International Journal of Trend in Research and Development, Volume 9(1), ISSN: 2394-9333
www.ijtrd.com

Research on UAV Target Tracking based on Deep Reinforcement Learning

[1]Suixin Shen, [2]Melnikov S.X and [3]Song Gao,
[1,3]Xi'an Technological University, Shaanxi, China
[2]Belarusian State University of Informatics and Radioelectronics, Minsk, Belarus

***Abstract*—**As the battlefield environment becomes more and more complex, it is of great significance to study the game process of UAV to understand battlefield behavior. Therefore, the search for a UAV movement strategy has become the focus of research. In addition to the traditional strategy, deep reinforcement learning as a decision algorithm with self-learning ability has attracted much attention. In this regard, for the target tracking task of our UAV to the enemy UAV, deep reinforcement learning is used to train the tracking strategy for our UAV. In order to find the most suitable deep reinforcement learning algorithm for UAV target tracking, a target tracking model was established, and the four algorithms were used for training, and the indexes of online training and the results of offline execution were compared. Finally, Dueling Double Deep q-learning and Proximal Policy Optimization achieved the best training effect and completed the target tracking task.

Keywords: *Deep Reinforcement Learning; Target tracking; Machine Learning;*

I. INTRODUCTION

In the battlefield environment, we can build our battlefield monitoring area through UAV to maintain the continuous tracking and monitoring of the enemy target [1], which can effectively prevent the enemy's fighting intention and protect the safety of our combat equipment. However, the actual combat environment is very complex, the enemy aircraft may escape from our monitoring area through maneuvering. In order to ensure the continuous tracking of our UAV to the enemy aircrafts, we use the deep reinforcement learning algorithm to train the tracking strategy of UAV. There are many algorithms in deep reinforcement learning, including Dueling Double Deep q-learning (D3QN) [2], Proximal Policy Optimization (PPO) [3], flexible executor-evaluator algorithm (SAC) [4], depth determination strategy gradient (DDPG) [5], etc. Different algorithms have different effects on UAV target tracking tasks.

II. RESERCH CONTENTS

A. Target tracking model analysis

The single UAV target tracking task can be described as: OUR UAV seeks safe movement strategy for keeping the moving target within the detection range. The above problems can be modeled as Markov decision process and trained by deep reinforcement learning [6]. This section is divided into two parts: the establishment of Markov decision process model and the design of training method.

The Markov decision process contains 5 tuple which is $<\mathrm{S},\mathrm{A},\mathrm{P},\mathrm{R},\gamma>$. Where, S,A are state space and action space respectively, representing the state and action of the defending UAV. P Is the transfer density function, which represents the probability that the body will transfer to the next state when performing an action in the current state. R Is the reward function, and represents the reward that can be obtained when the UAV is in the current state. $\gamma \in [0,1]$ is the discount factor, indicating the degree of attention to long-term returns. The purpose of Markov decision process is to find the optimal strategy for decision-making. Strategy is defined as $\pi = \pi(a \mid s)$, indicating that the probability of executing action *A* according to strategy when the current UAV cluster is in state *S*. Since strategy is the final result and γ is a hyperparameter. So the elements needed to be defined are $\mathrm{S},\mathrm{A},\mathrm{R}$ which are state, action and reward.

B. Definition of state, action, and reward

This chapter studies the target tracking of single agent in reinforcement learning, which will serve as the basis of multi-agent target tracking in the following chapters, and find out the reinforcement learning method, appropriate reward function, and the definition of state and action that are more suitable for this task, so as to complete the training of agents.

When reinforcement learning adopts DDPG, SAC, PPO and other continuous action space methods, A = (*dx*, *dy*), the dimension of action space is 2, its value is continuous, (*dx*, *dy*) represents the horizontal and vertical coordinates of the displacement of the agent within a moment.

When DQN, D3QN and other discrete action space methods are adopted in reinforcement learning, action needs to be defined as discrete quantity. Therefore, the action is defined as 13 dimensions, that is $\mathrm{A} = (a_0, a_1, \cdots, a_{12})$, 13 actions numbered from 0 to 12 represent 13 kinds of displacement in four directions, Figure 1 shows that the continuous space is discretized.

