, the improvement of the DDPG structure is shown in Section III.C.

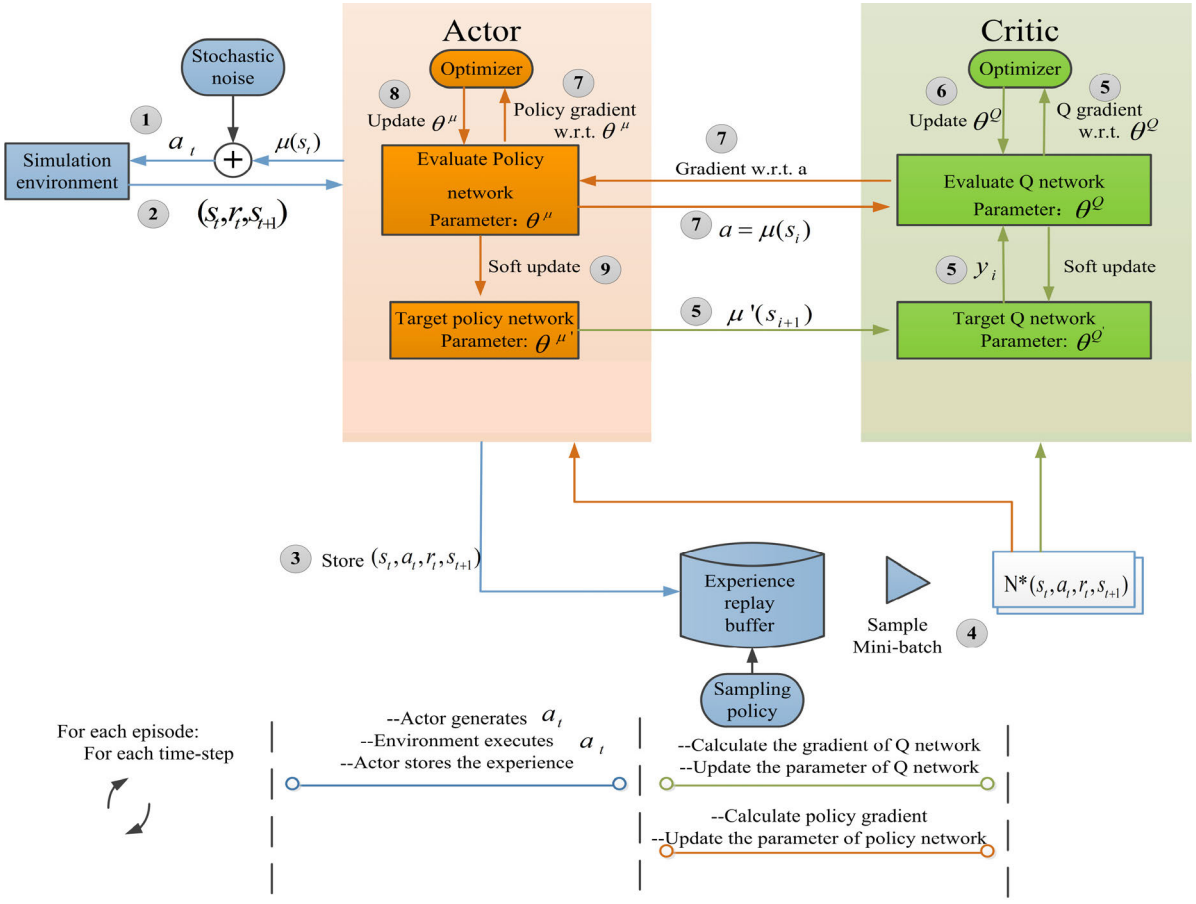


FIGURE 1. The flow-chart of DDPG algorithm.

A. STATE AND ACTION SPACE OF DRL IN AIR CONFRONTATION DECISION-MAKING MODEL

1) DETERMINATION OF STATE SPACE

From section II.B, we know that the deep neural network is a significant tool to approximate the value or policy function. Thus, the state space in air confrontation decision-making model, regarding as the input of neural network, is critical to determine.

It is assumed that the red is represented as us while the blue is enemy, which are distinguished by the subscript r and b, respectively. The situation in battlefield, especially the relative of the red and blue, may not be reflected commendably if the states of UAV in (1) are used as the input of neural network directly. Considering the relative states acquired expediently by GPS, LIDAR, and other sensors, $\boldsymbol{x_0}$ are selected as the state space, which contain 10 scalars

$$\boldsymbol{x}_0 = [d, q_r, q_b, \beta, \Delta h, \Delta V, v_{\mathrm{u}}, h, F_1, F_2] \tag{9}$$

where d denotes the relative distance of the two sides; q_r and q_b represent the angle between the velocity vector and the centroid; β is the angle between the velocity vector of the red and blue; Δh and ΔV are the difference of height and velocity of the two sides, respectively. Moreover, v_{u} and h are also included in $\boldsymbol{x_0}$ because of that $T_{\max}$ is a function of v_{u} and h,

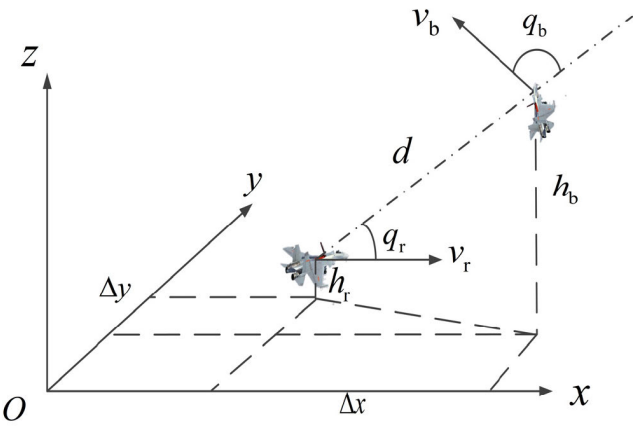


FIGURE 2. Definition of partial states of air confrontation model.

and it makes sense during the process of controls conversion; F_1 and F_2 denote the flag of reaching the goal and exceeding the limits of states, respectively.

Unknowns in $\boldsymbol{x_0}$ can be derived by air confrontation model, which was illustrated in (10), as shown at the bottom of the next page, and Figure 2.

Note that the states in $\boldsymbol{x_0}$ need to be normalized before input to the neural network.

