UAV maneuvering decision-making algorithm based on Twin Delayed Deep Deterministic Policy Gradient Algorithm

BAI Shuangxia¹, SONG Shaomei², LIANG Shiyang³, WANG Jianmei¹, LI Bo¹, Neretin Evgeny⁴

1 School of Electronics and Information, Northwestern Polytechnical University, Xi’an 710072, China;
2 Beijing Electro-Mechanical Engineering Institute, Beijing 100074, China;
3 Avic Luoyang Electro-optical Equipment Research Institute, Luoyang 471000, China
4 School of Robotic and Intelligent Systems, Moscow Aviation Institute, Moscow 125993, Russia;

* LI Bo: libo803@nwpu.edu.cn

Received Month X, XXXX | Accepted Month X, XXXX | Posted Online Month X, XXXX

Aiming at intelligent decision-making of UAV based on situation information in air combat, a novel maneuvering decision method based on deep reinforcement learning is proposed in this paper. The autonomous maneuvering model of UAV is established by Markov Decision Process. The Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm and the Deep Deterministic Policy Gradient (DDPG) algorithm in deep reinforcement learning are used to train the model, and the experimental results of the two algorithms are analyzed and compared. The simulation experiment results show that compared with the DDPG algorithm, the TD3 algorithm has stronger decision-making performance and faster convergence speed, and is more suitable for solving combat problems. The algorithm proposed in this paper enables UAVs to autonomously make maneuvering decisions based on situation information such as position, speed, and relative azimuth, adjust their actions to approach and successfully strike the enemy, providing a new method for UAVs to make intelligent maneuvering decisions during air combat.

Keywords: air combat, maneuvering decision-making, TD3, DDPG

I. Introduction
At present, unmanned aerial vehicles (UAVs) are widely used in military applications such as reconnaissance, attack, and jamming. Due to the complexity and variability of a battlefield, future UAVs need to be capable of undertaking autonomous operations. Therefore, maneuvering decision-making algorithm of UAV in a combat process of modern air war has become popular research subject⁵. The traditional rule evolution UAV maneuvering decision-making method based on genetic algorithm or genetic fuzzy system relies on human prior knowledge, and significantly lacks self-learning ability.

As an important paradigm in artificial
intelligence, deep reinforcement learning has shown great advantages in solving various problems, and has emerged in many applications in the field of UAV air combat maneuvering decision as well. Part of the research combines deep reinforcement learning with traditional methods to make UAV maneuvering decisions, such as Game Theory, Particle Swarm Optimization and so on. However, traditional methods such as Game Theory need to establish a clear and complete problem model. In another part of the research, the UAV maneuvering decision-making is realized by deep reinforcement learning, the autonomous maneuvering model of UAV is established by Markov Decision Process, and the decision function is fitted by neural network. Through training, UAV can master the optimal behavior strategy by the interaction with the environment. However, the existing research of UAV intelligent maneuvering decision based on deep reinforcement learning still has the following shortcomings: (1) The simulation environment is mainly two-dimensional space, so it lacks high-level exploration and analysis; and (2) it doesn’t consider the impact of radar and weapon on air combat, and therefore, it is difficult to apply to a complex battlefield environment.

In response to the above problems, this paper establishes a three-dimensional (3D) UAV air combat model, and a UAV maneuvering decision algorithm based on deep reinforcement learning is proposed. The remainder of this paper is organised as follows. In Section II, a UAV air combat model based on the characteristics of the 3D environment is defined. In Section III, intelligent UAV maneuvering decision-making algorithm based on deep reinforcement learning is proposed. In section IV, the simulation results demonstrate the effectiveness of the proposed algorithm in the field of air combat decision, and the simulation results of the Deep Deterministic Policy Gradient (DDPG) algorithm and Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm are compared. Finally, conclusions are presented in Section V.