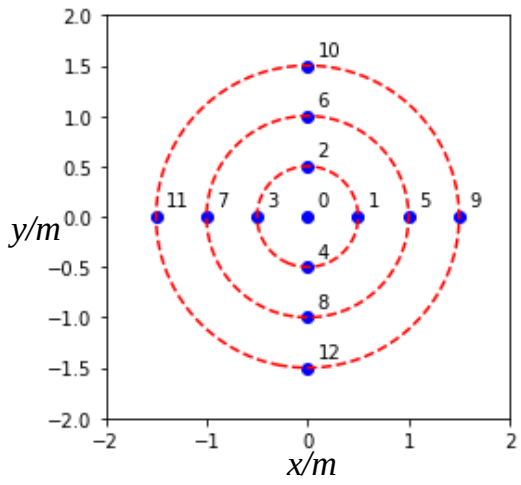


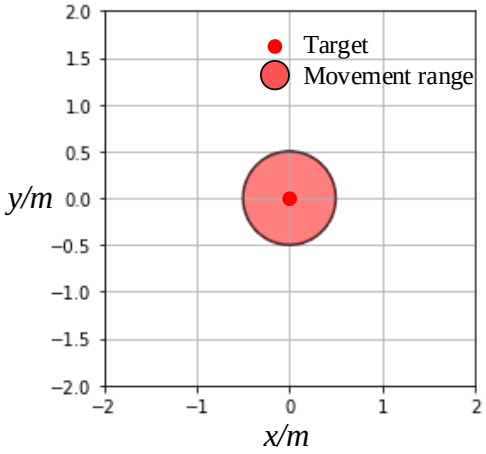
Figure 1: UAV discrete action diagram

This work was supported by the Shaanxi Key Research and Development Program under Grant 2019KWZ-10, 2020GY-194.

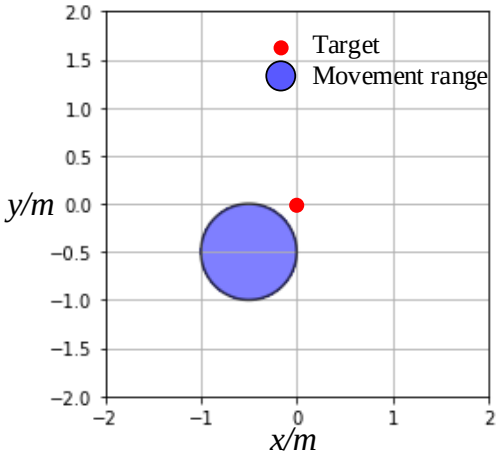
Assuming that the target is moving randomly in a two-dimensional plane, the agent needs to keep track of the target without getting too close to the target and causing the risk of collision. According to the above situation, the agent's state, action and reward function can be defined, where, $S=(Dx,Dy)$, Dx and Dy are the relative horizontal and vertical coordinates of the tracking target and the agent respectively; R is defined as the following formula:

$$R=\begin{cases}0 & 0\le\sqrt{dx^2+dy^2}<3\\ 100 & 3\le\sqrt{dx^2+dy^2}<20\\ -10 & \sqrt{dx^2+dy^2}\ge 20\end{cases} \quad (1)$$

Before training, the environment needs to be initialized. The initialization method will affect the training process to some extent. In the scenario here, the state is initialized as $S_0=(0,0)$, that is, the relative position of the agent and the target is 0, within the danger radius. The reason for this is that an agent can experience as many states as possible during the learning process.



(a) Movement patterns α



(b) Movement patterns β

Figure 2 Movement patterns of target

During training, the movement of the target should also be designed to traverse various states as much as possible, so the movement of the target is defined as two modes, which are defined as α and β, as shown in the Figure 2:

Where, α and β represent two motion modes of the target respectively, and P(α)=p, P(β)=1-p are the probabilities of the target in different motion modes at each moment. In the Figure 2, the origin is the position of the target at the current time, and the circular field represents the possible position of the target after movement, and its probability follows the uniform distribution within the region.

C. Network configuration

In deep reinforcement learning, it is necessary to constantly use the model to output value functions or actions to realize the interaction between the current agent and the environment. This characteristic requires that the network of deep reinforcement learning must complete training in a short time to obtain new training data, so it can only use a smaller network. Among them, all the networks use MLP network, except the input layer and output layer, the network has two hidden layers, the width of the hidden layer is 512. At this point, you can start training the agent.

The training environment was as follows: Pytorch deep learning library in Python was used. The hardware environment was a desktop host with Intel I9-10920X processor and 32GB memory. The graphics card is a Nvidia GeForce RTX 3090*2 with 24GB of memory.