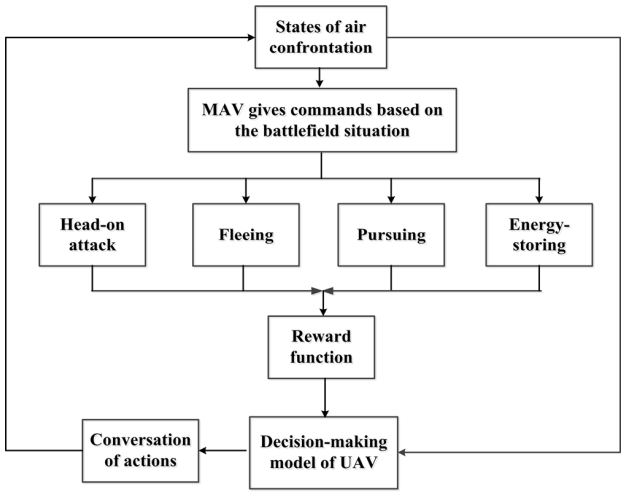


FIGURE 3. The framework of air confrontation intelligent decision-making of MAV/UAV.

2) DETERMINATION OF ACTION SPACE

The action space $\boldsymbol{a}$ generated in DDPG can be related to the controls of UAV, i.e. $\boldsymbol{u}_0 = [\delta, \alpha, \mu]$. In the light of physical meaning, we know that the range of throttle δ is the positive while the attack angle α and flight-path bank angle μ have no limit of positive or negative, so the relationship between $\boldsymbol{a}$ and $\boldsymbol{u}_0$ is designed as

$$\begin{cases} \delta = Sigmoid(a_1) \\ \alpha = Tanh(a_2)^* \boldsymbol{\alpha}_v \\ \mu = Tanh(a_3)^* \boldsymbol{\alpha}_\mu \end{cases} \tag{11}$$

in which *Sigmoid* and *Tanh* are activation function of neural network, of which the range are [0,1] and [−1,1] respectively, corresponding to the physical meaning mentioned above; $\boldsymbol{\alpha}_v$ and $\boldsymbol{\alpha}_\mu$ are the range of α and μ.

Next section will introduce the effect of MAV on the cooperative system through designing the reward functions in different intentions.

B. REWARD FUNCTION OF MAV INTENTIONS

To exhibit the advantage of MAV intelligence, several typical intentions were predefined by MAV. Reward function and constraint conditions were designed to describe different intentions. Then the decision-making model of UAV was trained based on the steses of $\boldsymbol{x_0}$ acquired by the sensors. As a result, the neural networks in diverse intentions were obtained through training, and next states could be got by executing the action of $\boldsymbol{a}$. The framework of air confrontation intelligent decision-making of MAV/UAV was shown in Figure 3.

In this paper, short-range air confrontation was researched. Assuming that the aerial cannon was adopted as the weapon of the two sides, and the relative angle, distance, height and velocity were selected as indicators for estimating the air confrontation [31], where the relative angle and distance were focus on the safety especially the condition of emit while the energy was highlighted by the height and velocity. Obviously, more chances for maneuvering can be got in case of high energy. Furthermore, a positive reward was gained when the goal of task was achieved while a negative reward was got in failure. The four intensions were introduced as follows.

1) HEAD-ON ATTACK

The goal of the intension was to accomplish the task of strike and tail chasing state was dispensable to carry out attacking, so called Head-on attack. When $q_r \in \boldsymbol{q}_{\text{attack}}$ and $d < d_{\text{emit}}$, the aerial cannon can be emitted, where $d > d_{\text{escape}}$ and d_{emit} were the aspect angle and distance of allowed attack; When the blue got the identical condition or $d > d_{\text{escape}}$, the red failed, in which d_{escape} was the escaped distance. Additionally, exceeding the limits of UAV state was also disallowed, which was regarded as failure. The detailed reward function was expressed as

$$\begin{cases} r_{a,d} = \begin{cases} (q_b - q_r)\exp\left(-(\dfrac{d - d_{\text{emit}}}{d_{\text{emit}}})^2\right), & \\ \qquad\qquad q_r \leqslant q_b, d > d_{\text{emit}} \text{ and } q_b \notin \boldsymbol{q}_{\text{attack}} \\ (q_b - q_r), \quad q_r \leqslant q_b, d < d_{\text{emit}} \text{ and } q_b \notin \boldsymbol{q}_{\text{attack}} \\ 0, \qquad \text{else} \end{cases} \\ r_v = \exp\left(-\dfrac{v_r - V_0}{V_0}\right) z \\ r_h = \begin{cases} \exp\left(-\dfrac{\Delta h - \Delta h_0}{\Delta h_0}\right), & \Delta h \geqslant \Delta h_0 \\ \exp\left(\dfrac{\Delta h - \Delta h_0}{\Delta h_0}\right), & \Delta h < \Delta h_0 \end{cases} \\ r_{\text{result}} = \begin{cases} 1, & q_r \in \boldsymbol{q}_{\text{attack}} \text{ and } d < d_{\text{emit}} \\ -1, & q_b \in \boldsymbol{q}_{\text{attack}} \text{ and } d < d_{\text{emit}}; \\ & \text{or } d > d_{\text{escape}}; \text{ or } \boldsymbol{s} \notin \boldsymbol{s}_r \\ 0, & \text{else} \end{cases} \end{cases} \tag{12}$$

where $r_{a,d}$, r_v, r_h and r_{result} represented the reward of angle and distance, velocity, height, result, respectively; V_0 and Δh_0 were known; $\mathbf{s}_r$ were the range of the red states of UAV. Specially, it was assumed that only a state of success or failure maintained 3 seconds or more, the r_{result} changed to non-zero value; If less than 3 seconds, the flags would operate to guide for training.

$$\begin{cases} d = \sqrt{(x_b - x_r)^2 + (y_b - y_r)^2 + (z_b - z_r)^2} \\ q_r = \arccos\left\{[(x_b - x_r)\cos\psi_r \cos\tau_r + (y_b - y_r)\sin\psi_r \cos\tau_r + (z_b - z_r)\sin\tau_r]/d\right\} \\ q_b = \arccos\left\{[(x_r - x_b)\cos\psi_b \cos\tau_b + (y_b - y_r)\sin\psi_b \cos\tau_b + (z_r - z_b)\sin\tau_b]/d\right\} \\ \beta = \arccos\left(\cos\psi_r \cos\tau_r \cos\psi_b \cos\tau_b + \sin\psi_r \cos\tau_r \sin\psi_b \cos\tau_b + \sin\tau_r \sin\tau_b\right) \end{cases} \tag{10}$$

2) PURSUING

The goal of the intension was to accomplish the task of accurate attacking based on flight safety, and it was also adapted to the situation the red dominated in energy. Compared to the Head-on attack, to accomplish the task, tail chasing was necessary. For reward function, the difference with (12) was described as follows

$$r_{\text{result}} = \begin{cases} 1, & q_r \in \boldsymbol{q}_{\text{attack}},\ q_b \in \boldsymbol{q}_0 \text{ and } d < d_{\text{emit}} \\ -1, & q_b \in \boldsymbol{q}_{\text{attack}} \text{ and } d < d_{\text{emit}}; \\ & \text{or } d > d_{\text{escape}};\ \text{or } \boldsymbol{s} \not\subset \boldsymbol{s}_{\text{r}} \\ 0, & \text{else} \end{cases} \tag{13}$$

where $q_b \in \boldsymbol{q}_0$ indicated that the aspect angle of the blue should be in the range of $\boldsymbol{q}_0$ when the red emitted. In general, $\boldsymbol{q}_0$ represented as an obtuse angle interval.

