Proceedings of the 41st Chinese Control Conference
July 25-27, 2022, Hefei, China

Air Combat Maneuver Decision Based on Deep Reinforcement Learning and Game Theory

Shuhui Yin[1], Yu Kang[1,2], Yunbo Zhao[1], Jian Xue[3]

1. Department of Automation, University of Science and Technology of China, Hefei 230026, China
E-mail: shuhui@mail.ustc.edu.cn, kangduyu@ustc.edu.cn, ybzhao@ustc.edu.cn

2. Institute of Advanced Technology, University of Science and Technology of China, Hefei 230088, China

3. College of Engineering and Information Technology, University of Chinese Academy of Sciences, Beijing 100049, China
E-mail: xuejian@ucas.ac.cn

Abstract: The autonomous maneuver decision of UAV plays an important role in future air combat. However, the strong competitiveness of the air combat environment and the uncertainty of the opponent make it difficult to solve the optimal strategy. For these problems, we propose the algorithm based on deep reinforcement learning and game theory, which settles the matter that the existing methods cannot solve Nash equilibrium strategy in highly competitive environment. Specifically, 1v1 air combat is modeled as a two-player zero-sum Markov game, and a simplified two-dimensional simulation environment is constructed. We prove that the algorithm has good convergence through the simulation test. Compared with the opponent's strategy using DQN, our algorithm has better air combat performance and is more suitable for the air combat game environment.

Key Words: Air Combat, Reinforcement Learning, Game Theory, Maneuver Decision

1 Introduction

With the informatization development of modern military warfare, air combat game of UAV is the main means to master aerial supremacy. Compared with the manned aircraft, it greatly reduces the risk of performing missions and improves combat efficiency. The traditional UAV works by ground stations [1], which is difficult to complete the accurate and timely control. Thus, it's of great significance to study the autonomous maneuver decision-making [2] capacity of UAV.

The modeling and calculating of traditional air combat maneuver autonomous decision are based on expert domain knowledge [3,4]. This method essentially relies on the experience of human experts, making it restrictive to completely cover all air combat situations due to the complex construction of the repository. In addition, methods based on supervised learning [5,6] have strong robustness and adaptability, but require a large amount of training data to obtain ideal maneuvering decision-making effects.

Reinforcement learning [7] adopts trial and error to interact with the environment. It evaluates the result of maneuver selection by calculating the cumulative rewards after performing the action in the current state. Therefore, reinforcement learning not only considers the influence of the current state on the air combat situation, but also considers the long-term impact of maneuvering actions, which can well satisfy the uncertainty in the air combat process. Furthermore, reinforcement learning does not require training samples [8] and deals with the problem of maneuvering decision-making effectively.

One of the challenges in applying reinforcement learning to air combat is how to store the action-value function. Due to the high-dimensional state space of air combat, traditional reinforcement learning algorithms suffer from a dimensional explosion. The emergence of deep reinforcement learning [9] solves this problem, which combines deep neural networks and reinforcement learning. The nonlinear fitting ability of deep neural networks [10] is used to break through the limitations of finite-dimensional state input, making air combat UAVs more capable of dealing with complex problems.

In the 1v1 air combat task, both the enemy and us update and adjust their strategies in time according to the battlefield situation. This is a dynamic game process, which is highly confrontational and involves complex conflicts of interest. This results in that air combat decision using the existing methods [11] cannot make specific decisions against the opponent's strategy in a highly competitive environment, so the win rate is low. Game theory [12] model expresses the interaction between the strategies of two sides, and acquires the optimal strategy of friend or foe. However, in traditional game, players have no ability to learn. They pay more attention to the strategies of other players at the current step. On this account, we combine the self-learning ability of reinforcement learning, the ability of neural network to handle high-dimensional states, and the idea of equilibrium decision in game theory, and apply them to air combat problems, aiming to obtain more intelligent and adversarial autonomous maneuvering decisions.

In this paper, we propose the algorithm based on deep reinforcement learning and game theory to deal with strong competitiveness of the air combat environment and the uncertainty of the opponent. This algorithm solves the problem that the existing methods only unilaterally optimize its own interests without considering the adversarial factors in the air combat game. In addition, we model the 1v1 air combat as a two-player zero-sum Markov game and construct a simplified two-dimensional simulation environment. We prove that our algorithm has good convergence through the simulation test. Compared with the opponent's strategy using DQN, our algorithm can more effectively make real-time decisions for the opponent's

*This work was supported by the National Key Research and Development Program of China (No. 2018AAA0100801).