II. UAV air combat model
The following assumptions are made for the establishment of the UAV motion and dynamics model:
- Assume that the UAV is a rigid body;
- Ignore the influence of the earth's rotation and revolution, and ignore the earth's curvature; and
- Due to the large range of maneuverability and short combat time in close air combat, the impact of fuel consumption on quality and the effect of wind are ignored

In a 3D space, UAV has physical descriptions such as position, speed, and attitude. The 3D space coordinate system where UAV is located is defined as $OXYZ$, the positive direction of the $X$ axis is north, the positive direction of the $Z$ axis is east, and the positive direction of $Y$ axis is vertical up.

\[
\begin{aligned}
\frac{dX}{dt} &= v \cdot \cos \theta \cdot \cos \phi \\
\frac{dY}{dt} &= v \cdot \sin \theta \\
\frac{dZ}{dt} &= v \cdot \cos \theta \cdot \sin \phi \\
\frac{dv}{dt} &= a \\
\frac{d\theta}{dt} &= \theta' \\
\frac{d\phi}{dt} &= \phi'
\end{aligned}
\]

(1)

where $[X,Y,Z]$ represents the position of the UAV in the coordinate system; $v$ represents the speed of UAV; $\theta$ represents the pitch
angle of UAV, is ranged from \([-90^\circ,90^\circ]\); \(\varphi\) represents the heading angle of UAV, is ranged from \([-180^\circ,180^\circ]\); \(dt\) represents integration step; \(a\) represents UAV acceleration; \(\theta '\) represents pitch angle variation; and \(\varphi '\) represents UAV heading angle variation.

The UAV is regarded as mass point when observing the movement of it. According to the principle of integration, the motion equation of the UAV with three degrees of freedom is shown in Eq. (1). Limited by UAV's throttle and overload performance, the maneuvering process of UAV in 3D space can be realized by setting suitable \(v\), \(a\), \(\theta '\) and \(\varphi '\).

The two sides in the battle are modeled in the OXYZ coordinate system, as shown in Fig. 1.

![Fig. 1. Air combat confrontation situation map.](image)

As shown in Fig. 1, \(O\) represents the position of our side in 3D space, and \(M\) represents the position of the enemy side. Our situation information includes the speed vector of UAV \(\mathbf{v} = (v_x, v_y, v_z)\), the position vector of UAV \(\mathbf{R} = (X, Y, Z)\), pitch angle \(\theta\), heading angle \(\varphi\), and the speed of UAV \(v\). Enemy situation information includes the speed vector of UAV \(\mathbf{v}_m = (v_{m_x}, v_{m_y}, v_{m_z})\), the position vector of UAV \(\mathbf{R}_m = (X_m, Y_m, Z_m)\), pitch angle \(\theta_m\), heading angle \(\varphi_m\), and the speed of UAV \(v_m\). The relative position vector between our UAV and enemy UAV is \(\mathbf{D}\), the direction of relative position vector is from our side to the enemy. The distance is between our UAV and enemy UAV is \(d\). The angle of \(\mathbf{V}\) and \(\mathbf{D}\) is relative azimuth \(q\).

Therefore, the combat situation of enemy and mine can be described by \(\mathbf{D}\), \(d\) and \(q\). The mathematical description of \(\mathbf{D}\), \(d\) and \(q\) is shown in Eq. (2) to Eq. (4).

\[
\mathbf{D} = \mathbf{R} - \mathbf{R}_m \quad (2)
\]
\[
d = ||\mathbf{D}|| \quad (3)
\]
\[
q = \arccos \left( \frac{\mathbf{D} \cdot \mathbf{V}}{||\mathbf{D}|| ||\mathbf{V}||} \right) \quad (4)
\]

### III. Intelligent UAV maneuvering decision-making algorithm based on deep reinforcement learning

#### A. Task specification

In air combat, the maneuvering decision-making of UAV plays a significant role in the combat result. After initializing the positions of the UAVs on both sides of the battle, the UAV can automatically generate maneuvering decision based on the battlefield situation information based on the deep reinforcement learning algorithm, so that it can occupy a favorable position in the air combat. In consequence, the lock-in and preemptive attack on the enemy has been realized.