The Pursuing intention was a common selection for MAV because of it was able to guarantee accuracy and stability of strike.

3) FLEEING

The goal of the intension was to keep away the enemy's strike, and it was also adapted to the situation the blue had advantage in performance or the red was immersed in trouble. To ensure the safety of the red in air confrontation, the constraints $q_b \in \boldsymbol{q}_0$ should be added besides $d > d_{\text{escape}}$, which demanded that the red flee in the opposite direction to the blue. The detailed reward function was shown in the following

$$\begin{cases} r_{a,d} = \begin{cases} (q_b - q_{0\_\min}) \exp\left(-(\dfrac{d - d_{\text{escape}}}{d_{\text{escape}}})^2\right), \\ \qquad q_b \notin \boldsymbol{q}_{\text{attack}} \text{ and } d \leqslant d_{\text{escape}} \\ (q_b - q_{0\_\min}),\ q_b \notin \boldsymbol{q}_{\text{attack}} \text{ and } d > d_{\text{escape}} \\ 0, \qquad \text{else} \end{cases} \\ r_v = \exp\left(-\dfrac{v_r - V_0}{V_0}\right) \\ r_h = \begin{cases} \exp\left(-\dfrac{\Delta h - \Delta h_0}{\Delta h_0}\right), & \Delta h \geqslant \Delta h_0 \\ \exp\left(\dfrac{\Delta h - \Delta h_0}{\Delta h_0}\right), & \Delta h < \Delta h_0 \end{cases} \\ r_{\text{result}} = \begin{cases} 1, & q_b \in \boldsymbol{q}_0 \text{ and } d > d_{\text{escape}} \\ -1, & q_b \in \boldsymbol{q}_{\text{attack}} \text{ and } d < d_{\text{escape}};\ \text{or } \boldsymbol{s} \not\subset \boldsymbol{s}_{\text{r}} \\ 0, & \text{else} \end{cases} \end{cases} \tag{14}$$

where $q_{0\_\min}$ was the minimum of $\boldsymbol{q}_0$.

4) ENERGY-STORING

With the development of performance in aerial vehicle, energy theory has been paid more and more attention, which supported a amount of unconventional actions, such as large overload maneuver, poststall maneuver and so on. Theses actions provided more opportunity for reversing disadvantages in air confrontation. As mentioned above, the total energy, contains kinetic energy and potential energy, was selected as the estimating indicators for the intention, which was in favor of storing energy and waiting for the new commands of MAV. The reward function was described as

$$\begin{cases} r_{a,d} = \begin{cases} (q_b - q_{0\_\min}) \exp\left(-(\dfrac{d - d_{\text{emit}}}{d_{\text{emit}}})^2\right), \\ \qquad d_{\text{emit}} < d < d_{\text{escape}} \text{ and } q_b \notin \boldsymbol{q}_{\text{attack}} \\ 0, \qquad \text{else} \end{cases} \\ r_v = \exp\left(-\dfrac{v_r - V_0}{V_0}\right) \\ r_h = \begin{cases} \exp\left(-\dfrac{\Delta h - \Delta h_0}{\Delta h_0}\right), & \Delta h \geqslant \Delta h_0 \\ \exp\left(\dfrac{\Delta h - \Delta h_0}{\Delta h_0}\right), & \Delta h < \Delta h_0 \end{cases} \\ r_{\text{result}} = \begin{cases} 1, & E_r > \eta E_b \text{ and } d_{\text{emit}} < d < d_{\text{escape}} \\ -1, & q_b \in \boldsymbol{q}_{\text{attack}} \text{ and } d < d_{\text{emit}}; \\ & \text{or } d > d_{\text{escape}};\ \text{or } \boldsymbol{s} \not\subset \boldsymbol{s}_{\text{r}} \\ 0 & \text{else} \end{cases} \end{cases} \tag{15}$$

in which E_r and E_b were the total energy of the red and blue, respectively; η was the coefficient of proportionality in energy.

Up to now, four intentions of MAV have been introduced. To sum $r_{a,d}$, r_v, r_h, r_{result} and F_1, F_2 by weights, the total reward function was obtained as

$$r = \omega_{a,d} r_{a,d} + \omega_v r_v + \omega_h r_h + \omega_{\text{result}} r_{\text{result}} + \omega_{f_1} F_1 + \omega_{f_2} F_2 \tag{16}$$

where weights ω were empirical values, taken differently in respective intentions.

C. RECONSTRUCTION OF REPLAY BUFFER

In DDPG algorithm, experience replay method was adopted and replay buffer was used as a tool to store experiences. The means of sampling can decrease the relevance of the experiences, which was in favor of the stability of neural network training. However, sampling randomly may result in the low efficiency of training and the poor performance in convergence. To deal with the problem, reconstruction of replay buffer was proposed in this section. Two ways were introduced in the following.

1) MECHANISM OF EXPERIENCE JUDGMENT

Experience judgment meant that success experience and failure experience were divided into two replay buffers. If the failure requirements of (12)-(15) were satisfied, the experience of this step would be recorded in the failure replay buffer, which was denoted as R_f. On the contrary, other experiences were regarded as the success storing to R_s on temporary. As we known, time delay was existed in the reward process of RL, so a number of experiences before access to the failure, which stored in R_s, were also relevant to the failure. We should extract these experiences from R_s into R_f accordance with the proportion of η_f.

