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

UAV Swarm Confrontation Based on Multi-agent Deep Reinforcement Learning

Zhi Wang[1], Fan Liu[2], Jing Guo[2], Chen Hong*[3], Ming Chen[3], Ershen Wang[2], Yunbo Zhao[4]

1. Department of General Aviation, Civil Aviation Management Institute of China, Beijing, 100102, China

2. School of Electronic and Information Engineering, Shenyang Aerospace University, Shenyang, 110136, China

3. College of Robotics, Beijing Union University, Beijing, 100101, China

Corresponding author: hchchina@sina.com

4. Department of Automation, University of Science and Technology of China, Hefei, 230022, China

Abstract: Multi-agent deep reinforcement learning (MADRL) has attracted a tremendous amount of interest in recent years. In this paper, we introduce MADRL into the confrontation scene of Unmanned Aerial Vehicle (UAV) swarm. To analysis the dynamic game process of UAV swarm confrontation, we build two non-cooperative game models based on MADRL paradigm. By using the multi-agent deep deterministic policy gradient (MADDPG) and the centralized training with decentralized execution method, we achieve the Nash equilibrium under 5 vs. 5 UAV confrontation scenes. We also introduce multi-agent soft actor critic (MASAC) method into the UAV swarm confrontation, simulation results indicate that the MASAC-based model outperforms the MADDPG-based model on exploring the UAV swarm combat environment, and more effectively converges to the Nash equilibrium. Our work will provide new insights into the confrontation of UAV swarm.

Key Words: UAV Swarm, Non-cooperative Game, Multi-agent, Deep Reinforcement Learning

1 Introduction

Unmanned Aerial Vehicles (UAVs) have been widely used in civil and military fields, including environmental monitoring, power inspection, disaster relief, counter-terrorism, etc. Autonomous decision-making of UAVs is a key problem for in-depth application in these fields, especially in the field of UAV swarm dynamic game[6]. The complexity of the execution of tasks and the uncertainty of the environment require UAVs are of high decision-making capabilities and autonomy. In recent years, Deep reinforcement learning (DRL) has achieved outstanding achievements in the field of autonomous decision-making, resulting in a dramatic increase in many inspiring applications. Recent works have explored beyond the single-agent scenarios and some multi-agent learning (MAL) methods have been proposed[2]. A multi-agent system can be denoted as a group of autonomous, interacting entities sharing a common dynamic environment, in which entities perceive with sensors, act with actuators and learn coordinated and confrontation strategies[2].

Multi-agent deep reinforcement learning (MADRL) is a powerful learning paradigm which incorporates MAL with DRL. Chen et al. presented a multi-agent collision avoidance algorithm based on deep reinforcement learning which includes a value network to encode joint configuration with neighbors[10]. It is known that continuous control of multi-agents is an important issue in complex dynamic environment. To deal with such continuous control problem, Lillicrap et al. proposed a deep deterministic policy gradient (DDPG) algorithm based on actor-critic (AC) architecture[11] and deep reinforcement learning[12]. Lowe et al. put forward an multi-agent actor-critic framework with respect to mixed cooperative-competitive environment[13], where action policies of other agents are considered and agents can successfully learn policies that require complex coordination.

To deal with the non-cooperative game with respect to the UAV swarm confrontation, the game can be transformed into a Markov Decision Process (MDP)[8]. Considering that the model cannot be trained directly in the actual UAV's fight, interactive virtual simulation environment is widely used to train the model. In the UAV swarm confrontation game, each UAV performs tasks independently and maximizes its own interests, meanwhile maximizes the benefits of its entire swarm. The independence and autonomy of single UAV need be taken into account, and the autonomous decision-making between UAV swarms should also be considered. There are two major difficulties in the scene of UAV swarm confrontation: (1) incomplete observation of single UAV, namely each UAV cannot fully perceive the information of the environment; (2) the trade-off between exploration and utilization.

