Intention prediction of UAVs based on improved DDQN

Tianpei Chen¹,²,³, Haotian Liu¹ and Yuhui Wang¹*

¹ College of Automation Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu Province, 211106, China
² Shanghai Institute of Spaceflight Control Technology, Shanghai 201109, China
³ Shanghai key Laboratory of Aerospace Intelligent Control Technology, Shanghai 201109, China

*Corresponding author’s e-mail: wangyh@nuaa.edu.cn

Abstract. Intention prediction plays an indispensable role in future informationized air combat, which will help the command and control center to make a more correct decision. In this paper, an intelligent intention prediction approach based on an improved double deep Q network (DDQN) is developed to generate real-time intention flight paths for unmanned aerial vehicles (UAVs) in complex air combat environment. Initially, by introducing an actual topographic map, the different threats of UAVs in different terrains on the map are analyzed, based on which, a terrain environment reward function is constructed. Secondly, by splitting a complete maneuver action into six basic maneuver units and determining the probability value of each unit, a maneuver reward function is obtained. Further, to improve the real-time performance and accuracy of the standard DDQN algorithm, an improved DDQN algorithm is proposed by using temporal-difference (TD) method and binary tree data structure. Finally, the simulation results verify show that the proposed algorithm has achieved better results under complex terrain conditions.

1. Introduction
In recent years, the role of UAVs in the military field is increasingly prominent. Compared with manned aerial vehicles, UAVs, characterized by a lack of human casualties, high mobility, light weight and low cost, have become important units of modern air combat[1]-[3]. Due to the rapid changing battlefield situation of air combat, if the next maneuvering flight intention for the opponent's UAVs can be predicted according to the before and current maneuvering actions, undoubtedly, it will help to achieve the confrontation advantage.

In recent years, intention prediction has drawn much attention in the field of the intelligent decision control for UAVs[4]. Most of the research results are mainly determine the intention of attack or defense based on the parameters and maneuverability of UAVs. Then, by using some intention prediction methods, such as Bayesian networks(BN)[11]-[13], lanchester equation[14], probabilistic neural networks (PNN)[15], infrared optical image of the target aircraft, and the image moment invariance[16], the paths of UAVs within a period can be obtained, but these results usually cannot directly used for some complex topographic maps.

Since traditional prediction methods have some difficulties in dealing with intent prediction in complex situations, especially for future intelligent air combat[17]-[18], it is necessary to explore a new effective method solve this problem. Deep reinforcement learning, as a newly proposed intelligent algorithm, has some unique advantages in processing intention prediction problems: good continuity,
high efficiency, and simple process. In [19] - [22], double deep Q network (DDQN) for intention prediction was considered and achieved satisfactory results. Compared with DQN algorithm, DDQN introduces a second layer Q network, so that the output of the first layer is no longer used as the output result, but as the input of the second layer, thus it avoids the overestimation problem and improves the accuracy of DQN algorithm.

However, due to the complex flight environment and flight characteristics of UAVs, there are few studies based on DDQN to predict the intention for UAVs. The key points is that the flight environment and flight characteristics must be modeled first when applying DDQN to UAVs. Therefore, in this paper, by establishing the terrain map, maneuvering library and their reward functions, DDQN can be considered for solving the intention prediction problems of UAVs.

Furthermore, to improve the slow calculation speed and low transmission efficiency of the standard DDQN algorithm, Temporal-Difference (TD) error[23] and binary tree data storage structure[24] are considered here, where TD method is introduced to evaluate the effect of the current iteration result. The quality of the selected action in the current iteration is inversely proportional to the TD error calculation result, according to the evaluation results, a more appropriate flight maneuver action can be selected. Next, the TD error result will be stored as the historical information, furtherly, to reduce the time spent in selecting historical information, a binary tree data structure is used to replace the experience pool data structure, and the results obtained by the TD method in each iteration can be stored in the leaf nodes of the binary tree.