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on December 01,2022 at 13:47:57 UTC from IEEE Xplore. Restrictions apply.

strategy, and is more suitable for the air combat game environment.

The rest of this paper is organized as follows. In Section 2, the air combat game is modeled. In Section 3, we describe our method. In Section 4, we demonstrate the experiments and analyze results. We conclude in Section 5.

2 Model

2.1 Two-player Zero-sum Markov Game

In this paper, we model the 1v1 air combat problem as a two-player zero-sum Markov Game.

Two-player zero-sum Markov Game [13] is an extension of Markov decision progress [14] combined with zero-sum matrix game. Markov decision process is a multi-process decision-making theory. The agent continuously interacts with its environment, gets feedback and makes actions according to the feedback to optimize its benefits. The zero-sum matrix game describes a two-player static zero-sum game. The static means that both players make actions at the same time. The zero-sum means that the sum of the payoff functions of two players is zero. We can obtain a two-player multi-process dynamic decision-making model combining the two, that is, Markov Game model, which can be defined as a quintile (S,A,O,T,R):

- *S*: environment state space.
- *A*: agent action space.
- *O*: opponent action space.
- *T*: $S\times A\times O\times S\to[0,1]$, represents the transition probability function from one state to the next state:

$$T(s,a,o,s')=P(s'\mid s,a,o) \tag{1}$$

where, $s,s'\in S$, represent the state of the environment. $a\in A$, $o\in O$, represent the actions of the agent and opponent respectively. *P* represents the conditional probability.

- *R*: the reward function of agents, $S\times A\times O\times S\to R$, which represents the expectation reward from executing the action to the next state in the current state:

$$R(s,a,o,s')=E\{r_{t+1}\mid s_t=s,a_t=a,o_t=o,s_{t+1}=s'\} \tag{2}$$

where, r_{t+1} represents the direct reward obtained from the moment $t+1$.

The decision-making basis of the agent is to maximize its own reward, so its goal is to find a strategy π, so that the agent can obtain the largest cumulative expectation reward according to the strategy π in the process of the game:

$$G^{\pi,\pi^-}(s)=E_{\pi,\pi^-}\left\{\sum_{k=0}^{n}\gamma^k R_{k+t+1}\mid s_t=s\right\} \tag{3}$$

where π and π^- represent the strategies of the agent and the opponent respectively. *n* represents the number of steps from the current moment to the termination moment. $\gamma\in[0,1]$ represents the discount factor.

2.2 1v1 Air Combat Description

A Aircraft Dynamics Model

Firstly, we establish a mathematical model of the aircraft in an ideal two-dimensional plane through dynamic analysis, and present a series of assumptions [15] to simplify the study.

- The ground is assumed to be an inertial reference frame.
- The mass, gravitational acceleration and rotational inertia of the aircraft are invariable.
- Regardless of the sideslip angle of the aircraft.
- The direction of the speed and the body are approximately coincident.

The aerodynamics equation [15] for the aircraft are shown in Eq. (4).

$$\begin{cases}\dot{x}=v\cos\psi\\ \dot{y}=v\sin\psi\\ \dot{v}=\dfrac{u_t}{m}\\ \dot{\varphi}=u_{\varphi}\\ \dot{\psi}=\dfrac{g\tan\varphi}{v}\end{cases} \tag{4}$$

where, g = 9.81. φ is expressed as the rotation angle of the body coordinate system about *o-x* axis. ψ is expressed as the rotation angle of the body coordinate system about *o-y* axis. The input of the dynamic model is thrust, and the output is velocity and angular velocity.

B State Space

1v1 air combat model is based on the two-player zero-sum Markov Game. *S* is the state space of air combat, composed of positions, velocities, angles:

$$S=[x_b,y_b,x_r,y_r,v_b,v_r,\phi_b,\phi_r,\psi_b,\psi_r] \tag{5}$$

where, (x_b,y_b), (x_r,y_r) are the positions of the blue aircraft and the red aircraft respectively. Both aircrafts fly in the two-dimensional plane with no restrictions on their positions. v_b and v_r are the speeds of blue and red respectively, limited to a certain range. ϕ_b and ϕ_r are the roll angles, confined to the maximum turning maneuverability of the blue and red. ψ_b and ψ_r are the heading angles, which can take any value within $\pm 180°$.