Algorithm 1 Reconstruction of Replay Buffer

```
for each step do
    if failure then
        Store experience into R_f
        Extract as amount as N_f experiences from R_s
        into R_f (with proportion of η_f)
        Sample η_s*Mini-batch experiences from R_s
        Sample (1 − η_s)*Mini-batch experiences
        from R_f
    else
        Store experience into R_s
    end if
    if R_s and R_f are full then
        Substitute the new experience for the past
        Train the neural network with Mini-batch
        experiences
    else
        if R_s or R_f are full then
            Substitute the new experience for the past
        else
            Continue
        end if
    end if
    Until failure or maximum step
end for
```

2) IMPROVEMENT OF SAMPLING STRATEGY

In original DDPG algorithm, randomly sampling was carried out. To make the sampling process more valid, sampling in proportion from R_f and R_s was adopted to instead. Assuming that the amount of sampling was *Mini-batch*, and the sampling proportion in R_s was η_s. Taken the learning efficiency in the earlier stage and the trouble of local optimum in the later stage into consideration, η_s should be decreased with progressed.

The reconstruction of replay buffer was presented in the following algorithm.

IV. EXPERIMENTS

In this paper, Python language was selected to implement our algorithm, and TensorFlow module was applied to supporting DRL. The actor and critic network were both used simple fully connected network architecture with a few hidden layers. It was because in the process of debugging, we found that the effect of too deep layers was often counterproductive, which was related to the exploration and exploitation theory of RL. The viewpoint was proposed by Fang, and he verified it in his thesis [32]. Finally, two hidden layers with 600 and 300 units were determined in the experiments, the learning rate was 1×10^{-4}. Moreover, each intention was trained 15000 episodes, which was about 10^6 steps. Significantly, the amount of training episodes was determined by comparative experiments, under which the results can be revealed commendably. Other parameters were listed in Table 1.

TABLE 1. The parameters of DDPG Algorithm.

Name	Specifications
Δt	1 s
Mini-batch	32
R_s	30000
R_f	10000
γ_d	0.9
τ	0.01
$\boldsymbol{q}_{\text{attack}}$	$[-\pi/6, \pi/6]$
d_{emit}	5000 m
d_{escape}	15000 m
V_0	100 m/s
Δh_0	500 m
$\boldsymbol{q}_0$	$[2\pi/3, \pi] \cup [-2\pi/3, -\pi]$
η	1.5
η_f	0.2

TABLE 2. The basic parameters of F-4C fighter.

Name	Specifications
$\mathbf{s}_r$	$[(-\infty,+\infty),(-\infty,+\infty),(1000,10000),$ $(80,400),(-\pi/4,\pi/4),(-2\pi,2\pi)]$
m	14680 kg
S	49.24 m^2
$\boldsymbol{\alpha}_v$	$[-\pi/18,\pi/6]$
$\boldsymbol{\alpha}_\mu$	$[-5\pi/6,5\pi/6]$

It was assumed that the type of both sides was F-4C fighter. The basic parameters of F-4C fighter were in the following table.

Thrust data for the F-4C was taken from data originally presented in [33], the maximum available thrust $T_{\max}$ was expressed in units of 1000 lb (i.e. 4436.26 N), and was a function of the Mach number $\bar{v}_{\text{u}}$ and altitude h in units of 10000 ft (i.e. 3048 m), which was shown in

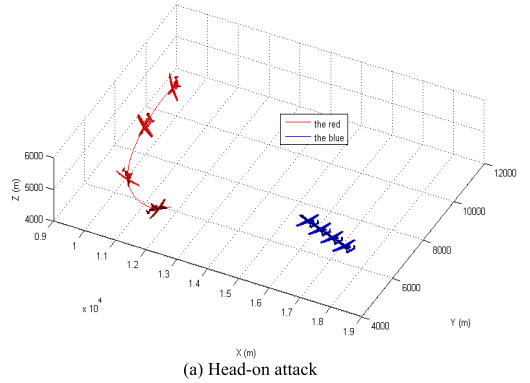
$$T_{\max} = \begin{bmatrix} 1 \\ \bar{v}_{\text{u}} \\ \bar{v}_{\text{u}}^2 \\ \bar{v}_{\text{u}}^3 \\ \bar{v}_{\text{u}}^4 \end{bmatrix}^{\text{T}} \begin{bmatrix} 30.21 & -0.668 & -6.877 & 1.951 & -0.1512 \\ -33.80 & 3.347 & 18.13 & -5.865 & 0.4757 \\ 100.80 & -77.56 & 5.441 & 2.864 & -0.3355 \\ -78.99 & 101.40 & -30.28 & 3.236 & 0.1089 \\ 18.74 & -31.60 & 12.04 & -1.785 & 0.09417 \end{bmatrix} \begin{bmatrix} 1 \\ h \\ h^2 \\ h^3 \\ h^4 \end{bmatrix} \tag{17}$$

C_L and C_D were determined by

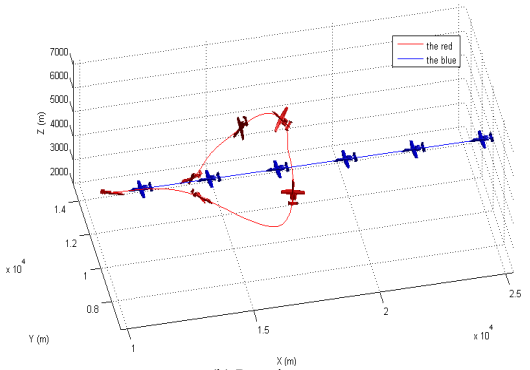
$$\begin{cases} C_L = (-0.0434 + 0.1369\alpha)\sin\alpha + (0.131 + 3.0825\alpha)\cos\alpha \\ C_D = (0.0434 - 0.1369\alpha)\cos\alpha + (0.131 + 3.0825\alpha)\sin\alpha \end{cases} \tag{18}$$

A. TRAINING RESULT OF FOUR INTENTIONS

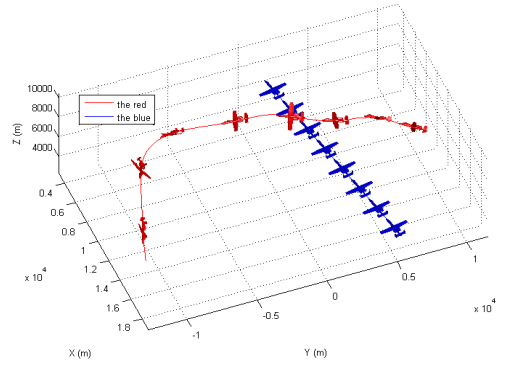
In the process of training the intensions, on one hand, to improve the convergence performance, the blue was set to fly in straight line; on the other hand, to guarantee the