In addition, to make the intention prediction result more intuitive, the intention prediction maneuvers trained by the DDQN algorithm are connected to obtain a flight path. Compared with other applications, there are more factors that need to be considered in the UAV's intention prediction, including terrain threats, fire threats, and maneuverability.

As discussed above, the main contributions of this paper are given as follows:

I. A new intention prediction scheme for UAVs is proposed by establishing the terrain map, maneuvering library and their reward functions.

II. An improved DDQN algorithm is proposed by using temporal-difference (TD) method and binary tree data structure, to increase the real-time performance and accuracy of the standard DDQN algorithm.

The structure of this paper is as follows. Section 2 establishes the actual terrain environment maps, and gives the terrain's reward functions and the maneuver reward functions. Then, the reward functions based on the improved DDQN algorithm for intention prediction are trained in Section 3. Section 4 shows the simulation results of two different terrains. The conclusions of the paper are given in Section 5.

2. Problem description and modelling
The intention prediction process of a UAV can be regarded as the process of continuous interaction between the UAV and the environment. Each interaction corresponds to a reward value, and the best maneuver can be selected according to the iterative result of the reward value. Figure 1 is a schematic diagram of the interaction between the UAV and the terrain environment, where $S_i$ represents the UAV position of the $t$ th iteration, $r_i$ is the reward function value of the $t$ th iteration, and $A_i$ is the action taken by the UAV of the $t$ th iteration.

Figure 1. Diagram of the interaction between the UAV and the environment.
From Figure 1, it can be seen that, when the UAV performs the action \( A_t \), it interacts with the current environment to generate a state position \( S_t \), and the environment gives a reward value \( r_t \) to reflect the performance of the action \( A_t \). Then, the action \( A_{t+1} \) can be selected according to \( A_t, S_t \) and \( r_t \), and will be performed at the next iteration \( t+1 \). In the iteration loop, the interaction between the UAV and the environment will continuously generate new information, based on which, the action strategy will be modified accordingly. After enough iterations, the optimal action strategy needed to complete the task will be obtained.

After the UAV performs each action, the decision-making system will evaluate the action by using a reward function. When the reward value is positive, it means that the action is recommended to be selected at the current moment. When the reward value is negative, it means that the action is not recommended to be selected at the current moment. Therefore, the reward function can affect the UAV’s choice of actions, which in turn affects the result of intention prediction.

In this paper, two reward functions are considered regarding intention prediction: the terrain reward function and the maneuver reward function. Then, the total reward function can be expressed as the weighted sum of the two reward functions as follows:

\[
r_t = \lambda_d r_d + \lambda_m r_m
\]

where \( r_d \) is the terrain reward function, \( r_m \) is the maneuver reward function, \( \lambda_d \) and \( \lambda_m \) are the corresponding weight coefficients.

2.1. The terrain reward

Usually, intention prediction is mainly carried out in a virtual three-dimensional (3D) map. Different from the previous research\(^\text{[11]-[16]}\), this paper applies the actual terrain environment to consider the effect of terrain environment factors to the UAV, which increase the practicability of the proposed method.

In actual combat, UAVs should avoid dangerous terrain environments(such as high-altitude mountainous areas), and fight in safe terrain environments (such as low-altitude flat areas). Other areas can be considered as general combat areas, and there is also a risk of collision, but the risks are much smaller than the dangerous combat areas.

According to the above analysis, the environment reward \( r_i \) is described as follows:

\[
r_i = \begin{cases} 
-1, & D \subseteq D_d \\
1, & D \subseteq D_s \\
0, & D \subseteq D_c 
\end{cases}
\]

(2)

where \( D \) means the area of the UAV located, \( D_d \) is the dangerous combat area, and \( D_s \) is the safe combat area.

In addition to ensure the flight safety of the UAV, the anti-collision reward \( r_j \) is described as follows:

\[
r_j = \begin{cases} 
-1, & d < d_{\text{min}} \\
1, & d > d_{\text{max}} \\
0, & d_{\text{min}} \leq d \leq d_{\text{max}} 
\end{cases}
\]

(3)

where \( d \) represents the distance between the UAV and the ground, \( d_{\text{min}} \) is the minimum safe distance, and \( d_{\text{max}} \) is the maximum safe distance.