C Action Space

1v1 air combat is a dynamic and continuous process of confrontation. In the current state, the aircraft calculates the state value at the next moment according to the kinetic equation. We utilize two continuous variables to control the maneuvering of the aircraft. In two-dimensional plane, we leave out the pitch and sideslip motion of the aircraft. The action space is defined as follows:

$$A=(u_t,u_{\varphi}) \tag{6}$$

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on December 01,2022 at 13:47:57 UTC from IEEE Xplore. Restrictions apply.

where, u_t is the thrust, used to control the velocity of the aircraft. u_φ is the roll angular rate, which commands the turning maneuverability and decides how much the aircraft can roll. Both two are restricted according to the actual performance of the aircraft.

D Rewards

In the 1v1 air combat mission set in this paper, the blue aircraft's target is to rapidly reach and maintain an advantage position behind the red aircraft, making it easy to launch a missile to destroy the red aircraft. Aspect angle (AA) and antenna train angle (ATA) are displayed to quantify the rewards (see Fig. 1).

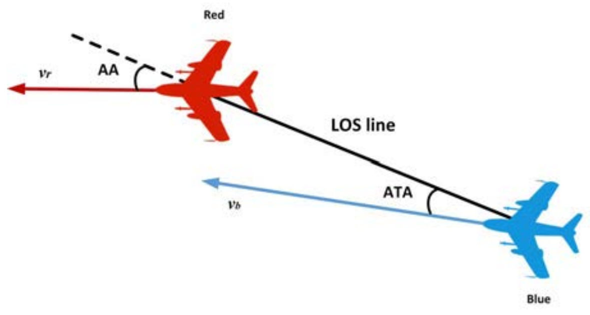


Fig. 1: AA and ATA

In Fig. 1, the blue aircraft's antenna train angle (ATA) is the angle between LOS line and its velocity, which reflects the degree of following the red aircraft. The blue's aspect angle (AA) is the angle between LOS line and the red's velocity, which reflects the degree of steadily following the opponent. Both the ATA and AA can take any value within ±180°.

1v1 air combat belongs to a strictly competitive game, the success of one side corresponds to the failure of the other side, and the success and failure of both aircrafts add up to zero. If the blue aircraft reaches the enemy's attack zone, it will gain a positive reward at the current step. Conversely, it will receive a negative reward if the opponent tracks its attack zone. In other conditions, the reward is zero. The reward function is designed in Eq. (7):

$$R_b = -R_r = \begin{cases} 1.0, & |ATA_b| < 30^\circ \wedge |AA_b| < 60^\circ \\ -1.0, & |ATA_r| < 30^\circ \wedge |AA_r| < 60^\circ \\ 0, & otherwise \end{cases} \tag{7}$$

The ATA and AA of two sides satisfy the following geometric relationship [16]:

$$\begin{gathered} |ATA_b| + |AA_r| = 180^\circ \\ |ATA_r| + |AA_b| = 180^\circ \end{gathered} \tag{8}$$

3 Method

3.1 Nash Equilibrium

The most widely used concept of a solution in game theory is the Nash equilibrium [17], which embodies the steady state of strategic game action. Game theory deals with strategic interactions between players, whose decisions are mutually influenced. Players make decisions under iterative considerations and choose the most beneficial strategy for themselves. Nash equilibrium refers to a state of the game in which no player can benefit more from changing his current decision if the other player's strategy remains unchanged. In other words, Nash equilibrium is a combination of strategies where each player's strategy is the optimal strategy for the others. Similar to the optimal solution in machine learning, finding Nash equilibrium is the optimal strategy solution problem.