The combat environment in this paper includes UAVs on both sides of the battle. The entire combat process is divided into three parts: the situation information acquisition module of both sides, the maneuvering decision module based on the
deep reinforcement learning algorithm, and the motion module. Among them, the situation information acquisition module of both sides calculates situation information and provides it to the decision module for decision making; the maneuvering decision module generates maneuvering control quantity based on deep reinforcement learning algorithm and provides it for the motion module used to our UAV maneuvering; the motion module updates our position information through the motion equations of UAV, realizes maneuvering, and provides information to the situation information acquisition module of both sides for calculating the corresponding situation.

B. Related theory
TD3[^16] algorithm is an actor-critic algorithm that can operate over continuous action spaces. DDPG[^15] algorithm is the theoretical basis of TD3 algorithm. DDPG is an actor-critic, model-free algorithm based on deterministic policy gradient that can operate over continuous action spaces. However, there is a shortcoming in DDPG algorithm that the estimated value function is larger than the true value function. To solve this problem, TD3 algorithm improves policy network and value network on the basis of DDPG algorithm, which makes TD3 algorithm perform better than DDPG in many continuous control tasks. The structure of TD3 is shown in Fig. 2.

Two sets of Critic networks $Q_1$, $Q_2$ are used to calculate different $Q$ in TD3 algorithm. And the minimum $Q$ of the two networks is selected to calculate the target $Q$, thereby suppressing continuous overestimation. As same as DDPG, Actor network $\pi_a$ is used to output action. Therefore, there are three sets of six neural networks in TD3 algorithm, including two Actor networks and four Critic networks. After the Critic network has been updated many times, the Actor network is updated. The delay of parameter update is to allow the Actor network to make action decisions after the Critic network is not overrated.

Select action with exploration noise shown as follows:

$$a \sim \pi_a(s) + \epsilon$$

where $\epsilon \sim N(0,\sigma)$

The calculation formula of target $Q$ is as follows:

$$Q(s_{t+1},a_{t+1}) = \min(Q_1(s_{t+1},a_{t+1}),Q_2(s_{t+1},a_{t+1}))$$ (6)

The Critic network is updated by calculating the loss function of the Critic network. The loss function is shown as follows:

$$Loss_1 = \frac{1}{N}\sum_i((r_i + \gamma Q(s_{t+1},a_{t+1})) - Q(s_t,a_t))^2$$ (7)

$$Loss_2 = \frac{1}{N}\sum_i((r_i + \gamma Q(s_{t+1},a_{t+1})) - Q(s_t,a_t))^2$$

The Actor network is updated by policy gradient as follows:

$$\nabla_a J(\phi) = \frac{1}{N}\sum_i \nabla_a Q(s,a)|a = \pi_a(s)\nabla_a \pi_a(s)$$ (8)

The target networks are updated as follows:

$$\theta_i \leftarrow \tau \theta_i + (1-\tau)\theta_i'$$

$$\phi_i \leftarrow \tau \phi_i + (1-\tau)\phi_i'$$ (9)

C. UAV maneuvering decision-making algorithm based on deep reinforcement learning.

Markov Decision Process (MDP) can be used to model RL problems. This model describes the relationship among the state, action and return of the agent in the environment. In this section, based on the MDP, a model is established for the UAV maneuvering decision-making in the air combat process, and the state space, action space and reward function are defined. Finally, the algorithm flow is introduced.
Fig. 2. The structure of TD3.

1) The design of action space and state space
The UAV makes maneuvering decisions based on the situation data, and executes maneuvers after obtaining the maneuvering decisions, so that the enemy enters its own missile attack envelop to complete the combat mission. Therefore, the state space includes the UAV's own information and the enemy information that can be obtained, and the action space is the control quantity of the UAV's actions.

According to Eq. (1) to Eq. (4), the state space of this paper is described by a tuple including eight elements and expressed in a vector as follows:

$$[X, Y, Z, v, \theta, \phi, d, q]$$

(10)

where $X, Y, Z$ respectively represent the position of our UAV on the three coordinate axes, $v$ represents the speed of our UAV, $\theta$ represents the pitch angle of our UAV, $\phi$ represents the heading angle of our UAV, $d$ represents the distance between us and the enemy, and $q$ represents the relative azimuth of enemy.