Then, the terrain reward \( r_d \) is a weighted sum of \( r_i \) and \( r_j \):

\[
r_d = \phi_1 r_i + \phi_2 r_j
\]

(4)

where \( \phi_1 \) and \( \phi_2 \) are the weight coefficients.
2.2. The maneuver reward

This section mainly analyzes the maneuver of the UAV, and then provides the required maneuver reward function. UAV combat maneuvers can be split into basic maneuver units, including direct flight, flip, jump, left jump, right jump, left turning, right turning, dive, left dive, and right dive. By combining these basic maneuver units in different orders, most complex maneuvers can be described. Table 1 characterizes the correspondence relationship between the complex maneuvers and the basic maneuver units.

From Table 1, it can be seen that the left(right) jump and the left(right) dive are usually used together to track targets. Therefore, to simplify the model, these four basic maneuver units are collectively referred to as track. Similarly, left and right turning are collectively referred to as turning. Then the basic maneuvering units are simplified into six units, which are direct flight, jump, flip, dive, turning, and track. The complex maneuvers can be described by combining the above six basic maneuver units. The detailed connection rule is shown as Figure 2.

Denoting that the complex maneuver of the UAV is \( MR \), the relationship between \( MR \) and the basic maneuver unit \( Mr_i \) is shown as follows:

\[
MR = f \left( (Mr_i, \tau_i), (Mr_j, \tau_j), \ldots, (Mr_n, \tau_n) \right)
\]

(5)

where \( Mr_i(i \in \text{direct flight, jump, flip, dive, turning, track}) \) represents the six basic maneuver units in Figure 2, and \( f(\cdot) \) means the sequence of performing maneuver units. The performing time of \( MR \) is \( \tau \), which is divided into \( n \) intervals: the basic maneuver unit at time \( \tau_i \) is performed first, then the basic maneuver unit at time \( \tau_2 \) is performed, until the basic maneuver unit at time \( \tau_n \) is performed, it indicates that \( t \) the complex maneuver \( MR \) performed completely. \( \tau_1, \tau_2, \ldots, \tau_n \) are the time required for the UAV to complete \( Mr_i \) respectively, and satisfy the following relationships:

\[
t_{\text{min}} \leq \tau = \tau_1 + \tau_2 + \ldots + \tau_n \leq t_{\text{max}}
\]

(6)

where \( t_{\text{min}} \) and \( t_{\text{max}} \) are the shortest time and longest time required for the UAV to complete \( MR \).

| Common maneuver     | Basic maneuver unit                     |
|---------------------|----------------------------------------|
| Somersault          | direct flight + jump + direct flight + dive |
| Half somersault flip| jump + flip + direct flight             |
| High yoyo           | direct flight + left(right) jump + left(right)dive |
| Low yoyo            | direct flight + left(right) dive + left(right)jump |
| Roller              | direct flight + turning + jump + dive + jump |
| Broken S            | direct flight + flip + dive + direct flight |
| Innemman's maneuver | direct flight + dive + jump             |
Figure 2. Connection diagram of basic maneuver units

| Table 2. Initial probability of selecting basic maneuver units |
|--------------------------------------------------------------|
| $A^{t+1}$ | direct flight | jump | flip | dive | turning | track |
| direct flight | 0.30 | 0.25 | 0.10 | 0.15 | 0.10 | 0.10 |
| jump | 0.20 | 0.10 | 0 | 0.40 | 0.05 | 0.25 |
| flip | 0.15 | 0 | 0.25 | 0.60 | 0 | 0 |
| dive | 0.25 | 0.30 | 0 | 0.10 | 0.05 | 0.30 |
| turning | 0.20 | 0.05 | 0 | 0.05 | 0.20 | 0.50 |
| track | 0 | 0.45 | 0 | 0.45 | 0 | 0.10 |

In summary, a complex maneuver $MR$ is split into a sequence of basic maneuver units $Mr_i$ combinations. According to Figure 2, and considering the actual flight, the reward function $r_m$ can be obtained as follows:

$$r_m = R^{t+1} + \tau_i$$

where $\tau_i$ ($i \in \text{direct flight, jump, flip, dive, turning, track, round, track}$) is a continuous selection probability, and $R^{t+1}$ is the initial probability of selecting basic maneuver units from the $t$th iteration to the $(t+1)$th iteration. The initial probability $R^{t+1}$ is shown in Table 2.