Most previous works use computer games to study the multi-agent cooperative-competitive game. In this paper, we introduce MADDPG and MASAC algorithms into the confrontation scenes of UAV swarm, and construct the MADDPG-based and the MASAC-based UAV swarm confrontation models, respectively. The result implies that UAV swarm under MASAC can more effectively explore the combat environment and converge to the Nash equilibrium than that of MADDPG.

*This work is supported by the National Key R&D Program of China (Grant No. 2018AAA0100804), the Beijing Education Commission Science and Technology Project (KM201811417005, KM201911417010), the Zhejiang Key laboratory of General Aviation Operation technology(JD GA2020-7).

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on December 01,2022 at 13:44:28 UTC from IEEE Xplore. Restrictions apply.

2 Methodology

2.1 The partially observable Markov game

We consider a multi-agent extension of the Markov decision process (MDP) called the partially observable Markov game. A Markov game with N agents can be defined by a set of states S describing the possible configurations of all agents, a set of actions $A_1, ..., A_N$ and a set of observations $O_1, ..., O_N$. To choose an action, agent i uses a stochastic policy π_{θ_i} : $O_i \times A_i \mapsto [0,1]$, which produces the next state according to the state transition function $P : S \times A_1 \times ... \times A_N \mapsto S$. Agent i obtains a reward $r_i : S \times A_i \mapsto \mathbb{R}$ and receives a private observation O_i : $S \mapsto O_i$. The initial states are determined by the distribution $\rho : S \mapsto [0,1]$. The objective of agent i is to maximize the expected reward $R_i = \sum_{t=0}^{T} \gamma_i r_i^t$, where γ is a discount factor and T is the time horizon. Since the general reward is often sparse in the UAV swarm confrontation environment, it is necessary to design a reasonable reward mechanism for efficient training and fast convergence[7].

2.2 Policy Gradient

Policy gradient (PG) is another popular choice for a variety of reinforcement learning (RL) tasks. The main idea is to adjust the policy parameter θ directly for maximizing the objective $J(\theta) = E_{s \sim p^{\pi}, a \sim \pi_\theta}[\boldsymbol{R}]$ by taking steps in the direction of $\nabla_\theta J(\theta)$. The policy gradient can be written as [3]

$$\nabla_\theta J(\theta) = E_{s \sim p^{\pi}, a \sim \pi_\theta}[\boldsymbol{\nabla_\theta} \log \pi_\theta(a \mid s) Q^{\pi}(s,a)] \tag{1}$$

where p^{π} is the state distribution. The policy gradient idea has given insights to several practical algorithms. For example, a sample reward $R^t = \sum_{i=t}^{T} \gamma^{i-t} r_i$ [5], an approximation of the true action-value function $Q^{\pi}(s,a)$ [4].

However, policy gradient methods often exhibit high variance, especially in the multi-agent learning process. Since in this scene the reward of agent usually depends on the actions of other agents. When the actions of other agents are not considered in the agent's learning process, the variance of gradients will emerge.

3 The Model

3.1 The UAV Kinematic Model

In the UAV swarm confrontation environment, we simplify the UAV motion as a two-dimensional motion with fixed fly height, and the UAV kinematic model can be denoted as

$$\begin{cases} \phi = \phi + r_\phi dt, -30 < \phi < 30 \\ r_\varphi = (9.81 \cdot m / (F \cdot dt)) \cdot \tan \phi \\ \varphi = \varphi + r_\varphi dt, -180 < \varphi < 180 \\ F_x = F \sin \varphi \\ F_y = F \cos \varphi \\ v_x = (F_x / m - \lambda_x) \cdot dt \\ v_y = (F_y / m - \lambda_y) \cdot dt \\ v_z = 0 \\ x = x + v_x dt \\ y = y + v_y dt \\ z = z + v_z dt \end{cases} \tag{2}$$

where r_ϕ denotes the roll angular velocity; r_φ is the heading angular velocity; ϕ represents the roll angle; φ is the heading angle; F is the driving force; λ_x, λ_y are the drag acceleration; v_x, v_y, v_z are the UAV's speed; x, y, z are the spatial position. The constraints of the angles can be defined as

$$s.t. \begin{cases} \phi = \{-\phi_{\max}, \phi_{\max}\}, & \phi_{\max} = 30^\circ \\ \varphi = \{-\varphi_{\max}, \varphi_{\max}\}, & \varphi_{\max} = 180^\circ \end{cases} \tag{3}$$

where the maximum roll angle is 30° and the maximum heading angle is 180°.

