IOP Conf. Series: Journal of Physics: Conf. Series **1487** (2020) 012006 doi:10.1088/1742-6596/1487/1/012006

A RDA-Based Deep Reinforcement Learning Approach for Autonomous Motion Planning of UAV in Dynamic Unknown Environments

Kaifang WAN[1,2*], Xiaoguang GAO[1], Zijian HU[1] and Wei ZHANG[2]

[1] School of Electronic and Information, Northwestern Polytechnical University, Xi'an 710072, China
[2] Science and Technology on Electronic Information Control Laboratory, Chengdu 610036, China

E-mail: wankaifang@nwpu.edu.cn

Abstract. Autonomous motion planning (AMP) in dynamic unknown environments emerges as an urgent requirement with the prosperity of unmanned aerial vehicle (UAV). In this paper, we present a DRL-based planning framework to address the AMP problem, which is applicable in both military and civilian fields. To maintain learning efficiency, a novel reward difference amplifying (RDA) scheme is proposed to reshape the conventional reward functions and is introduced into state-of-the-art DRLs to constructs novel DRL algorithms for the planner's learning. Different from conventional motion planning approaches, our DRL-based methods provide an end-to-end control for UAV, which directly maps the raw sensory measurements into high-level control signals. The training and testing experiments demonstrate that our RDA scheme makes great contributions to the performance improvement and provides the UAV good adaptability to dynamic environments.

1. Introduction

Over the past few years, there has been an uptrend of UAVs applied in a wide range of practical missions, such as intelligence, surveillance, and reconnaissance (ISR) [1], suppression of enemy air defences (SEAD) [2], search and rescue [3], and goods delivery [4]. One key requirement among such applications is how to build an intelligent system for UAV to carry out tasks autonomously without any human interventions [5][6]. To be specific, it is essential that we should develop advanced intelligent techniques to autonomously navigate a UAV from arbitrary departures to destinations while avoiding threats in dynamic unknown environments. To realize this attractive envision, there are at least two challenges stand there: 1) Partial observability of environment. UAV knows nothing about the environment at the beginning and only partial information can be sensed during the task. This features turns some rule-based path planning methods [7]-[8] unavailable, because it is impossible to design complete rules for all possible situations while facing with uncertain environments. 2) Unpredictability of environment. The irregular mobility of the scattered objects brings UAV an unstable environment that the SLAM-based navigation methods [9] will become intractable, because the mobile objects require a continuous mapping and this will lead to unaffordable computation costs. Besides, an open-loop mechanism of the sensing-planning-based methods [10] that makes decisions without any prediction and reasoning of the future, blocks their adaptations to dynamic environments.

To address these challenges, researchers have resorted to reinforcement learning (RL) techniques and committed themselves to design learning-based planners for UAVs [11]-[15].

As a kind of machine learning algorithms, RL is often used to solve sequential decision making problems and is deeply connected to adaptive dynamic programming (ADP) [16]. The special scheme of RL enables it to learn an intelligent planner by trial-and-error interacting with the environment. The RL-based planner uses Markov decision processes (MDP) [17] to model problems and produce strategies based on predicted long-term rewards, which makes it adapt to stochastic dynamic environments without knowing the system model . However, the problems of curse of dimension prevent the conventional RL algorithms being further applied. To address curse of dimension and maintain a better representation of the high-dimensional continuous state space, deep neural network is introduced into the conventional RL and produces deep reinforcement learning (DRL) method. By leveraging the representative capabilities of DL and the decision-making capabilities of RL, DRLs have achieved outstanding performance in the field of UAV motion planning [18]-[22]. Loquercio [18] provides a powerful deep neural network DroNet that can directly map the images ahead of the UAV to a desired yaw angle and an desired forward speed of UAV. This end-to-end solution makes the UAV achieve the purpose of autonomous navigation. Kersandt [19] realizes and compares the effects of some value-based DRL, such as DQN [20], Double DQN [21] and Dueling DQN [22] in the same UAV navigation mission.

Although some encouraging achievements have been made in aforementioned studies, there are few researches on applications of DRL in dynamic unknown environments, which are often encountered in practice. In fact, challenges arise when we consider UAV navigation problems in dynamic unknown environments. The ubiquitous mobile threats and targets increase the difficulty for training an available and stable planner that can adapt dynamic environment. In this paper, we treat the UAV autonomous motion planning problem as a higher-level discrete control problem and solve it in a DRL framework. Considering such practical applications are sensitive to reward signals, we propose a novel reward-shaping scheme by amplifying the differences between the current and last rewards. This reward difference amplifying (RDA) approach comes from the phenomenon of psychology [23] and is introduced into state-of-the-art DRLs to constructs novel DRL algorithms for the planner's learning. To demonstrate the performance of the proposed DRL-based planner, a general simulation platform is constructed and a series experiments is conducted in our work. The details will be found following sections.