It should be noted that, the probabilities in Table 2 are only initial values, and the probabilities of continuously performing maneuvers need to consider a continuous selection probability $\tau_i$. For example, if the selected basic maneuver unit in the $t$th iteration is the direct flight unit, and the selected unit in the $(t+1)$th iteration is still the direct flight unit, then the probability of selecting direct flight in the $(t+2)$th iteration is $0.3 + \tau_{\text{direct flight}}$, where $\tau_{\text{direct flight}}$ is the continuous selection probability of the direct flight unit.

Similarly, the continuous selection probabilities of the jump, flip, dive, turning and track units are $\tau_{\text{jump}}, \tau_{\text{flip}}, \tau_{\text{dive}}, \tau_{\text{turning}}, \text{ and } \tau_{\text{track}}$. The values of the continuous selection probabilities of the units will be given in Section 4.
3. Improved DDQN algorithm

Compared with other intelligent algorithms, deep reinforcement learning algorithm has outstanding advantages in solving intent prediction problems. In this section, the standard DDQN algorithm is improved to avoid its slow calculation speed and low transmission rate, and combined with the reward function model in Section 2, the improved algorithm is applied to predict the intention of the UAV.

Figure 3 is a block diagram of strategy action selection based on deep reinforcement learning, where the strategy $\pi_t$ is the basis for the UAV to choose actions.

Figure 3. Block diagram of strategic action selection based on deep reinforcement learning

In Figure 3, $i \in \{\text{direct flight}, \text{jump}, \text{flip}, \text{dive}, \text{turning track}\}$, $A_j$ is the action $i$ in the $t$th iteration, $S_{A_j}$ is the state position of the UAV after performing the action $A_j$ in the $t$th iteration. The selected action $A_{t+1,j}$, state position $S_{A_{t+1,j}}$, and reward function $r_{t+1}$ in the $(t+1)$th iteration can be calculated through the strategy $\pi_t$ and the information in the $t$th iteration.

The process of calculating $\pi_{t+1}$ needs to introduce the action value function $\hat{q}$, which is determined by the neural network parameter $\theta$, and the feature vector, the expression is:

$$\hat{q}(A_{t+1,j}) = f(\theta, \Phi(S_{A_j}))$$

(8)

Figure 4 shows the approximate process of the value function $\hat{q}(A_{t+1,j})$. By taking $S_{A_j}$ and $A_j$ as the inputs to obtain the feature vector $\Phi(S_{A_j})$, the value $\hat{q}(A_{t+1,j})$ of the action $A_{t+1,j}$ can be calculated through the neural network parameter $\theta$.

Figure 4. Approximation process of the action value function

The value function $\hat{q}(A_{t+1,j})$ is an approximation value, and the actual one needs to be selected by using the strategy $\pi_{t+1}$.

The design idea of the strategy $\pi_{t+1}$ is to increase the number of samples in the early stage of training, and in the later stage with enough samples, by choosing the maximum value function to ensure that the algorithm converges quickly. Thus, $\pi_{t+1}$ can be described as:

$$\pi_{t+1} = \left(1 - \epsilon\right) \cdot \pi_{t+1,\text{best}} \land \epsilon \cdot \pi_{t+1,\text{other}}$$

(9)
where $\pi_{t+1,\text{best}} = \{A_{t+1,i} | \hat{q}(A_{t+1,i}) = \hat{q}_{\text{max}}\}$, $(i \in \text{direct flight, jump, flip, dive, turning, track})$ represents the strategy of selecting the action with the maximum value, $\pi_{t+1,\text{other}} = \{A_{t+1,k} | \hat{q}(A_{t+1,k}) \neq \hat{q}_{\text{max}}\}$, $(k \in \text{direct flight, jump, flip, dive, turning, track}; k \neq i)$ represents the strategy of selecting the other actions except the action with the maximum value.