D. Online training index analysis and off-line test execution

Each reinforcement learning method has different performance when dealing with different tasks. In this section, the above methods are applied to the defined agent, and the performance of each algorithm is compared according to the convergence index of reward function

Figure 3 shows the reward curves for D3QN, PPO, SAC, and DDPG. In order to see the convergence trend of the reward curve, all curves are smoothed and shown as solid lines in various colors, while the highly transparent "burrs" are the raw data. It can be seen that PPO and D3QN can converge to the same level respectively, while SAC and DDPG cannot converge in this task. From the perspective of reward alone, the reward curve of PPO algorithm can approach the theoretical maximum value, and D3QN converges to half of the theoretical value.

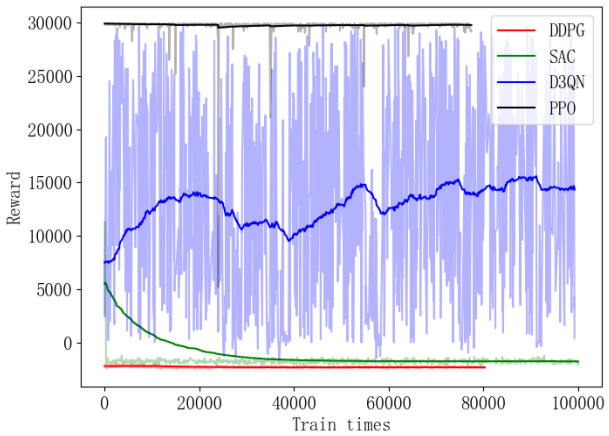


Figure 3: Reward curve of the DDPG, SAC, D3QN and PPO

After the training, the strategy network and Q network obtained by PPO and D3QN were tested respectively. The testing environment was different from the training environment, that is, the strategy of linear movement and random sharp turn to get rid of the tracked target. The test results are shown in the Figure 4, in which red is the target movement track and blue is the UAV track respectively in D3QN. It shows that the UAV can follow the target on the premise of maintaining a safe distance after learning the D3QN algorithm.

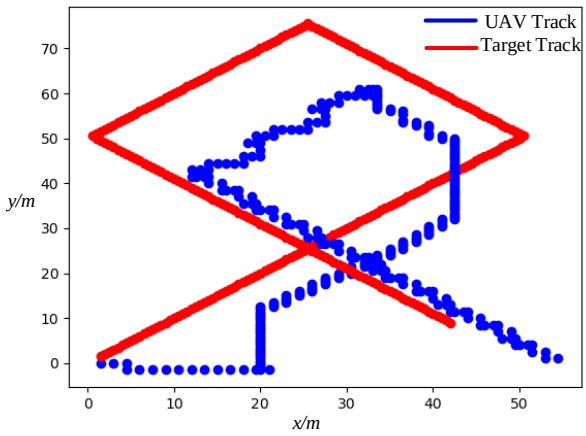


Figure 4 The result of D3QN target tracking

CONCLUSION

A single UAV target tracking problem is studied to find a suitable deep reinforcement learning algorithm for UAV target tracking. After the establishment of the tracking model, D3QN, DDPG, SAC and PPO are firstly used to train the target tracking task of a single UAV. The reward curves of the four algorithms in the online training stage and the target tracking effect of the offline execution are counted. Finally, the reward curves of the training process and the task execution results are compared. DDPG and SAC algorithms failed to acquire strategies during training. The reward curve of PPO converges close to the theoretical maximum value, and the target tracking loss rate is close to 0% in the implementation stage. D3QN failed to converge to the maximum value, but target tracking could be completed at a low target loss rate in the execution stage. The effectiveness of PPO and D3QN algorithm is verified.

References

[1] Wenhong Z, Jie L I, Zhihong L I U, et al. Improving multi-target cooperative tracking guidance for UAV swarms using multi-agent reinforcement learning[J]. Chinese Journal of Aeronautics, 2021.

[2] Xie L, Wang S, Markham A, et al. Towards monocular vision based obstacle avoidance through deep reinforcement learning[J]. arXiv preprint arXiv:1706.09829, 2017.

[3] Schulman J, Wolski F, Dhariwal P, et al. Proximal policy optimization algorithms[J]. arXiv preprint arXiv:1707.06347, 2017.

[4] Haarnoja T, Zhou A, Abbeel P, et al. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor[C]//International Conference on Machine Learning. PMLR, 2018: 1861-1870.

[5] Lillicrap T P, Hunt J J, Pritzel A, et al. Continuous control with deep reinforcement learning[J]. arXiv preprint arXiv:1509.02971, 2015.

[6] Howard R A. Dynamic programming and markov processes[J]. 1960.