The reminder of this manuscript is organized as follows. Section 2 presents the autonomous motion planning problem formulation. Section 3 introduces the whole solution of the problem and derives our RDA-based learning approaches. Section 4 validates our method through a series of experiments. Section IV concludes this paper and envisages our future work.

2. Problem formulation

2.1. Problem description

Considering an autopilot will provide the UAV low-level flight control in a fast-inner loop and ensure it fly in a certain attitude, heading, altitude and Mach number [24]. We adopt the kinematics as a substitute for UAV's dynamics and emphasize the high-level control of the UAV only. The assumption of a fixed altitude and constant velocity leaves the angular velocity of heading as the only control command. It is reasonable in many realistic cases and this simplification allows us to focus more on the motion planning algorithm [25]. Then the equations of motion of the UAV read:

$$\dot{\boldsymbol{\xi}}_{\boldsymbol{u}} = \frac{d}{dt}\begin{pmatrix} x_u \\ y_u \\ \psi_u \end{pmatrix} = \begin{pmatrix} v_u \cos\psi_u + \eta_x \\ v_u \sin\psi_u + \eta_y \\ -(g/v_u)\tan\phi_u + \eta_\theta \end{pmatrix} \tag{1}$$

where $\boldsymbol{p}_u := (x_u, y_u) \in \mathbf{R}^2$ is the planar position, $\psi_u \in \mathbf{S}^1$ is the heading angle and $\phi_u \in \mathbf{S}^1$ is the roll angle of the UAV. g denotes the acceleration due to gravity and v_u is the linear velocity of the UAV. The control command $\boldsymbol{u}$ is defined by the roll angle ϕ_u and the UAV state is represented by vector $\boldsymbol{\xi}_{\boldsymbol{u}}$

$:= [x_u, y_u, \psi_u, \phi_u]^T$. In addition, the model takes the disturbance terms $\eta_x, \eta_y, \eta_\theta$ into account, which are drawn from normal distributions $N(0, \sigma_x^2)$, $N(0, \sigma_y^2)$, and $N(0, \sigma_\theta^2)$, respectively. According to equation (1), a well-designed ϕ_u will makes the UAV motion stably.

2.2. Problem modeling

In this paper, we focus on learning-based approaches. More specifically, a RL technique is used to design the planner for the UAV. RL uses a Markov decision process to model the sensing-acting cycle of the agent. At each epoch t, the agent perceives the system state $\boldsymbol{s_t}$ and selects an action $\boldsymbol{a_t}$. Then the selected action is taken and the system state moves to $\boldsymbol{s_{t+1}}$ with a reward signal r_t return to the agent. Differing from traditional methods that make decisions relying only on the one-step reward r_t, RL selects optimal action by maximizing an expected long-term reward Q^*, which gives the agent abilities of addressing dynamic and uncertain inputs. For a given state-action pair $(\boldsymbol{s_t}, \boldsymbol{a_t})$, the expected long-term reward $Q(\boldsymbol{s_t}, \boldsymbol{a_t})$ can be defined by the following equation:

$$Q(\boldsymbol{s_t}, \boldsymbol{a_t}) = \mathbf{E}\left(\sum_{l=0}^{H} \gamma^l r(\boldsymbol{s_{t+l}}, \boldsymbol{a_{t+l}}) | \boldsymbol{s_t}, \boldsymbol{a_t}\right) = \mathbf{E}_{s' \sim \boldsymbol{P}(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a})} \left(r(\boldsymbol{s}') + \gamma \max_{\boldsymbol{a}'} Q(\boldsymbol{s}', \boldsymbol{a}')\right) \quad (2)$$

where $\boldsymbol{P}(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a})$ represents the state transition probability (i.e., from $\boldsymbol{s}$ to $\boldsymbol{s}'$ due to action $\boldsymbol{a}$), H is the horizon of the prediction, and γ is the discount factor belonging to (0,1]. Subsequently, the optimal action can be determined by:

$$\pi^*(\boldsymbol{s_t}) = \arg\max_{\boldsymbol{a}} Q(\boldsymbol{s_t}, \boldsymbol{a_t}) \quad (3)$$

In our scenario, the system state $\boldsymbol{s}$ consists of UAV state $\boldsymbol{\xi_u} = [x_u, y_u, \psi_u, \phi_u]^T$, target state $\boldsymbol{\xi_T} = [x_T, y_T]^T$ and environment state $\boldsymbol{\xi_e}$, where $\boldsymbol{\xi_e}$ is a N_r -dimension vector $\boldsymbol{\xi_e} = [d_1, d_2, \dots, d_{N_r}]^T$ representing the detected threats relative distances by a N_r-rays LiDAR sensor (see in figure 1). The action is defined as the desired turn direction, where $\boldsymbol{a}$ = -1, 0, or 1 means the suggestion is to turn left, go straight or turn right, respectively (see in figure 2).