In addition, to achieve $\pi_{t+1}$, it is necessary to introduce a greedy value $\varepsilon \in (0,1)$ to perform the selection of $\pi_{t,\text{max}}$ or $\pi_{t,\text{other}}$. Assuming $\varepsilon=0.8$, the probability of performing $\pi_{t,\text{max}}$ is $1-\varepsilon=0.2$, and the probability of performing $\pi_{t,\text{other}}$ is $\varepsilon=0.8$. The initial value of the greedy value $\varepsilon$ is given by the designer, and will continue to decrease with the iteration.

Due to that the action value function $\hat{q}(A_{t+1,i})$ is calculated through the parameter $\theta_t$ of the Q network, Therefore, it is necessary to update $\theta_t$ for action selection.

DDQN has two layers Q network structure to avoid the overestimation problem of DQN, where the first layer Q network is used to find the action with the maximum value function, and the second layer Q network is used to calculate the Q value of the selected action. The selected action can be calculated by using the first layer Q network:

$$A_{t,\text{choose}} = \arg \max_{A_t} Q^1(\Phi(S_{t,i}^1), A_t, \theta^1_t)$$

(10)

where $Q^1$ is the first layer network, $\theta^1_t$ is the parameter of $Q^1$, $S_{t,i}^1$ represents the state position of $Q^1$ in the $t$th iteration, $\Phi(S_{t,i}^1)$ is the feature vector of $Q^1$ in the $t$th iteration.

The second layer Q network is used to calculate the action value function output value $y_t$ of the action $A_{t,\text{choose}}$:

$$y_t = r_t + \gamma Q^2(\Phi(S_{t,i}^2), A_{t,\text{choose}}, \theta^2_t)$$

(11)

where $Q^2$ is the second layer network, $\theta^2_t$ is the parameter of $Q^2$ assigned by $\theta^1_t$, $\Phi(S_{t,i}^2)$ is the feature vector of $Q^2$ in the $t$th iteration.

In this section, the standard DDQN algorithm is improved to avoid its slow calculation speed and low transmission rate by using TD method and binary tree structure.

In the following, TD method is introduced to evaluate the action of the current iteration by comparing with the next iteration. If the TD error is large, it indicates that the current selection is inappropriate and should be selected again. The calculation formula is:

$$d_t = y_t - Q^2(\Phi(S_{t,i}^2), A_{t-1,i}, \theta^2_{t-1})$$

(12)

where $d_t$ is the TD error.

In addition, to reduce the time spent in selecting historical information, the data structure of a binary tree is considered to replace the data structure of the experience pool, as shown in Figure 5. The results obtained by using the TD method are stored in the leaf nodes of the binary tree, and the node of the binary tree is directly proportional to the size of the TD error result $d_t$. 


Based on TD method and binary tree structure, historical information with large errors can be efficiently selected, which improves the calculation speed and accuracy of the standard algorithm. The value $p_i$ of each binary leaf node depends on the size of $d_i$ as follows:

$$ p_i = |d_i| + \mu $$

(13)

where $\mu$ is a small number to avoid $p_i$ being 0, $t \in 1, 2, ..., j$, and $j$ is the capacity of the binary tree.

The selection principle of the priority sampling theorem is: the larger value of a leaf node in the binary tree, the higher the priority and the greater the probability of being selected. By this method, the TD error result $d_i$ can be selected quickly and efficiently, thereby reducing the running time of the algorithm. The priority sampling $P_j(t)$ can be calculated:

$$ P_j(t) = p_i / \left( \sum_{i=1}^{j} p_i \right) $$

(14)

Sampling through priority sampling may lead to inhomogeneous distribution of algorithm results, resulting in deviations. In order to reduce the risk of premature convergence and improve the stability of the sampling process, the method of importance sample weights (ISW) is introduced. The goal of this method is to estimate the nature of a certain distribution, and reduces the variance by adjusting the probability distribution. The calculation formula is:

$$ \omega_j(t) = 1 / \left( j \cdot P_j(t) \right) $$

(15)

Next step, the neural network parameter $\hat{\theta}_t$ is updated through back propagation:

$$ \hat{\theta}_t = \theta_t^2 + \omega_t(t) \cdot (d_t)^2 $$

(16)

where $\hat{\theta}_t$ is the Q network parameter in the $t$ th iteration.