Next, we establish the confrontation rules. The confrontation procedure is divided into three stages: the exploration and discovery, the locking and tracking and the attacking. The exploration and discovery is to scout the enemy UAVs by using radar. The scope of radar scanning and reconnaissance is denoted as a circular area where the UAV is the center of the circle. The locking and tracking is to lock the target and communicate the acquired information with the neighbors. As long as the UAV appears within the range of the signal detection, it can receive information from its friend neighbors. The communication range is the circular area denoted by d_s where the UAV is the center of the circle. According to the target position provided by communication or reconnaissance, UAV can release interference signal. The interference signal range is a circular area where the interference source is the center and the radius is d_{in}. The interference can make opponents' distance calculation wrong. In the attacking stage, a target is selected, and the missile path is planned to hit the target. Every UAV has a dominant area, namely the attacking area, which is represented by the radius d_{ac} and the angle θ_{ac}. On the other hand, every UAV owns a vulnerable area, namely the defending area, which is denoted by the angle θ_{de} and the radius d_{de} (Figure 1).

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on December 01,2022 at 13:44:28 UTC from IEEE Xplore. Restrictions apply.

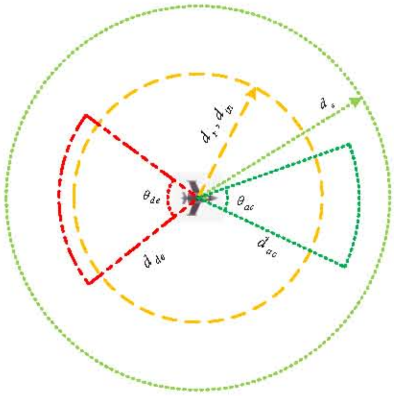

Fig. 1: The schematic diagram of the confrontation ranges

The Euclidean distance between UAV i and UAV j is represented as

$$\begin{aligned} d_{ij} &= \left\| \boldsymbol{p_i} - \boldsymbol{p_j} \right\|_2 \\ &= \sqrt{(\boldsymbol{p_i} - \boldsymbol{p_j})^H (\boldsymbol{p_i} - \boldsymbol{p_j})} \\ &= \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \end{aligned} \tag{4}$$

where p_i is the position of UAV i, and p_j is the position of UAV j. Figure 2 displays the schematic diagram of the confrontation situation of two UAVs.

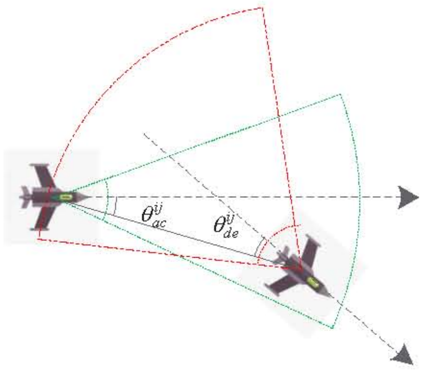

Fig. 2: The schematic diagram of the confrontation situation of two UAVs

The attacking angle θ_{ac}^{ij} and the defending angle θ_{de}^{ij} are calculated as

$$\begin{aligned} \theta_{ac}^{ij} &= \arccos\left(\frac{v_{xi}(x_i - x_j) + v_{yi}(y_i - y_j)}{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} + \sqrt{{v_{xi}}^2 + {v_{yi}}^2}} \right) \\ \theta_{de}^{ij} &= \arccos\left(\frac{v_{xj}(x_i - x_j) + v_{yj}(y_i - y_j)}{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} + \sqrt{{v_{xj}}^2 + {v_{yj}}^2}} \right) \end{aligned} \tag{5}$$