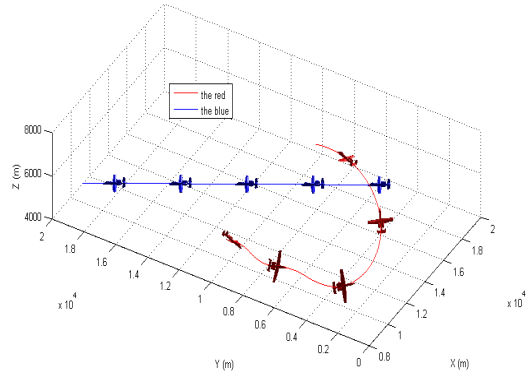
(a) Head-on attack



(b) Pursuing



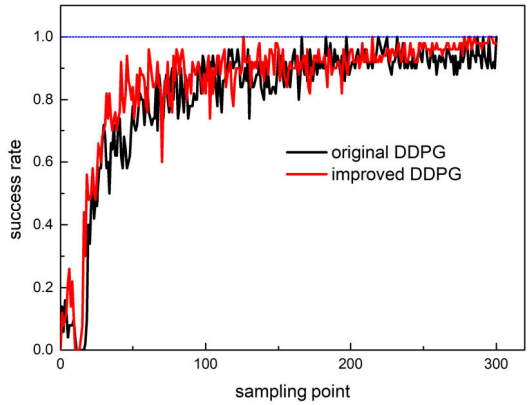
(c) Fleeing



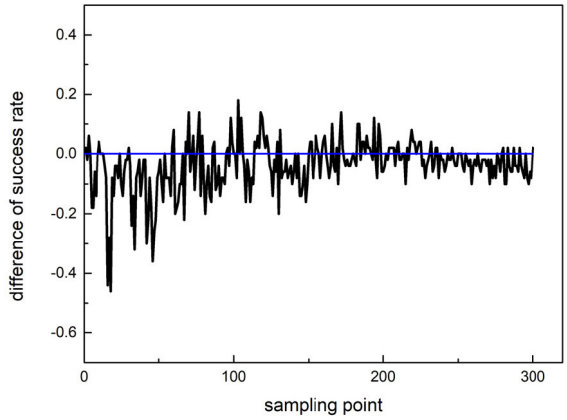
(d) Energy-storing

FIGURE 4. Confrontation maneuvering trajectory under different intentions.

diverse of training samples, the initial states of the blue were generated randomly. If the initial state of the blue was



(a) Success rate of two algorithms



(b) The difference of success rate between original DDPG and improved DDPG.

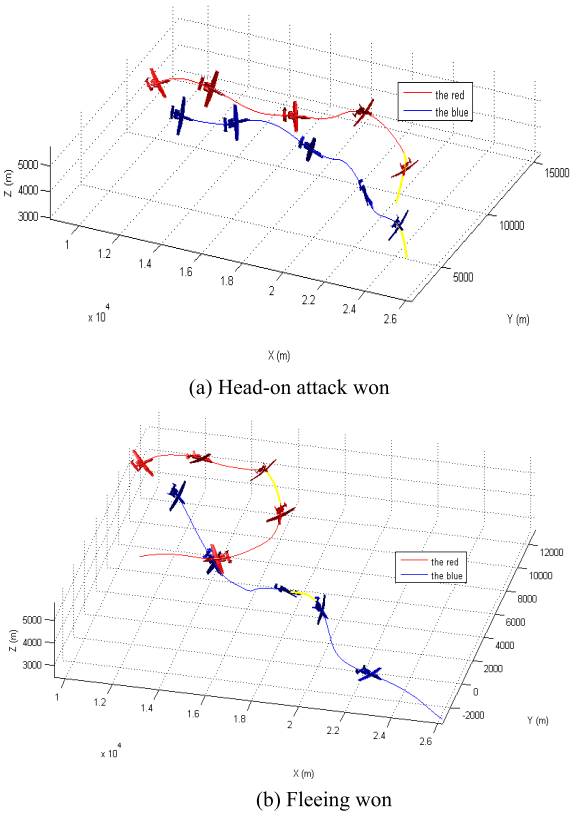
FIGURE 5. Comparison of improved DDPG and original DDPG.

fixed and idealized, the learning capacity would be decreased, which has been verified in previous experiments. The training results were shown in Figure 4, in which the attitude of UAVs was also expressed.

From Figure 4 (a) and (b), we can see that the success condition of Pursuing was more rigorous than Head-on attack. Compared to Head-on attack, Pursuing paid more attention to accurate striking and flight safety via bypassing to the rear of the blue. However, the intention may miss the optimal opportunity of striking owning to the discreetness. Combining to Figure 4 (c) and (d), we knew that Fleeing was a direct intention to avoid striking, but little energy can be accumulated. By contrast, Energy-storing can not only guarantee the red to avoid striking, but also attempt to search for the opportunity of counter-attack by accumulating the advantage of height and velocity.

B. VERIFICATION OF THE IMPROVED DDPG ALGORITHM

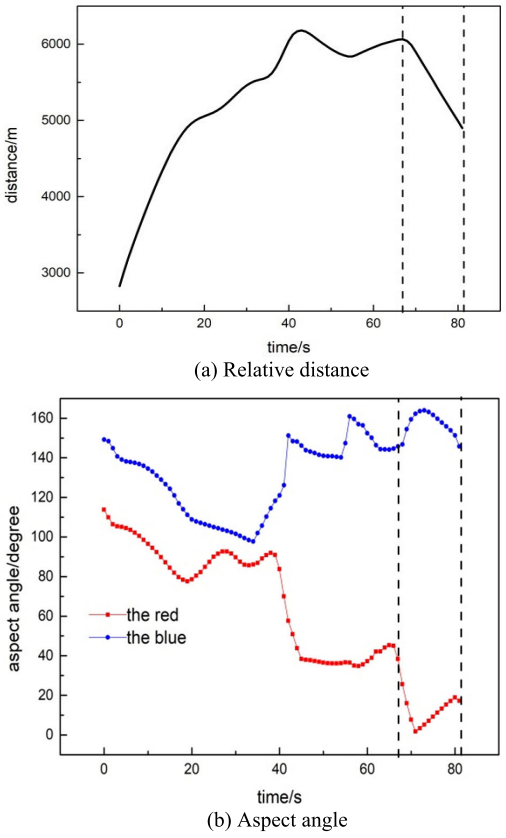
To validate the reconstruction of replay buffer in DDPG algorithm, an identical model with the same intention was trained by the improved DDPG and original DDPG. After 15000 episodes, the success rate of task was taken as an index to evaluate the effect of training. In order to decrease



(a) Head-on attack won

(b) Fleeing won

FIGURE 6. Result of Head-on attack intention versus Fleeing intension.