After the neural network parameters are obtained, the approximate process of the action value function can be used to calculate the strategy $\pi_{t+1}$. If the next iteration is not the end state position, the action $A_{t+1}$ at the next iteration can be determined according to Eq (9). The algorithm loops until it reaches the final state, and the algorithm flow chart is shown in Figure 6.
4. Simulation

This section is based on the improved deep reinforcement learning algorithm for simulation. Table 3 shows the required parameters in the algorithm training. Table 4 is the consecutive selection probability values of the six basic units.

In order to verify the effectiveness of the proposed algorithm, two terrain maps (terrain A and terrain B) are considered here (see Figures. 7-10), and the start and end points of the flight track are set respectively based on the maps, where the blue dot is the starting point and the green dot is the ending point. And in order to further verify the practicability of the proposed algorithm, short-distance and long-distance intention predictions on two different terrains are simulated. The difference between short and long-distance path is the distance between the start and end points of the intention predictions, set threshold is 20km, the distance less than this value is a short-distance path, and the value greater than this value is a long-distance path. In addition, since the actual topographic map is more complicated, two-dimensional (2D) maps are considered here. To make up for the deficiencies of the 2D maps, in the simulations the real-time height change curves of the UAV are provided.

Figure 6. Flow chart of the improved deep reinforcement learning algorithm

Table 3. Parameters of the training

| Parameters                  | Value  | Description                                           |
|-----------------------------|--------|-------------------------------------------------------|
| Number of iterations \( N \) | 2000   | Number of training cases over which each stochastic gradient descent update is computed |
| \( \epsilon_{\text{init}} \) | 1      | Initial value of \( \epsilon \) in \( \epsilon \)-greedy exploration |
| \( \epsilon_{\text{final}} \) | 0.01   | Final value of \( \epsilon \) in \( \epsilon \)-greedy exploration |
| Number of binary tree nodes \( n \) | 10000 | The number of binary tree nodes represents the upper limit of data that can be stored |
| Attenuation factor \( \gamma \) | 0.9    | Constrain the value of the dynamic value function calculated by the gradient backpropagation method |
| \( A_{s,0} \)                 | Direct | Initial state value                                     |
Table 4. Consecutive selection probability values of basic maneuver units

| Symbol       | $\tau_{\text{direct flight}}$ | $\tau_{\text{jump}}$ | $\tau_{\text{flip}}$ | $\tau_{\text{dive}}$ | $\tau_{\text{turning}}$ | $\tau_{\text{track}}$ |
|--------------|--------------------------------|----------------------|-----------------------|------------------------|--------------------------|------------------------|
| Value        | 0.1                            | 0.1                  | 0.025                 | 0.15                   | 0.025                    | 0.1                    |

For terrain A, short-distance and long-distance paths based on improved DDQN algorithm are shown in Figure 7(a) and Figure 8(a), and the height change curves are shown in Figure 7 (b) and Figure 8 (b).

![Figure 7](image1.png)  
**Figure 7.** Short-distance path based on improved DDQN algorithm on terrain A

From Figure 7, it can be seen that when $t = 50 \sim 180s$, the UAV has performed four basic maneuvers: direct flight, turning, jump, and dive, based on this, it can be inferred that the UAV may have performed a roller maneuver during the period. When $t = 260 \sim 330s$, the UAV performed the three basic maneuver units of flip, dive, and direct flight, and it can be inferred that the UAV has performed the broken S maneuver during the period.