3.2 The MADDPG-based UAV Swarm Confrontation Model

In the multi-agent confrontation game, if each agent learns the environment independently, it will lead to the non-stationary of the learning process and cannot effectively converge to the Nash equilibrium (NE). Each agent's policy changes as the training goes on, and the environment becomes unstable from the perspective of each agent, namely, the changes cannot be well adapted by each agent's own policy. On the other hand, if all agents are combined into a single large agent, then an exponential explosion of action space dimension will take place.

Centralized training decentralized execution (CTDE) is an useful reinforcement learning paradigm. During the training process of the actor-critic framework, an overall evaluation is made by inputting additional global information to the Critic. During the testing process, however, the Critic is not used anymore, only the Actor is used to interact with the environment and make decisions. Consequently, CTDE can effectively avoid non-stationary of the multi-agent environment and the non-convergence of MARL algorithms. By using the mechanism of CTDE, agents can make correct decisions even though only locally observed states are used.

In the algorithm of the multi-agent deep deterministic policy gradient (MADDPG), two networks: the Actor network and the Critic network, will be constructed. The Actor network mainly interacts with the environment for decision-making and execution. While the Critic network estimates the value function according to the global state of the environment and the joint action of agents. Since the output of the Critic network is used for gradient update, the Actor can make more effective decisions.

In order to get independent learning ability, each agent owns two independent Actor and Critic networks. The Actor network obtains the decision action $a_i = f_i(o_i)$ based on the observation state o_i, which can be defined as $\pi_i^\theta(a_i \mid o_i)$, while the Critic network achieves the state-action value from the global observation state and the joint action $Q_i = f_i(\boldsymbol{o}, \boldsymbol{a})$. The network is denoted as $Q_i^\varphi(\boldsymbol{o}, \boldsymbol{a})$, where the multi-layer perceptron (MLP) is used. The framework of MADDPG is plotted in Figure 3.

The objective function of the Actor network is $J_i^\pi(\theta_i) = \mathrm{E}_{o_i \sim \boldsymbol{D}}[\boldsymbol{R}(\boldsymbol{o}, a)]$, and its gradient is defined as

$$\nabla_{\theta_i} J_i^\pi(\theta_i) = \mathrm{E}_{\boldsymbol{o},\boldsymbol{a} \sim \boldsymbol{D}}[\boldsymbol{\nabla}_{\boldsymbol{\theta}_i} \pi(a_i \mid o_i) \nabla_{a_i} Q_i^\varphi(\boldsymbol{o}, \boldsymbol{a})] \tag{6}$$

The objective function of the Critic network is

$$\begin{aligned} J_i^Q(\varphi) &= \mathrm{E}_{\boldsymbol{o},\boldsymbol{a},\boldsymbol{r},\boldsymbol{o}' \sim D}[(\boldsymbol{Q}_i^\varphi(\boldsymbol{o}, \boldsymbol{a}) - y_i)^2] \\ y_i &= r_i + \gamma Q_i^{\bar{\varphi}}(\boldsymbol{o}', \boldsymbol{a}')\big|_{\boldsymbol{a}' = \pi_i^{\bar{\theta}}(a_i \mid o_i')} \end{aligned} \tag{7}$$

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on December 01,2022 at 13:44:28 UTC from IEEE Xplore. Restrictions apply.

3.3 The MASAC-based UAV Swarm Confrontation Model

Although MADDPG can solve the convergence problem when incomplete information exists in the environment, but it also leads to the sensitivity of hyperparameter, poor stability. And the action output of MADDPG only adds Gaussian distribution noise for exploration, thus the exploration ability of deterministic strategy is relatively poor[9]. The single agent soft actor-critic (SAC)[9] is an effective off-policy stochastic policy algorithm, in which the expected reward is the combination of the reward and the policy entropy that measures the uncertainty of policy distribution. On the other hand, by minimizing the divergence between the policy distribution and the Q-value distribution, the agent's policy distribution can approach to the Q-value distribution.