(a) Relative distance

(b) Aspect angle

FIGURE 7. Situation relationship for Head-on attack winning.

the effect of data fluctuation and highlight the tendency of training result, data was averaged per 50 episodes to express the success rate. The η_s was determined by

$$\eta_s = 0.9^* \exp(-i/15000) \tag{19}$$

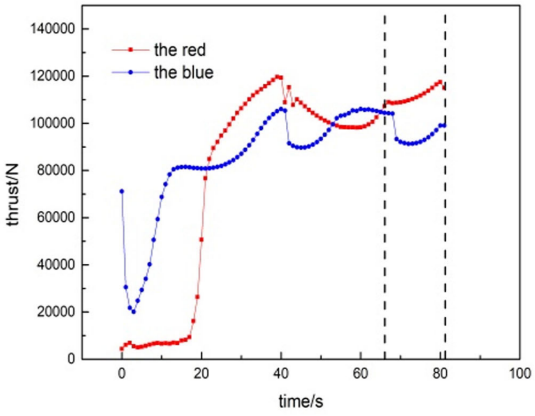
where $i = 1, 2 \cdots 15000$ indicated the index of episodes.

Comparison of improved DDPG and original DDPG was shown in Figure 5.

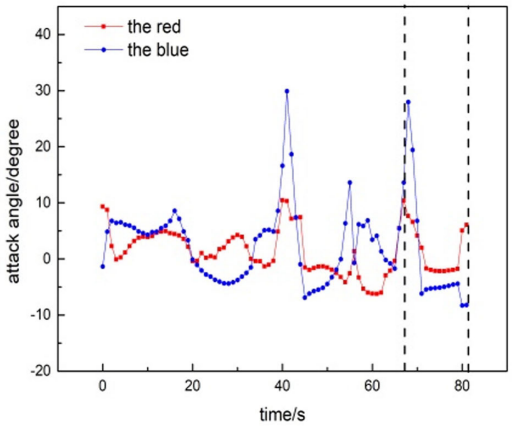
From Figure 5 (a), the success rate of both algorithms was increased over 90% from about 10% gradually, but the improved algorithm had faster convergence in previous period and weaker fluctuation when access to high success rate. Combining to Figure 5 (b), the difference of success rate between original DDPG and improved DDPG, we can see that the improved one had 44% ahead at most in the first 50 sampling point. In the later of sampling, especially the last 50 sampling point, the difference was decreased, but the improved one possessed higher degree of stability while the original fluctuated obviously. It was because that the improved DDPG kept improving the neural network during trial and error with the adjustment of η_s while the original DDPG sampled randomly all the time. In summary, the reconstruction of replay buffer was a valid way to improve DDPG algorithm. Additionally, from the results we can see that the amount of training episodes was selected reasonably. If it was much less than 15000, the convergence effect would not be performed commendably.

C. DEMONSTRATION OF AIR CONFRONTATION

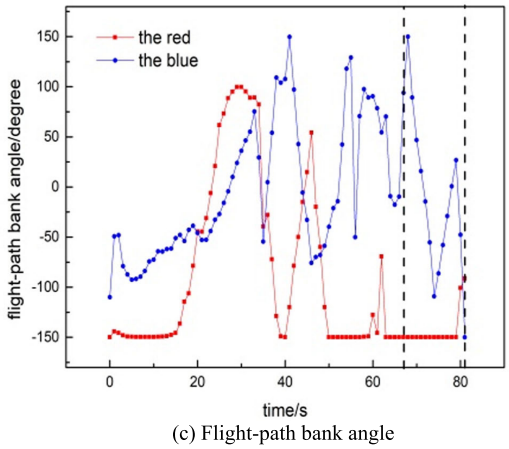
To demonstrate the scenarios of air confrontation, the trained neural network in section IV.A was reloaded. Air confrontation of the cooperative system was led by the virtual MAV. Assuming that the initial relative distance was in favor of striking for the red while the aspect angle was beneficial to flee for the blue, and the result of Head-on attack intention versus Fleeing intension was shown in Figure 6. From Figure 6 (a) and (b), we can see that winning or losing in the same initial environment (only tiny stochastic noise were drew in γ and ψ) were uncertain, where was determined by the intellgent decision-making of the two sides. Situation relationship and history of controls in the two episodes were shown in Figure 7- Figure 10. When Head-on attack won, the initial aspect angle of the red was far away from the allowed aspct angle for attack, so the red adjusted the aspect angle in priority by means of decreasing the attack angle with low thrust in the earlier 20 s of air confrontation. Meanwhile, the blue increased thrust in a short time with an attempt to enlarge the distance between the red. At the time of 38 s and 67 s, two approaching maneuvers of the red was carried out, especially the later (from 67 s to 81 s, trajectory shown in Figure 6 (a) with yellow lines and border



(a) Thrust

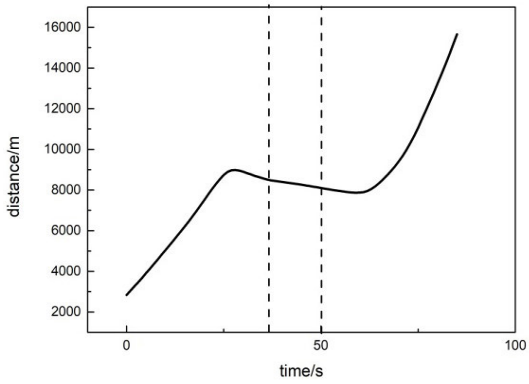


(b) Attack angle

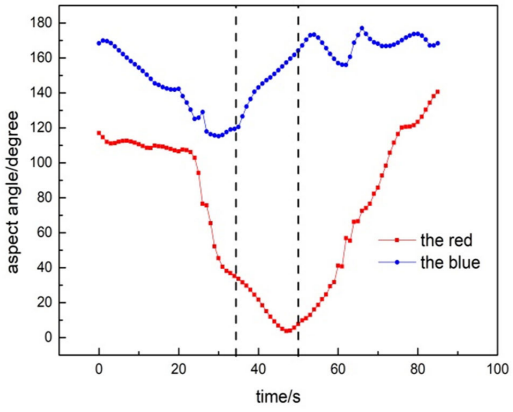


(c) Flight-path bank angle

FIGURE 8. History of controls for Head-on attack winning.