According to Eq. (2) to Eq.(4), it can be seen that the motion of UAV is controlled by acceleration $a$, pitch angle variation $\theta'$, and heading angle variation $\phi'$. Therefore, the action space of UAV can be designed as a tuple including three elements and expressed by a vector as follows:

$$[a, \theta', \phi']$$

(11)

2) The reward function of the air combat situation
We need to get enemy UAV into our missile attack envelop to accomplish our mission. The range of missile attack envelop is determined by the air-to-air missile's maximum attack distance $D_{max}$, minimum attack distance $D_{min}$ and maximum off-axis launch angle $q_{max}$. And air-to-air missiles need a certain enemy lock time $t_{max}$ before they can be launched.

The time that the enemy has been continuously in our missile attack envelop is $t$. When Eq. (12) is satisfied, it can be considered that our missiles were successfully launched, and the enemy was destroyed by our missiles, and our combat was successful. The reward function in this paper is composed of continuous rewards and sparse rewards. Among them, the continuous reward function is negatively correlated with the relative azimuth and the relative distance. According to Eq. (12), the angle reward and distance reward have been considered in this paper.
\[
\begin{align*}
D_{\text{max}} < D < D_{\text{max}} \\
q < q_{\text{max}} \\
t_{\text{w}} > t_{\text{max}}
\end{align*}
\]

The angle reward \( r_a \) is shown as follows:
\[
\begin{align*}
r_{a_1} &= -q / 180 \\
r_{a_2} &= \begin{cases} 
3, & \text{if } q < q_{\text{max}} \\
0, & \text{else}
\end{cases}
\end{align*}
\]

The distance reward \( r_d \) is shown as follows:
\[
\begin{align*}
r_a &= -d / (5 \cdot D_{\text{max}}) \\
r_{a_1} &= \begin{cases} 
3, & \text{if } D_{\text{min}} < d < D_{\text{max}} \\
-1, & \text{if } d < D_{\text{min}}
\end{cases}
\end{align*}
\]

where \( r_{a_1} \) is continuous reward, \( r_{a_2} \) is sparse reward, \( D_{\text{min}} \) is minimum attack distance, \( D_{\text{max}} \) is maximum attack distance, \( d \) is the distance between us and the enemy.

The reward function is defined as follows:
\[
R = u_1 \cdot R_{a_1} + u_2 \cdot R_{a_2}
\]

where \( u_1, u_2 \) are weight coefficients, and we set \( u_1 = u_2 = 0.5 \).

**Table 1.** Training process of the UAV maneuvering decision-making algorithm based on TD3.

<table>
<thead>
<tr>
<th>Algorithm TD3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong></td>
</tr>
<tr>
<td><strong>Output:</strong></td>
</tr>
</tbody>
</table>

1. Initialize Critic networks \( Q_\theta, Q_{\phi_2} \), and Actor network \( \pi_{\theta_0} \) with random parameters \( \varphi_1, \varphi_2, \theta \).
2. Initialize target network \( Q' \) and \( \mu' \) with weights \( \vartheta \leftarrow \theta, \varphi_1 \leftarrow \varphi_1, \varphi_2 \leftarrow \varphi_2 \)
3. Initialize replay buffer \( R \)
4. **FOR** \( t=0, T \) **do**:
5. Select action with exploration noise \( a = \pi_{\theta_0}(s) + \xi \), where \( \xi \) obey normal distribution
6. The UAV performs action \( a \), and observes reward \( r \) and new state \( s' \)
7. Store transition \( (s, a, r, s') \) in \( R \)
8. **IF** the transition storage capacity in \( R \) is larger than batch_size:
9. Sample batch_size of transitions \( (s, a, r, s') \) from \( R \)
10. \( a' = \pi_{\theta_0}(s'), r = r + \gamma \min(Q_{\phi_1}(s, a), Q_{\phi_2}(s, a)) \)
11. \( \text{loss1} = \sum_{i=1}^{N} (Q_{\phi_1}(s, a) - y)^2, \text{loss2} = \sum_{i=1}^{N} (Q_{\phi_2}(s, a) - y)^2 \)
12. Update parameters of Critic network \( \varphi_1, \varphi_2 \)
13. **IF** \( t \mod d \) then
14. \( \nabla_\theta J(\theta) = \frac{1}{N} \sum \nabla_{s, \theta} Q_{\phi_1}(s, a) \big|_{a = \pi_{\phi_1}(s)} \nabla_\theta \pi_{\phi_1}(s) \)
15. Update parameters of Actor network \( \theta \)
16. Update target networks:
17. \( \theta' \leftarrow \theta + (1 - \tau) \theta \)
18. \( \varphi_1' \leftarrow \varphi + (1 - \tau) \varphi_1 \)
19. \( \varphi_2' \leftarrow \varphi + (1 - \tau) \varphi_2 \)
20. **END IF**
21. **Skip to step 5, and** \( s \leftarrow s' \)
22. **END FOR**
3) Algorithm procedure
According to the above definition, the training process of the UAV maneuvering decision-making algorithm based on TD3 is shown in Table 1.