Here, we build an UAV swarm confrontation model with the multi-agent soft actor-critic (MASAC) framework, where the policy entropy is used. By adopting centralized training, the non-stationarity of the environment can be quickly reduced in the early training stage, and the environment can be stabilized and will converge quickly in the later stage. When two confrontation players are not equally rational, MASAC can find better strategies than the NE. During the testing stage, based on the partial observation information, the learning can approach to the NE.

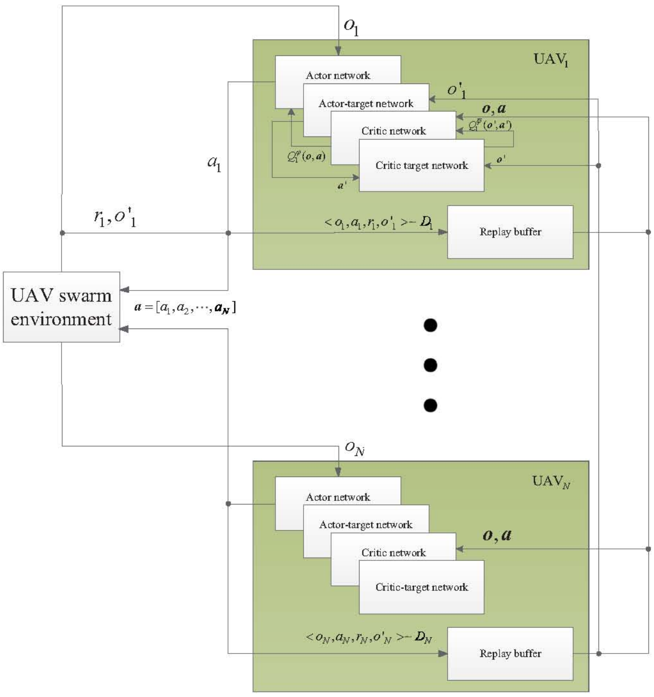

Fig. 3: The framework of the MADDPG-based UAV swarm confrontation

MASAC maximizes both the expected reward and the policy entropy. Since the policy distribution is close to the global Q value distribution, the policy not only considers the agent's state, but also takes into account the behavior of other agents. The divergence of the two distributions is calculated as

$$\pi_i' = \arg\min_{\pi_i \in \Pi} D_{KL}\left(\pi_i(\cdot \mid o_{i,t}) \,\|\, \frac{\exp(\frac{1}{\alpha}) Q_i^{soft}(\boldsymbol{o}_t, \cdot)}{Z_i^{soft}(\boldsymbol{o}_t)} \right) \quad (8)$$

where α is the temperature coefficient of the entropy.

Here, the objective function of the Actor is $J_i^{\pi}(\theta_i) = \mathrm{E}_{o_i \sim \boldsymbol{D}}[\boldsymbol{R_i}(o,a) + \alpha H(\pi_i^{\theta}(\cdot \mid o_i))]$, and its gradient is

$$\begin{aligned}
&\nabla_{\theta_i} J_i^{\pi}(\theta) \\
&= \mathrm{E}_{\boldsymbol{o},\boldsymbol{a} \sim \boldsymbol{D}}[\boldsymbol{\alpha} \boldsymbol{\nabla}_{\theta_i} \log \pi(a_i \mid o_i) \\
&+ \nabla_{\theta_i} f_{\theta_i}(\varepsilon_i; o_i)(\nabla_{a_i} \hat{Q}_i^{\varphi}(\boldsymbol{o},\boldsymbol{a})\Big|_{a_i = f_{\theta_i}(\varepsilon_i; o_i)} - \alpha \nabla_{\theta_i} \log \pi(a_i \mid o_i))]
\end{aligned}$$