![Figure 8](image2.png)  
**Figure 8.** Long-distance path based on improved DDQN algorithm on terrain A

From Figure 8, it can be seen when $t = 100 \sim 350s$, the UAV performed three basic maneuvers of direct flight, jump and dive, based on this, it can be inferred that the UAV has performed a Low yoyo maneuver during the period.

In order to reflect the advantages of the improved DDQN algorithm, Figure 9 and Figure 10 respectively show the short-distance intent prediction path based on the improved algorithm and the standard algorithm on terrain B.
Figure 9. Short-distance path based on improved DDQN algorithm on terrain B

Figure 10. Short-distance path based on traditional DDQN algorithm on terrain B

It can be seen from Figure 9 and Figure 10 that the improved DDQN algorithm can predict a more suitable and shorter flight path. It can be seen from Table 5, that the improved DDQN algorithm has higher simulation success rate than traditional algorithms, and the simulation time is shorter.

Table 5. Comparison table of the time used by the two algorithms

|                          | Traditional DDQN | Improved DDQN |
|--------------------------|------------------|---------------|
| Predict the average time required for a maneuver action unit | 4.09s            | 2.73s         |
| Total time spent for predicting complete intention         | 441.72s          | 248.43s       |
| Simulation success probability                              | 90%              | 100%          |
| Reselection times of basic maneuvering unit                 | 1526             | 875           |

It can be seen from Table 5 that the improved DDQN algorithm has shorter simulation time, because the binary tree structure used in the improved DDQN algorithm can save a lot of time in the selection of effective data and iterative calculations. And the simulation success probability of the improved DDQN algorithm is higher than that of the traditional one, and the number of reselections of basic maneuver units is significantly less than that of the traditional algorithm.

From the above simulation results, it can be seen that compared with the standard algorithm, the improved algorithm can predict the intention under complex terrain conditions and display the path based on the intention prediction. And, the real-time movement of the UAV can be determined in combination with the height change.

5. Conclusions

This paper improves the DDQN algorithm for deep reinforcement learning and applies it to the intention prediction of UAVs. In the simulation, the terrain reward function and the maneuver reward function are considered. After the DDQN algorithm training, the maneuver intention of the UAV is
predicted, and the path is also obtained based on the intention prediction result. From the simulation results, we can see that the improved DDQN algorithm can predict the intention of UAVs with faster speed and higher success probability, which provides an important technical support for future UAV air combat.

Acknowledgement
This work was supported by the Foundation of the Key Laboratory for Equipment (6142504200106), National Natural Science Foundation of China(61773204), Major Projects for Science and Technology Innovation 2030 (Grant No. 2018AA0100800).