IV. Experiment and analysis
A. Experimental parameter settings
1) Parameter settings of TD3 algorithm
The parameters of TD3 algorithm are shown in Table 2, where, training round represents the number of training rounds for the network in the algorithm in a certain initial state; the maximum simulation step size indicates the maximum number of actions performed by the agent in a training round, when this number of times is reached, the training of this round is over; the time step represents the time interval for the agent to perform actions; batch_size represents the number of samples taken from the replay buffer each time during training.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training round</td>
<td>2000</td>
</tr>
<tr>
<td>Maximum simulation step size</td>
<td>800</td>
</tr>
<tr>
<td>Time step</td>
<td>0.1</td>
</tr>
<tr>
<td>batch_size</td>
<td>256</td>
</tr>
<tr>
<td>Discount factor</td>
<td>0.99</td>
</tr>
<tr>
<td>Exploration noise</td>
<td>$N(0,0.2)$</td>
</tr>
<tr>
<td>Action bound</td>
<td>[-1,1]</td>
</tr>
<tr>
<td>Optimization</td>
<td>Adam</td>
</tr>
<tr>
<td>Learning rate of Actor</td>
<td>0.001</td>
</tr>
<tr>
<td>Learning rate of Critic</td>
<td>0.0001</td>
</tr>
</tbody>
</table>

2) Parameter settings of missile and UAV
The parameters of missile and UAV are shown in Table 3. It is assumed that the target locking time of the missile is 2 seconds. When the time step is 0.1s, the missile needs to lock the target with 20 simulation steps to launch.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>heading angle variation range</td>
<td>6°</td>
</tr>
<tr>
<td>pitch angle variation range</td>
<td>4°</td>
</tr>
<tr>
<td>Maximum variation of speed</td>
<td>20 m/s</td>
</tr>
<tr>
<td>Maximum speed of UAV</td>
<td>350 m/s</td>
</tr>
<tr>
<td>missile's maximum attack distance</td>
<td>6km</td>
</tr>
<tr>
<td>missile's minimum attack distance</td>
<td>1km</td>
</tr>
<tr>
<td>maximum off-axis launch angle</td>
<td>30°</td>
</tr>
<tr>
<td>Lock-time of UAV</td>
<td>2s</td>
</tr>
</tbody>
</table>

Fig. 3. The structure of network.
As shown in Fig. 3, the policy network Actor outputs the maneuvering action based on current state. According to the state and action space of UAV maneuvering decision-making, the number of Actor network input nodes is 8, and the number of Actor network output nodes is 3. And since the activation function of output layer is tanh function, the output is limited to [-1,1]. The value network Critic is used to evaluate the value of decision that performs the action in current state. The number of neurons in the input layer and output layer are 11 and 1 respectively. Both Actor and Critic networks are fully connected neural networks with two hidden layers. The number of neurons in
the hidden layer is 256, and the activation function is the Relu function.