Two Critic networks are constructed to calculate two Q values, and the smallest Q value is passed to the Actor network to reduce the variance of the policy network. The objective function of the Critic is

$$\begin{aligned}
&J_i^{Q_j}(\varphi_j) = \mathrm{E}_{\boldsymbol{o},\boldsymbol{a},\boldsymbol{r},\boldsymbol{o}' \sim D}[(\boldsymbol{Q}_i^{\varphi_j}(\boldsymbol{o},\boldsymbol{a}) - y_i)^2] \\
&y_i = r_i + \gamma\left(\hat{Q}_i^{\bar{\varphi}}(\boldsymbol{o}',\boldsymbol{a}') - \alpha \log \pi(a'_i \mid o_i')\right)_{a_i' = \pi_i^{\bar{\theta}}(a_i' \mid o_i')} \\
&\qquad for\ j = 1,2
\end{aligned}$$

where $\hat{Q}_i^{\varphi}(\boldsymbol{o},\boldsymbol{a}) = \min_{j=1,2}\left(Q_i^{\varphi_j}(\boldsymbol{o},\boldsymbol{a})\right)$.

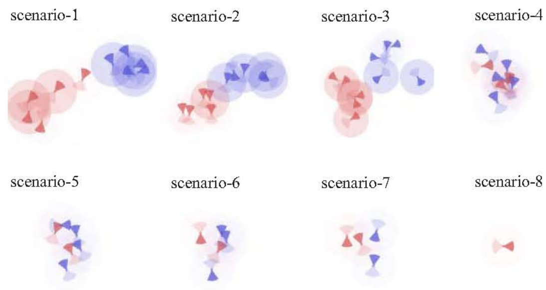

Fig. 4: The confrontation process with respect to 5 red UAVs vs. 5 blue UAVs

4 Simulation Results and Discussion

To verify the UAV swarm confrontation model based on MADRL, we construct a 5 vs. 5 adversarial environment, in which 5 red UAVs combat with 5 blue UAVs and UAVs in the same team can communicate freely. As shown in Figure 4, the confrontation process is divided into 8 scenarios, where UAVs take actions according to MADDPG algorithm. Scenario-1 shows the initial positions (randomly generated) of all red and blue UAVs. In scenario-2 and 3, UAVs explore the battlefield to obtain the confrontation situation of both

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on December 01,2022 at 13:44:28 UTC from IEEE Xplore. Restrictions apply.

sides. From scenario-4 to scenario-7, each UAV executes its strategy to win the battle. The final situation of the confrontation is displayed in scenario-8. The hyperparameters of the MADDPG-based model are shown in Table 1, where the max step refers to the maximum number of the confrontation steps.

Table 1: The hyperparameters of the MADDPG-based model

Hyperparameters	Values
Red number	5
Blue number	5
Learning rate	Actor=0.01, Critic=0.001
Optimizer	Adam
discount factor	0.95
soft update factor	0.99
Replay Buffer size	100w
Number of hidden layers	2
Number of units	128
Batch size	512
Save step	500
Max step	100
Max episode	6k
Soft update step	200

To investigate the performance of the MADDPG-based UAV swarm confrontation model, we draws the relationship between the mean reward and the episode, where red and blue UAVs all use MADDPG algorithm (Figure 5). Here, the mean reward is the average value of the total reward of red and blue teams. One can see that the mean reward increases as the increment of the episode and approaches to zero when the value of the episode is large, which implies that UAVs can be effectively trained. Consequently, by utilizing the information of other UAVs, UAV can make effective decisions under the MADDPG-based UAV swarm confrontation model.