(a) Relative distance

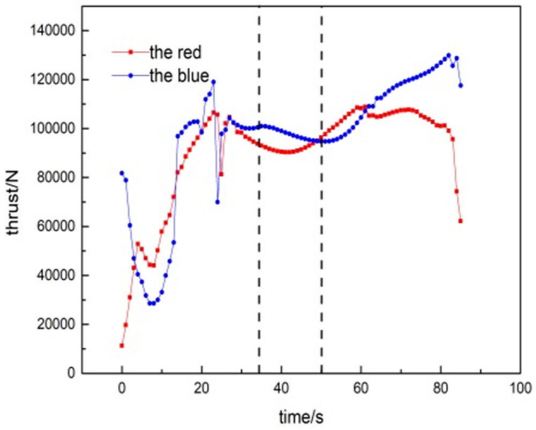


(b) Aspect angle

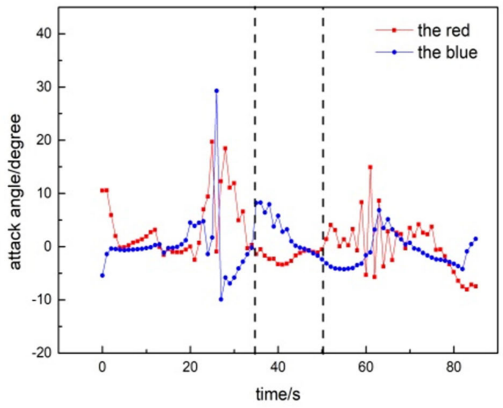
FIGURE 9. Situation relationship for Fleeing winning.

shown in Figure 7-Figure 8 with dotted black lines), the red continued to increase thrust, and approached to the blue in the attitude of swooping based on the stored height energy previously. During the period, the blue tried to alter flght-path bank angle to avoid, but beaten by the red at the time of 81 s. When Fleeing won, at the time of 35 s, the blue was cast into the area of aspect angle where the red allowed to strike, but the distance was out of allowed striking range, the red did not take strike action. Within the following 15 s (from 35 s to 50 s, trajectory shown in Figure 6 (b) with yellow lines and border shown in Figure 9-Figure 10 with dotted black lines), the blue made decisions to fly away the line of sight of the red. Concretely, the UAV of the blue was pulled by increasing the attack angle and flight-path bank angle rapidly, while the red maneuvered accordingly to keep the striking advantages. As a result, the blue fled in failure. Then the second attempt to flee for the blue was performed, the way of increasing thrust and swooping in negative attack angle was adopted. Consequently, the relative distance was enlarged rapidly and the aspect angle was changed to beneficial to avoid striking. Finally, the blue accomplished fleeing task at the time of 86 s.

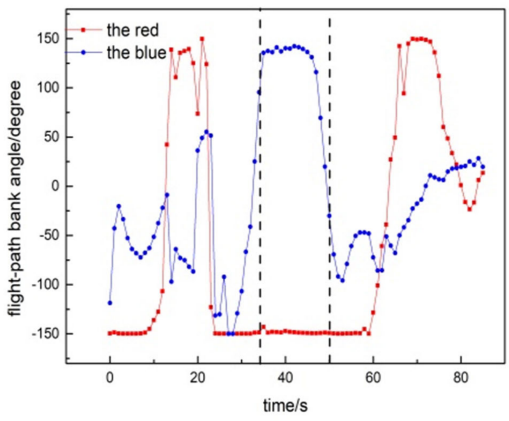
With the same model, the effect of self-learning was verified. For comparison, current neural network of the red was maintained while that of the blue trained further. A test would be executed per 5000 training episodes, where contained 200 air confrontation episodes in each test. The maximum step of each episode was set as 5000 and a draw would be recorded once exceeding the maximum value. For the blue, the results of air confrontation after self-learning were shown in Table 3. As we known, the initial situation was in favor of the winning of the red, so the number of episodes the red won was in majority all the time. However, the fleeing possibility of the blue was increased with the self-learning progressed,



(a) Thrust



(b) Attack angle



(c) Flight-path bank angle

FIGURE 10. History of controls for Fleeing winning.

TABLE 3. The results of air confrontation after self-learning.

Result	Self-learning episode of the blue			
	0	5000	10000	15000
Red win	162	148	145	142
Blue win	23	34	39	36
Draw	15	18	16	22

where the index of non-failure possibility was increased from 19% to 29%. As a whole, self-learning was a key to improve the degree of intelligent decision-making.

V. CONCLUSION AND FUTURE WORK

In this paper, application of DLR in air confrontation intelligent decision-making of MAV/UAV cooperative system is proposed. To highlight the leading influence of MAV and realize complementary advantages of MAV and UAV, four typical intentions containing Head-on attack, Fleeing, Pursuing and Energy-storing, are elaborately designed. Corresponding models are embedded into DDPG structure, which is an algorithm of DRL adapted for solving continues controls. Mechanism of experience judgment and improvement of sampling strategy are put forward to reconstruct the replay buffer. Several experiments are performed to test the proposed approach in different situations. Results show that the improved DDPG outperform the original in terms of convergence and stability of training. And the degree of intelligent decision-making can be improved by self-learning, which is of significance to achieve real intelligent air confrontation in future. Next, we will focus on the cooperative system of MAV with multiple UAVs, and the maneuver of high angle of attack and post-stall will also be researched.

REFERENCES

[1] C. Shen, L. Li, and Y. Wu, "Research on the capability of the U.S manned /unmanned autonomous collaborative operations," *Tactical Missile Technol.*, vol. 29, no. 1, pp. 16–21, Jan. 2018.

[2] C. Humphreys, R. Cobb, D. Jacques, and J. Reeger, "Optimal mission path for the uninhabited loyal Wingman," in *Proc. 16th AIAA/ISSMO Multidisciplinary Anal. Optim. Conf.*, Dallas, TX, USA, Jun. 2015, pp. 2792–2802.

[3] M. B. Reilly. (Apr. 6, 2018). *Beyond Video Games: New Artificial Intelligence Beats Tactical Experts in Combat Simulation. Cincinnati, USA*. Accessed: Mar. 12, 2020. [Online]. Available: http://magazine.uc.edu/editors_picks/recent_features/alpha.html

[4] G. I. Seffers. (May 1, 2016). *Commanding the Future Mission, Arkansas, USA*. Accessed: Mar. 12, 2020. [Online]. Available: https://www.afcea.org/content/Article-commanding-future-mission

[5] D. Hambling, "Robot pilot gets its wings and takes to the skies," *New Scientist*, vol. 243, no. 3246, p. 16, Sep. 2019.

[6] *Unmanned System Integrated Roadmap FY 2013-2038*. Accessed: Dec. 31, 2019. [Online]. Available: https://archive.defense.gov/pubs/DOD-USRM-2013.pdf

[7] M. Angelina. *Heavy and Rapid: 'The Hunter' for the First Time Rose Into the Sky*. Accessed: Dec. 31, 2019. [Online]. Available: https://www.gazeta.ru/army/2019/08/03/12555025.shtml

[8] Y. Wang, T. Sun, G. Rao, and D. Li, "Formation tracking in sparse airborne networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 2000–2014, Sep. 2018.