Nash equilibrium joint strategy [18] can be expressed as $\pi^* = (\pi_1^*, \cdots, \pi_n^*)$, $V_i = (\pi_1^*, \cdots, \pi_n^*)$ represents the expected payoff of the players under the joint strategy, that is, the value function. For all $s \in S$ and $i = 1, \cdots, n$:

$$V_i(s, \pi_i^*, \pi_{-i}^*) \ge V_i(s, \pi_i, \pi_{-i}^*), \forall \pi_i \in \prod_i \tag{9}$$

where, π_{-i}^* is the joint strategy expect the player *i*, $\prod_i$ is the policy profile of *i*, $V_i(s, \pi_i^*, \pi_{-i}^*)$ is the discounted payoff of all players in the current state *s*. $Q_i(a_1, \cdots, a_n)$ is defined to represent the expected reward obtained by the player *i* when performing joint actions. $\pi_i(a_i)$ is denoted as the probability of the player *i* choosing an action a_i.Then Nash equilibrium can be expressed as:

$$\begin{aligned} &\sum_{a_1, \cdots, a_n \in A_1 \times \cdots \times A_n} Q_i(a_1, \cdots, a_n) \pi_1^*(a_1) \cdots \pi_n^*(a_n) \ge \\ &\sum_{a_1, \cdots, a_n \in A_1 \times \cdots \times A_n} Q_i(a_1, \cdots, a_n) \pi_1(a_1) \cdots \pi_n^*(a_n), \forall \pi_i \in \prod_i \end{aligned} \tag{10}$$

3.2 Deep Q Learning

Traditional Q learning uses a Q table to store the Q value of each state-action, which leads to an explosion of dimension when the state and action spaces are high-dimensional and continuous. In order to solve this problem, DQN integrates deep neural network and reinforcement learning, takes advantage of the nonlinear fitting ability of deep neural network to approximate the state-action value function, and iteratively updates the policy through gradient descent algorithm [10].

In addition to using a neural network to approximate the value function, experience reply and target network are proposed as two innovative points of DQN. Experience reply breaks the correlation between training data. It not only solves the problem of large parameter update variance caused by continuous samples, but also improves the utilization of the data. The target network has the same structure as the Q network, but does not train itself. The purpose of constructing the target network is to enhance the stability of training Q network and reduce the non-convergence of the network weights. Therefore, the update of the value function is expressed as:

$$Q_{t+1}(s, a) = (1-\alpha) Q_t(s, a) + \alpha [r + \gamma \max Q(s', a')] \tag{11}$$

3.3 DQN for Two-player Game

Traditional reinforcement learning has the problem of dimensional explosion and tends to unilaterally maximize its own interests. As a result, deep neural network is introduced to approximate the Q function for the high-dimensional and

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on December 01,2022 at 13:47:57 UTC from IEEE Xplore. Restrictions apply.

continuous state space of air combats. And we apply the concept of Nash equilibrium to solve the air combat maneuver decision problem with strong confrontation and complex conflict. Therefore, we propose the algorithm extending DQN to the two-player zero-sum Markov Game. This algorithm can be used to acquire approximate equilibrium maneuver decisions combined with the opponent's strategy in 1v1 air combat scenarios. The algorithm flow is shown in Algorithm 1.

```
Algorithm 1: DQN for Two-player Game
Input: $S, A^b, A^r$, training parameters
Initialize replay memory $D$ to capacity $N$
Initialize $Q$-network with random weights $\theta$
Initialize target network $Q^-$ with weights $\theta^- = \theta$
Initialize $Q_{b,r}(s, a^b, a^r) = 0 \; \forall a^b \in A^b, a^r = A^r$
for episode=1, M do
    Get initial state $S_0$
    Select $A_t^b$ according to exploration utilization method
    Get $A_t^r$ according to environment settings
    Execute $A_t^b$ and $A_t^r$, get $R_t$, enter next state $S_{t+1}$
    Store transition $(S_t, A_t^b, A_t^r, R_t, S_{t+1})$ in $D$
    Sample minibatch of $(S_i, A_i^b, A_i^r, R_i, S_{i+1})$ from $D$
    Get $NashQ_t^i(s') = \pi^b(s')\pi^r(s') \cdot Q_t^i(s')$ by linear program
    for each $i \in b, r$, compute the target:
    $Q_{t+1}^i(s, a^b, a^r) = (1-\alpha_t)Q_t^i(s, a^b, a^r) + \alpha_t[r_t^i + \gamma NashQ_t^i(s')]$ (12)
    Update $Q$-network: perform a gradient descent step
    Update target $Q^-$: update $\theta^- \leftarrow \theta$ every $C$ steps
end for
Output: $Q$-network and equilibrium policy
```

4 Experiment and Analysis

A Simulation Environment

We construct a dynamically simplified two-dimensional plane. Our aircraft and the enemy aircraft are represented by blue and red particles respectively. The aircraft's dynamic model and air combat environment are programmed by Python 3, which integrally logs the state data of the two aircrafts. The simulation environment is shown in Fig. 2.