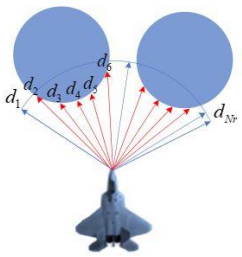

Figure 1. UAV sensed data.

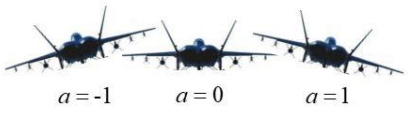

Figure 2. UAV actions.

Once an optimal action a_t^* is selected, it will be converted to roll angle by $\phi_{u,t} = a_t^* * \Delta\phi_u$, where the $\Delta\phi_u$ is the roll angle that can be rotated by performing a single action. Then the control command for UAV kinematics can be

$$u^* = \min\left(\phi_{u,\max}, |a_t^* * \Delta\phi_u|\right) \quad (4)$$

where $\phi_{u,\max}$ sets a upper bound for the roll angle.

3. Methodology

3.1. DRL-based planning architecture

Since $Q(\boldsymbol{s}, \boldsymbol{a})$ in equation (3) is intractable, we use DRL to provide an approximate solution. DRL uses a neural network $Q(\boldsymbol{s}, \boldsymbol{a}; \boldsymbol{\theta})$ in a critic module to fit $Q(\boldsymbol{s}, \boldsymbol{a})$ and an actor module selects the optimal action $\boldsymbol{a}^*$ through $\arg\max_{\boldsymbol{a}} Q(\boldsymbol{s}, \boldsymbol{a}; \boldsymbol{\theta})$ or ϵ-greedy. In a standard Deep Q-leaning (DQN) [20], temporal difference algorithm is adopted and a gradient descent step is executed for training the parameterized critic network. An experience replay strategy and a fixed target network strategy are

used to disrupts the correlation between experiences and maintains the learning efficiency. The target network $Q'(\boldsymbol{s},\boldsymbol{a};\boldsymbol{\theta}')$ is copied from the critic network periodically and used for targets generating. Let y denotes the target for a given current sample $(\boldsymbol{s},\boldsymbol{a},\boldsymbol{r},\boldsymbol{s}')$:

$$y(\boldsymbol{s},\boldsymbol{a}) = r(\boldsymbol{s}') + \gamma \max_{\boldsymbol{a}'} Q'(\boldsymbol{s}',\boldsymbol{a}';\boldsymbol{\theta}') \tag{5}$$

where $Q'(\boldsymbol{s}',\boldsymbol{a}';\boldsymbol{\theta}')$ is the target network and the parameters $\boldsymbol{\theta}'$ are a copy of critic network $Q(\boldsymbol{s},\boldsymbol{a};\boldsymbol{\theta})$ from the moment of last target recomputation. Let's define the MSE loss function as $\text{Loss} = \big(Q(\boldsymbol{s},\boldsymbol{a};\boldsymbol{\theta}) - y(\boldsymbol{s},\boldsymbol{a})\big)^2$ and the parameters of critic network can be updated by executing a gradient descent step:

$$\boldsymbol{\theta_{t+1}} = \boldsymbol{\theta_t} + \alpha_t\big(y(\boldsymbol{s},\boldsymbol{a}) - Q(\boldsymbol{s},\boldsymbol{a};\boldsymbol{\theta_t})\big)\nabla_\theta Q(\boldsymbol{s},\boldsymbol{a};\boldsymbol{\theta}) \tag{6}$$

3.2. RDA-based learning approach

Experience replay and fixed target network make it practical for DRL in solving UAV's motion planning problem. To enhance the performance while ensuring convergence, we propose a reward difference amplifying ideas in the learning process.

3.2.1. Reward function design. As we all know, rewards are the only feedback signals that can be used for the agent's learning. A well-shaped reward function usually contains as much human experience information as possible that will provide the agent a better learning performance.