[9] X. Cao, P. Yang, M. Alzenad, X. Xi, D. Wu, and H. Yanikomeroglu, "Airborne communication networks: A survey," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 1907–1926, Sep. 2018.

[10] Q. Wu, J. Xu, and R. Zhang, "Capacity characterization of UAV-enabled two-user broadcast channel," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 1955–1971, Sep. 2018.

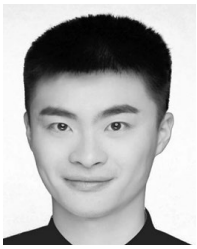
[11] Y. Cai, F. Cui, Q. Shi, M. Zhao, and G. Y. Li, "Dual-UAV-enabled secure communications: Joint trajectory design and user scheduling," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 1972–1985, Sep. 2018.

[12] C. H. Liu, Z. Chen, J. Tang, J. Xu, and C. Piao, "Energy-efficient UAV control for effective and fair communication coverage: A deep reinforcement learning approach," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 2059–2070, Sep. 2018.

[13] N. Veerasamy, "A high-level mapping of cyberterrorism to the OODA loop," in *Proc. 5th Eur. Conf. Inf. Manage. Eval.*, Sonning Common, U.K., 2011, pp. 352–360.

[14] H. Du, M. Cui, T. Han, and Z. Wei, "Maneuvering decision in air combat based on multi-objective optimization and reinforcement learning," *J. Beijing Univ. Aeronaut. Astronaut.*, vol. 44, no. 11, pp. 2247–2256, Nov. 2018.

[15] S. Zhou, W. Wu, N. Zhang, and J. Zhang, "Overview of autonomous air combat maneuver decision," *Aeronaut. Comput. Technique*, vol. 42, no. 1, pp. 27–31, Jan. 2012.
[16] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," 2013, *arXiv:1312.5602*. Accessed: Dec. 31, 2019. [Online]. Available: http://arxiv.org/abs/1312.5602
[17] T. P. Lillicrap, J. J. Hunt, and A. Pritzel, "Continuous control with deep reinforcement learning," in *Proc. Int. Conf. Learn. Represent.*, San Juan, Puerto Rico, 2016, pp. 1120–1139.
[18] 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, pp. 279–298, Nov. 2018.
[19] Y. Zhang, W. Zu, Y. Gao, and H. Chang, "Research on autonomous maneuvering decision of UCAV based on deep reinforcement learning," in *Proc. Chin. Control Decis. Conf. (CCDC)*, Shenyang, China, 2014, pp. 230–235.
[20] 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, 2020.
[21] Q. Cheng, X. Wang, J. Yang, and L. Shen, "Automated enemy avoidance of unmanned aerial vehicles based on reinforcement learning," *Appl. Sci.*, vol. 9, no. 4, pp. 166–187, Nov. 2019.
[22] B. Zhang, M. He, and X. Chen, "Self-driving via improved DDPG algorithm," *Comput. Eng. Appl.*, vol. 55, no. 10, pp. 264–270, Oct. 2019.
[23] J. Schulman, F. Wolski, and P. Dhariwal, *Proximal Policy Optimization Algorithms*. San Francisco, CA, USA: OpenAI, Aug. 2017. Accessed: Mar. 12, 2020. [Online]. Available: http://search.ebscohost.com/login.aspx?direct=true&db=edsarx&AN=edsarx.1707.06347&lang=zh-cn&site=eds-live
[24] N. Heess *et al. Emergence of Locomotion Behaviours in Rich Environments*. New York, NY, USA: DeepMind, Sep. 2017. Accessed: Mar. 12, 2020. [Online]. Available: http://search.ebscohost.com/login.aspx?direct=true&db=edsarx&AN=edsarx.1707.02286&lang=zh-cn&site=eds-live
[25] L. Tan, Q. Gong, and H. Wang, "Pursuit-evasion game algorithm based on deep reinforcement learning," *Aerosp. Control*, vol. 36, no. 6, pp. 3–8, Jun. 2018.
[26] C. Su, H. Zhao, Y. Wang, and H. Zhou, "UCAV autonomic maneuver decision-making method based on reinforcement learning," *Fire Control Command Control*, vol. 44, no. 4, pp. 142–149, Apr. 2019.
[27] X. Zong, G. Xu, G. Yu, H. Su, and C. Hu, "Obstacle avoidance for self-driving vehicle with reinforcement learning," *SAE Int. J. Passenger Cars-Electron. Electr. Syst.*, vol. 11, no. 1, pp. 30–39, 2017.
[28] H. Wei, "Research of UCAV air combat based on reinforcement learning," M.S. thesis, Dept. Electron, HIT Univ, Harbin, China, 2015.
[29] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*, 1st ed. Cambrige, MA, USA: MIT Press, 2017, pp. 66– 129.
[30] S. David, L. Guy, H. Nicolas, and D. Thomas, "Deterministic policy gradient algorithms," in *Proc. 31st Int. Conf. Mach. Learn.*, Beijing, China, 2014, pp. 76–84.
[31] Y. Gao, M. Yu, and Q. Han, "Air combat maneuver decision-making based on improved symbiotic organisms search algorithm," *J. Beijing Univ. Aeronaut. Astronaut.*, vol. 45, no. 3, pp. 429–436, Mar. 2019.
[32] C. Fang, "Research of the lane following decision-making of autonomous vehicle based on deep reinforcement learning," M.S. thesis, Dept. Electron, Nanjing University, Nanjing, China, 2019.
[33] P. Williams, "Real-time computation of optimal three-dimensional aircraft trajectories including terrain-following," in *Proc. AIAA Guid., Navigat., Control Conf. Exhibit*, Keystone, CO, USA, Aug. 2006, pp. 6603–6624.



YUE LI received the M.S. degree from NUDT, China, in 2016. He is currently pursuing the Ph.D. degree with Naval Aviation University, China. He is also working with Naval Aviation University. His research interests include control of manned/unmanned aerial vehicle systems, path planning for robots, optimal control, and related fields, such as process industry control and automation.

WEI HAN received the Ph.D. degree from the Nanjing University of Aeronautics and Astronautics, China, in 2003. He is currently a Professor with Naval Aviation University, China. His research interests include general area of the process industry control and automation and related fields, such as aerodynamics.

YONGQING WANG received the M.S. degree from the Beijing University of Aeronautics and Astronautics, China, in 2000. He is currently a Research Scientist with the Shenyang Aircraft Design and Research Institute, China. His research interests include general area of the process industry control and automation and related fields, such as aerodynamics.