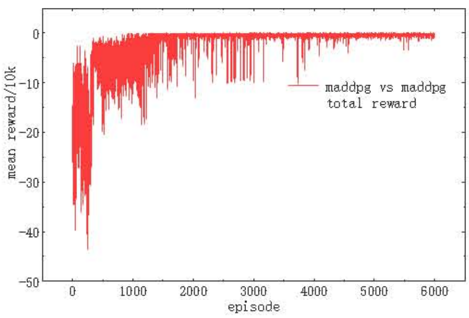

Fig. 5: The mean reward of two teams as a function of the episode under the MADDPG-based Model, where red and blue UAVs all use MADDPG algorithm

The win rate is an important indicator for UAV swarm confrontation. To further investigate the performance of the MADDPG-based Model, we plot the relationship between the win rate and the episode in Figure 6. It shows that, with the increase of the number of episode, the win rate value becomes more and more stable, which means that in the later stage of the training both red and blue UAVs get their own optimal responses, reflecting that the Nash equilibrium of the game is obtained. On the other hand, in the middle stage of the training, the blue team noticed that the red team has learned a good strategy to improve the win rate, and the blue team will learn a countermeasure strategy accordingly. And thus the win rate of the red team is inhibited in the subsequent training stages.

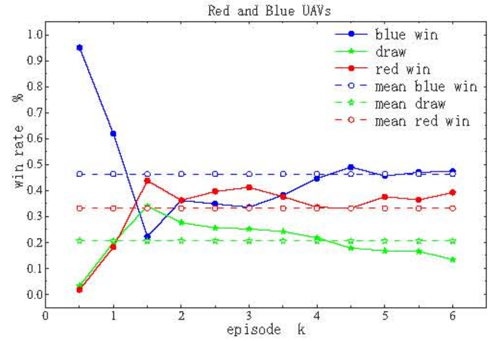

Fig. 6: The win rate as a function of the episode with respect to red and blue UAVs under the MADDPG-based Model

Next, we investigate the effect of the MASAC-based UAV swarm confrontation model. Figure 7 shows the relationship between the mean reward and the episode, where red and blue UAVs all use MASAC algorithm. Here, the mean reward is also the average value of the total reward of two UAV teams. It displays that the mean reward value is close to convergence in the initial stage of the training, indicating that UAVs learn an effective strategy quickly. Figure 8 draws the relationship between the win rate and the episode. This shows that the win rate of the blue team is increased while the red team win rate is decreased. This means that the blue team has learned good strategy to against the red team. On the other hand, although the reward is converged in the initial stage, the red team is still suppressed by the blue team and cannot learn corresponding countermeasures. The red team only maintains a relatively low value of the win rate. The reward and the win rate for both the red and blue teams are relatively stable where the value of the episode larger than 3000, illustrating that the non-cooperative game has entered an approximate equilibrium.

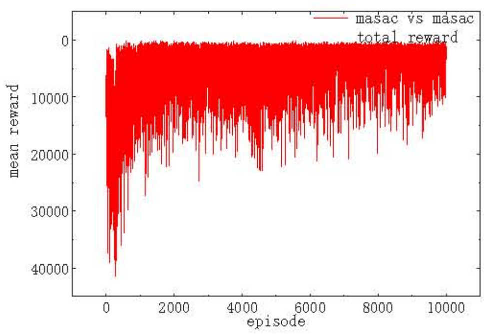

Fig. 7: The mean reward of two teams as a function of the episode under the MASAC-based Model, where red and blue UAVs all use MASAC algorithm

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on December 01,2022 at 13:44:28 UTC from IEEE Xplore. Restrictions apply.

Obviously, for the MADDPG-based model, the blue UAV team cannot effectively maintain its high win rate in the initial stage, and is suppressed by the red UAV team in the later stage. However, in the case of the MASAC-based model, the blue team can learn a number of effective strategies quickly in the initial stage, and will maintain a high value of the win rate. Consequently, the game equilibrium of the blue team under the MASAC-based model is better than that of the MADDPG-based model.