**B. Simulation experiment and analysis**

In this section, the application of TD3 algorithm and DDPG algorithm in air combat maneuvering decision task is realized by designing relevant experiments, and the efficiency of the algorithm is compared. In the experiment, the red side is an intelligent body that uses deep reinforcement learning algorithms, and the blue side is a non-intelligent body that performs fixed maneuvers. The initial distance between UAVs is 15km, and the initial relative azimuth is $40^\circ$. The parameters and the structure of network of DDPG algorithm are same as TD3 algorithm in Section A.

1) **Convergence speed comparison**

In order to better evaluate the convergence speed of the algorithm, the total reward obtained by us in each round was recorded during the experiment to determine whether the reward function converges. The change curves of total reward of DDPG algorithm and TD3 algorithm in 4000 training rounds under the same initial conditions are shown in Fig. 4.

As shown in the Fig. 4, the TD3 algorithm converged locally within 250 to 2800 rounds, and jumps out of local convergence at 2800 rounds to achieve global convergence. The DDPG algorithm reached local convergence in multiple stages, and finally did not jump out of local convergence. In the end, the TD3 algorithm converges 200 rounds earlier than the DDPG algorithm, and the maximum reward value that it converges to is greater. At the same time, the DDPG algorithm has a lot of jitter during the convergence process, and the TD3 algorithm is more stable. Therefore, TD3 algorithm has faster training speed and better training results compared with DDPG algorithm.

2) **Test results comparison**

Fig. 5 and Fig. 6 respectively show the combat process of UAV approaching the enemy and meeting launch conditions in different planes.

**Fig. 5.** Top view of combat trajectory.

Fig. 5 shows the combat trajectory of the UAV on the horizontal plane. It can be seen from the Fig. 5 that after the start of the battle, the blue side with no attack ability moves randomly, and the initial relative azimuth angle and distance of the blue side's UAV relative to the red side's UAV are relatively large. In order to make the blue side enter its missile launch area, the red
side first quickly changed the heading, reduced the relative azimuth angle, and made a tail-back attack on the blue side.

Fig. 6 shows the altitude change of the UAV during combat. As can be seen from the Fig. 6, in the initial state, when the enemy and our side had a height difference and the enemy was lower than us, the red side of TD3 algorithm gradually reduced the height difference during the movement, while the red side of DDPG algorithm always had a large height difference and was always above the enemy's height.

The decision-making process of the two algorithms is to first change the direction, reduce the relative azimuth angle, and then shorten the distance, and finally reached an attack situation that satisfies the launch conditions. However, by comparing Fig. 5 and Fig. 6, it can be seen that the red side turning range in the early stage of the TD3 algorithm is smaller, and the relative azimuth angle is reduced faster. When the launch conditions are finally met, the red side of the TD3 algorithm is closer to the enemy than the DDPG algorithm, and the relative azimuth angle is smaller.

Comprehensive comparison of combat trajectories, compared with the DDPG algorithm, the maneuver strategy generated by the TD3 algorithm can enable the red side to meet the launch conditions more quickly, strike the enemy successfully, which is more suitable for actual combat.

V. Conclusion
In this paper, a UAV combat maneuvering decision-making algorithm based on deep reinforcement learning is established. UAV maneuvering model and UAV combat model are established by mathematical algorithm. At the same time, in order to make the battlefield environment more real, the concept of missile attack envelop is introduced in the process of confrontation. Then, this paper realizes the UAV air combat maneuvering decision-making based on DDPG and TD3 algorithm. Experimental results show that compared with DDPG algorithm, TD3 algorithm has better convergence speed and optimization ability, and is more suitable for solving UAV maneuvering decision problem.

Acknowledgements
The authors would like to acknowledge National Natural Science Foundation of China (Grant No. 61573285, No.62003267), Open Fund of Key Laboratory of Data Link Technology of China Electronics Technology Group Corporation (Grant No. CLDL-20182101) and Natural Science Foundation of Shaanxi Province (Grant No. 2020JQ220) to provide fund for conducting experiments.
References