References
[1] V Jia N, Yang Z, Yang K. Operational Effectiveness Evaluation of the Swarming UAVs Combat System Based on a System Dynamics Model[J]. IEEE Access, 2019, 7(9): 25209-25224.
[2] Chen H X, Nan Y, Yang Y. A Two-Stage Method for UCAV TF/TA Path Planning Based on Approximate Dynamic Programming[J]. Mathematical Problems in Engineering, 2018, 10(11):1-11.
[3] You S, Gao L, Diao M. Real-Time Path Planning Based on the Situation Space of UCAVS in a Dynamic Environment[J]. Microgravity Science and Technology, 2018, 30(6): 899-910.
[4] Boubeta-Puig J, Moguel E, Sánchez-Figueroa F, et al. An Autonomous UAV Architecture for Remote Sensing and Intelligent Decision-Making[J]. IEEE Internet Computing, 2018, 22(3): 6-15.
[5] Guan X, Lv R, Sun L, et al. A Study of 4D Trajectory Prediction Based on Machine Deep Learning[C]/ 12th World Congress on Intelligent Control and Automation (WCICA). IEEE, 2016: 24-27.
[6] Guan X, Zhang X, Han D, et al. A Strategic Flight Conflict Avoidance Approach Based on a Memetic Algorithm[J]. Chinese Journal of Aeronautics, 2014, 27(1): 93-101.
[7] Bühlmann P, Hothorn T. Boosting Algorithms: Regularization, Prediction and Model Fitting[J]. Statistical Science, 2007, 22(4): 477-505.
[8] Guan X, Zhang X, Wei J, et al. A strategic Conflict Avoidance Approach Based on Cooperative Coevolutionary with the Dynamic Grouping Strategy[J]. International Journal of Systems Science, 2016, 47(9): 1995-2008.
[9] Du W, Gao Y, Liu C, et al. Adequate is Better: Particle Swarm Optimization with Limited-Information[J]. Applied Mathematics and Computation, 2015, 268: 832-838.
[10] Gao Y, Du W, Yan G. Selectively-Informed Particle Swarm Optimization[J]. Scientific Reports, 2015, 5: 9295-9306.
[11] Wang F, Mei W, Wang C, et al. Maneuvering Mode Prediction of Aerial Target Based on Tactical Intention[J]. Electronics Optics & Control, 2009, 2(1):1-21.
[12] Schulz J, Hubmann C, Löchner J, et al. Multiple Model Unscented Kalman Filtering in Dynamic Bayesian Networks for Intention Estimation and Trajectory Prediction[C]/ 21st International Conference on Intelligent Transportation Systems (ITSC). IEEE, 2018: 1467-1474.
[13] Zu W, Gao Y, Chang H, et al. A UAV Formation Maneuvering Decision Algorithm Based on Heuristic Tactics Knowledge[C]/2017 29th Chinese Control and Decision Conference (CCDC). IEEE, 2017: 7280-7284.
[14] Huang J C, Zhou D. Design of Lanchester Equation and Prediction of Operational Process for Cooperative Operation of UAVs[J]. Electronics Optics & Control, 2018, 5(12): 40-44.
[15] Mao M Y, Zhang A, et al. Reinforcement Learning of UCAV Air Combat Based on Maneuver Prediction[J]. Electronics Optics & Control, 2019, 26(2): 5-10, 22.
[16] Dong Y, Huang J, Ai J. Visual Perception-Based Target Aircraft Movement Prediction for Autonomous Air Combat[J]. Journal of Aircraft, 2014, 52(2): 538-552. IEEE, 2017: 7280-7284.
[17] Van Hasselt H, Guez A, Silver D. Deep Reinforcement Learning with Double Q-Learning[C]//The thirtieth AAAI Conference on Artificial Intelligence. 2016: 1-20.
[18] Wu K, Wang H, Esfahani M A, et al. BND*-DDQN: Learn to Steer Autonomously through Deep Reinforcement Learning[J]. IEEE Transactions on Cognitive and Developmental Systems, 2019, 7(11): 1-11.
[19] Muñoz G, Barrado C, Çetin E, et al. Deep Reinforcement Learning for Drone Delivery[J]. Drones, 2019, 3(3): 72-82.
[20] Xu T, Feng Z H, Wu X J, et al. Joint Group Feature Selection and Discriminative Filter Learning for Robust Visual Object Tracking[C]//Proceedings of the IEEE International Conference on Computer Vision. 2019: 7950-7960.
[21] Zhang D, Yao L, Wang S, et al. Fuzzy Integral Optimization with Deep Q-Network for EEG-Based Intention Recognition[C]//Pacific Asia Conference on Knowledge Discovery and Data Mining. Springer, Cham, 2018: 156-168.
[22] Mnih V, Kavukcuoglu K, Silver D, et al. Human-level Control through Deep Reinforcement Learning[J]. Nature, 2015, 518(7540): 529-533.
[23] Anschel O, Baram N, Shimkin N. Averaged-DQN: Variance Reduction and Stabilization for Deep Reinforcement Learning[C]//International Conference on Machine Learning. PMLR, 2017: 176-185.
[24] Li Y, He H, Peng J, et al. Energy Management Strategy for a Series Hybrid Electric Vehicle Using Improved Deep Q-network Learning Algorithm with Prioritized Replay[J]. DEStech Transactions on Environment, Energy and Earth Sciences, 2018,978(1):1-6.