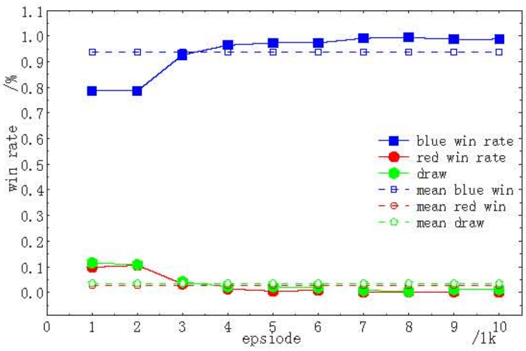

Fig. 8: The win rate as a function of the episode with respect to red and blue UAVs under the MASAC-based Model

5 Conclusions

To summarize, we have introduced the multi-agent deep reinforcement learning (MADRL) paradigm into the confrontation scene of Unmanned Aerial Vehicle (UAV) swarm. Based on the multi-agent deep deterministic policy gradient (MADDPG) algorithm and the multi-agent soft actor-critic (MASAC) algorithm, we construct two UAV swarm non-cooperative game models: the MADDPG-based model and the MASAC-based model. We construct an UAV swarm adversarial environment, in which 5 red UAVs combat with 5 blue UAVs. The results indicate that all two MADRL-based game models converge to the equilibrium, and the performance of the MASAC-based model is better than that of the MADDPG-based model.

In the further work, we will study more effective MADRL-based models to improve the win rate of UAV swarm under large-scale and multi-type confrontation game scenes.

References

[1] L. Busoniu, R. Babuska and B. De Schutter, A Comprehensive Survey of Multiagent Reinforcement Learning, *IEEE Transactions on Systems, Man, and Cybernetics*, Part C (Applications and Reviews), 38(2): 156 172, 2008, doi: 10.1109/TSMCC.2007.913919.

[2] Chao Y , Zhang M , F Ren, et al. Multiagent Learning of Coordination in Loosely Coupled Multiagent Systems. *IEEE Transactions on Cybernetics*, 45(12): 2853 2867, 2015.

[3] Sutton R S, McAllester D, Singh S, et al. Policy gradient methods for reinforcement learning with function approximation. in *Neural information processing systems. NIPS*, 2000: 1057 1063.

[4] Sutton R S, Barto A G. Reinforcement learning: An introduction[M]. MIT press, 2018. 31.

[5] Williams R J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. Advances *in Machine learning*, 8(3): 229 256, 1992.

[6] Garau-Luis J J, Crawley E, Cameron B. Evaluating the progress of Deep Reinforcement Learning in the real world: aligning domain-agnostic and domain-specific research. arXiv preprint arXiv:2107.03015, 2021.

[7] Silva M A L, de Souza S R, Souza M J F, et al. A reinforcement learning-based multi-agent framework applied for solving routing and scheduling problems. *Expert Systems with Applications*, 131(10): 148 171, 2019.

[8] Tang Z, Shao K, Zhao D, et al. Recent progress of deep reinforcement learning: from AlphaGo to AlphaGo Zero. *Control Theory & Applications*, 34(12): 1529 1546 2017.

[9] Haarnoja T, Zhou A, Abbeel P, et al. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor, in *International conference on machine learning. ICML*, 2018: 1861 1870.

[10] Chen Y F, Liu M, Everett M, et al. Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning, in *International conference on robotics and automation . ICRA*, 2017: 285 292.

[11] Mnih V, Badia A P, Mirza M, et al. Asynchronous methods for deep reinforcement learning, in *International conference on machine learning. ICML*, 2016: 1928 1937.

[12] Lillicrap T P, Hunt J J, Pritzel A , et al. Continuous control with deep reinforcement learning. in *International conference on robotics and automation . ICRA*, 2016: 1 8.

[13] Lowe R, Wu Y I, Tamar A, et al. Multi-agent actor-critic for mixed cooperative-competitive environments, in *Neural information processing systems. NIPS*, 2017: 6379 6390.

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on December 01,2022 at 13:44:28 UTC from IEEE Xplore. Restrictions apply.