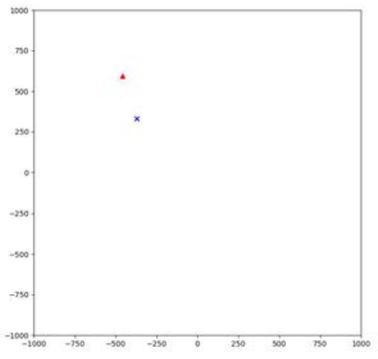
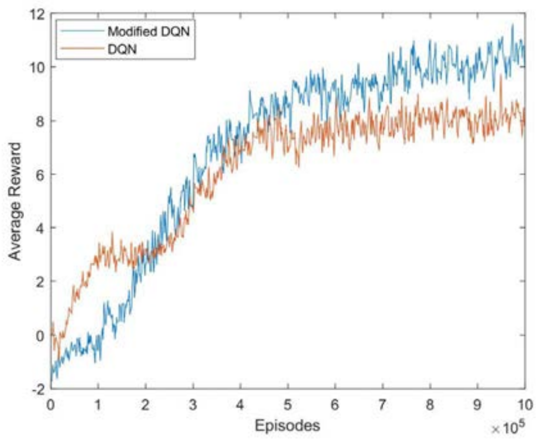


Fig. 2. The air combat simulation environment

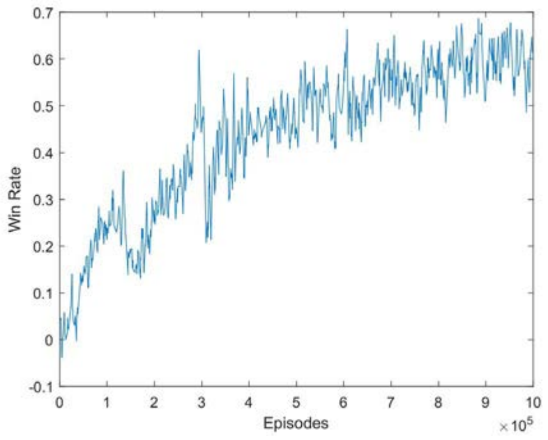
B Training

The blue aircraft model is trained by modified DQN, and the red is trained by DQN. Tensorboard is used to observe the entire training process. We simulate a total of 1 million episodes. The exploration rate drops from 1 to 0.05 within 100000 episodes, and remains at 0.05 in the rest of episodes. The buffer size is 10^6, the batch size is 32, the discount factor γ is 0.99 and the learning rate α is 0.0001. The target update interval is set to 1000. Fps is about 714, means that 714 actions can be completed in one second. If an air combat is set to 25 steps, that is, 28 battles can be fought in one second. The airspace is limits to ± 1000. At the beginning of each episode, the two aircraft are in random states.

The average reward of each episode is used to measure the air combat advantage. The superiority of two algorithms are visually compared by the win rate. The convergence process of average reward for the two aircrafts and the win rate are severally shown in Fig. 3.



(a) Average Reward



(b) Win Rate of Modified DQN

Fig. 3. The convergence process of average reward and win rate in the training process.

In Fig.3(a), the average reward of both modified DQN and DQN tends to be stable, proving that the two algorithms have good convergence. However, the average reward of modified DQN is higher than DQN at the end of the training process. In Fig.3(b), Modified DQN did not explore better strategies at the beginning, resulting in a very low win rate

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on December 01,2022 at 13:47:57 UTC from IEEE Xplore. Restrictions apply.

of confrontation. With the increase of training times, it rises gradually and finally converges to about 60 percent.

C Evaluating

In order to evaluate strategies generated by the two algorithms, the two aircrafts fight against each other for 10000 times. We define that the blue aircraft wins the game if it satisfies Eq. (7) for ten steps in one air battle, and it loses in other cases. The game results are recorded in Table 1.

It can be seen that the blue aircraft wins 6113 times and the red wins 3887 times. The win rate of blue aircraft is 61.1% by calculating, shows that it has a slight advantage over the red aircraft in the same adversarial environment. This proves that Modified DQN learns more intelligent and accurate maneuver strategies in strong confrontational environment, and it is more suitable for the air combat game compared with DQN.

Table 1: Modified DQN vs DQN

Result	Modified DQN	DQN
Win	6113	3887
Lose	3887	6113
Win rate	61.1%	38.9%

5 Conclusion

In this paper, an air combat maneuver strategy algorithm is proposed based on deep reinforcement learning and game theory. In terms of the strong competitiveness of the air combat environment and the uncertainty of the opponent, this algorithm deals with the problem that the existing methods are difficult to solve Nash equilibrium strategy in highly competitive environment. Simulation results show that this algorithm has good convergence. Moreover, it possesses better air combat performance compared with the opponent's strategy using DQN.

There are still some issues that have not been resolved. Our aircraft modeling and simulation environment is too simple and very different from real air combat scenarios. In addition, we only study the strategy problem of 1v1 air combat, whereas the real-world scenario is many-to-many air combat, which involves more complex cooperative and adversarial conflicts. Our future work will be devoted to the more challenging problems of air combat maneuver decision.

References

[1] H. Yu, L. Qi, Development and challenges of UAV autonomous control technology, *China New Communications*, 2020.

[2] S. Zhou, W. Wu, N. Zhang, at al., A review of autonomous air combat maneuver decision-making methods, *Aviation Computing Technology*, 42(1): 5, 2012.

[3] L. Fu, F. Xie, G. Lei, at al., An expert system for UAV air combat decision-making based on rolling time domain, *Journal of Beijing University of Aeronautics and Astronautics*, 041(011): 1994-1999, 2015.

[4] S. Gao, Application of expert system in multi-aircraft air combat tactical maneuver decision-making simulation system, *Academic Annual Meeting of Information Systems Engineering Professional Committee of Chinese Society of Systems Engineering*, 1997.

[5] B. Zhang, Y. Kou, W. Meng, at al., Close-range air combat situation assessment using deep belief network, *Journal of Beijing University of Beijing University of Aeronautics and Astronautics*, 43(7): 1450-1459, 2017.

[6] X. Yang, J. Ai, Research on the maneuvering strategy of UAV to evade missiles in autonomous air combat, *Journal of System Simulation*, 30(5): 10, 2018.

[7] C. Sun, H. Zhao, Y. Wang, at al., Decision-making method for autonomous maneuvering of unmanned aerial vehicles based on reinforcement learning, *Firepower and Command and Control*, 44(4): 8, 2019.

[8] R. Sutton, A. Barto, Reinforcement learning, *Journal of Cognitive Neuroscience*, 11(1): 126-134, 1999.

[9] Y. Li, Deep reinforcement learning: An overview, in *arXiv*: *1701.07274*, 2017.

[10] V. Mnih, K. Kavukcuoglu, D Silver, at al., Playing atari with deep reinforcement learning, in *arXiv*: *1312.5602*, 2013.

[11] X. Ma, L. Xia, Q. Zhao, Air-combat strategy using Deep Q-Learning, in *IEEE 2018 Chinese Automation Congress (CAC)*, 2018: 3952-3957.

[12] S. Tadelis, Game theory: an introduction, *Princeton University Press*, 2013.

[13] M. Johnson, S. Bhasin, W. Dixon, Nonlinear two-player zero-sum game approximate solution using a policy iteration algorithm, in *IEEE 2011 50th Conference on Decision and Control and European Control Conference*, 2011:142-147.

[14] Z. Wei, J. Xu, Y. Lan, at al., Reinforcement learning to rank with Markov decision process, in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2017: 945-948.

[15] W. Kong, D. Zhou, Y. Zhen, Air combat strategies generation of CGF based on MADDPG and reward shaping, in *IEEE 2020 International Conference on Computer Vision, Image and Deep Learning (CVIDL)*, 2020: 651-655.

[16] W. Kong, D. Zhou, K. Zhang, at al., Air combat autonomous maneuver decision for one-on-one within visual range engagement base on robust multi-agent reinforcement learning, in *IEEE 2020 16th International Conference on Control and Automation (ICCA)*, 2020: 506-512.

[17] E. Maskin, The theory of implementation in Nash equilibrium: A survey, 1983.

[18] J. Hu, M. Wellman, Nash Q-learning for general-sum stochastic games, *Journal of Machine Learning Research*, 4: 1039-1069, 2003.

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on December 01,2022 at 13:47:57 UTC from IEEE Xplore. Restrictions apply.