Sparse reward is the most commonly used scheme in some benchmark environments that provides only fixed signals when some conditions meet. In our mission, if the UAV reaches the target, the agent will receive a positive reward r_a; if the UAV collides with some threats, it will receive a negative penalty r_b; otherwise zero will set to reward.

$$r_{sp}(\boldsymbol{s},\boldsymbol{a}) = \begin{cases} r_a & if\ arrive\ the\ target \\ r_b & if\ collide \\ 0 & every\ step \end{cases} \tag{7}$$

Since sparse scheme is inefficient in our scenario because the UAV need to fly lots of steps to successfully avoid threats and reach the target, which brings the agent numerous invalid rewards. To address this problem, an intuitive idea is to add progressive reward signals with full considerations of the characteristics of the motion planning problem, which produces our Intermediate reward.

$$r_{in}(\boldsymbol{s},\boldsymbol{a}) = \begin{cases} r_a & if\ arrive\ the\ target \\ r_b & if\ collide \\ \mu_1\big(D_{ut}^{pre} - D_{ut}^{cur}\big) + \mu_2\left(-\dfrac{\Delta\psi}{4}\right) + \mu_3\left(\dfrac{D_f}{D_s} - 1\right) & every\ step \end{cases} \tag{8}$$

where $D_{ut}^{pre}, D_{ut}^{cur}$ denote the previous and current relative distances between UAV and the target; $\Delta\psi$ denotes the angle of the UAV flight direction deviating from the target; D_s is the detection distance of laser; D_f is the distance of the detected threat in front of the UAV and if there is no threat ahead of it, D_f will be set to D_s. Obviously, these three sub items in equation (8) represent the contributions of distance, angle and threat to the rewards respectively, and the contribution rates can be tuned by μ_1, μ_2, μ_3.

3.2.2. Reward shaping scheme. Moreover, we propose a novel reward shaping method by amplifying the differences between the current and last rewards. This idea derives from the status quo phenomenon in psychology [23], in which a person is proved more likely to take an action that has brought him a better reward in other similar missions. And if he fails unexpectedly near success, he will be cautious while facing with the same situation again, even stay at that state without taking any risk. To model these experiences, we define reward difference $\delta_r = (r_t - r_{t-1})/min\,(r_t, r_{t-1})$ and utilize it as a signal to tune the reward, that is, to amplify the reward for a positive difference and to minify the reward for a negative difference. The transform law is:

$$r_{da}(\delta_r) = \begin{cases} r_t + |r_t|\left(\tan^{-1}\left(\frac{\delta_r + \lambda}{\eta} * \frac{\pi}{2}\right) - \lambda\right) & if\ |\delta_r + \lambda| > \eta \\ r_t & otherwise \end{cases} \quad (9)$$

where r_t, r_{t-1} are the rewards at current and last time step; $\lambda = sgn(r_t - r_{t-1})$ is used to balance the equation that a negative difference brings $\lambda = -1$ and a positive difference brings $\lambda = 1$. η is a tuning parameter that makes the scheme adapt to different missions. Reward difference amplifying scheme can reinforce the utilization of excellent historical experiences and avoid the bad historical experiences by strengthening the extreme reward signals. It is worth mentioning that the reward difference amplifying is just a scheme for adjusting existing reward functions and can be introduced into many DRL algorithms. In addition, this scheme makes the agent draw on advantages and avoid disadvantages in decision-making and makes it easier to find the optimal solution of the problem.

Adding the reward difference amplifying scheme (equation (9)) to classic DQN with original reward functions (equation (7) (8)), we obtain RDA-DRL based motion planning algorithm:

Algorithm 1: RDA-DRL based Motion Planning Algorithm (with DQN)
1: **initialize** Hyperparameters: experience pool ***D,*** batch size ***B***, target network update frequency ***K***, Q network with arbitrary weights $\boldsymbol{\theta}$, target network Q' with weights $\boldsymbol{\theta'} = \boldsymbol{\theta}$, greedy exploration parameter ϵ
2: **repeat** (for each episode)
3: **initialize** $\boldsymbol{s_0} \leftarrow (\boldsymbol{\xi_{u,0}}, \boldsymbol{\xi_{T,0}}, \boldsymbol{\xi_{e,0}})$ randomly;
4: **while** (not target **and** not collide **and** $t < T$) **do**
5: choose $\boldsymbol{a_t}$ randomly with ϵ-greedy or $\boldsymbol{a_t} = \arg\max_{\boldsymbol{a}} Q(\boldsymbol{s_t}, a; \boldsymbol{\theta})$
6: $\boldsymbol{u_t} \leftarrow$Equation(4)
7: execute $\boldsymbol{u_t}$, new system state $\boldsymbol{s_{t+1}} \leftarrow (\boldsymbol{\xi_{u,t+1}}, \boldsymbol{\xi_{T,t+1}}, \boldsymbol{\xi_{e,t+1}})$;
8: **original reward** $\boldsymbol{r_t} \leftarrow$Equation(7) or Equation(8);
9: **difference amplified reward** $\boldsymbol{r'_t} \leftarrow$Equation(9);
10: store $(\boldsymbol{s_t}, \boldsymbol{a_t}, \boldsymbol{r'_t}, \boldsymbol{s_{t+1}})$ in ***D;***
11: sample transitions $[\boldsymbol{s_j}, \boldsymbol{a_j}, \boldsymbol{r'_j}, \boldsymbol{s_{j+1}}]_{j=1\ldots B}$ from ***D;***
12: $\delta_{q,t,j} = \boldsymbol{r'_j} + \gamma \max_{\boldsymbol{a'}} Q'(\boldsymbol{s_{j+1}}, \boldsymbol{a'}; \boldsymbol{\theta'}) - \boldsymbol{Q}(\boldsymbol{s_j}, \boldsymbol{a_j}; \boldsymbol{\theta_t})$;
13: $\boldsymbol{\theta_{t+1}} \leftarrow \boldsymbol{\theta_t} + \alpha_t \boldsymbol{\delta}_{q,t} \nabla_\theta Q(\boldsymbol{s_t}, \boldsymbol{a_t}; \boldsymbol{\theta_t})$;
14: $\boldsymbol{\theta'} \leftarrow \boldsymbol{\theta}$ if t mod $K = 0$;
15: **end while**
16: **until** desired number of episodes

4. Experiments

4.1. Experiment settings

For the model training and testing, we construct a general simulation environment in this paper. The environment simulate a world with a total size of 400×300 m2 and a series of moving threats are randomly scattered in the world. The sensor equipped on the UAV is capable of detecting an area of 40 meters ahead and ±45 degrees from left to right. There are 30 beams distances detection. The velocity of the UAV is set to 15 m/s and the disturbance parameters are set as $\sigma_x = \sigma_y$=0.1, σ_θ=0.01. The roll angle rotating in one action $\Delta\phi_u$ is set to 15°and the maximum roll angle $\phi_{u,\max}$ is 90°.

As for the DRL settings, the critic network is set to own one input layer with 36 nodes (4-dimensions UAV state, 2-dimension target state, 30-dimention sensed information), two hidden layers with 100 nodes for each and one output layer with 3 nodes, which means there are hundreds of thousands parameters have to be tuned. We set the discount factor $\gamma = 0.9$, the batch size B=32, the -experience pool size D=5000 and the target network update frequency K=200. A variable ϵ-greedy policy $\epsilon_t = \max(1 - N_l \Delta\varepsilon, \epsilon_0)$ is used, where N_l is the number of learning that has been carried out,

$\Delta\epsilon$ is the reduction rate with a value of 0.00002 and $\epsilon_0 = 0.0001$ sets a lower bound of ϵ. Besides, the reward parameters are set to r_a=3, r_b=-6, μ_1=0.3, μ_2=0.4, μ_3=0.5 and η=2.

4.2. Experiment results

For comparing, we construct four Agents (as shown in Table 1), where different reward functions and reward shaping scheme are bind to DQN. Agent 1 and Agent 3 come from DQN with a sparse reward and an intermediate reward respectively. Agent 2 and Agent 4 are the improved models with our reward difference amplifying scheme corresponding to Agent 1 and Agent 3. To simulate a dynamic environment, the positions of UAV, target and threats are randomly generated in each episode, and 90% of the threats are set to move with randomly generated velocities obey uniform distribution U(1,5). All the algorithms are implemented with Tensorflow and Agents are trained with a GeForce RTX 2080 GPU in 5000 episodes.

Table 1. Experiments and algorithms

Agent 1	DQN with sparse reward (equation (7))
Agent 2	DQN with sparse reward (equation (7)) and RDA scheme (equation (9))
Agent 3	DQN with intermediate reward (equation (8))
Agent 4	DQN with intermediate reward (equation (8)) and RDA scheme (equation (9))

To evaluate the performance, we design average hit rate as the quantitative indicator. The average hit rate is defined as the probability of the UAV successfully hitting a target in the last 100 episodes. All of agents in Table 1 are trained to in a dynamic environment. The experiment results are illustrated in figure 3.

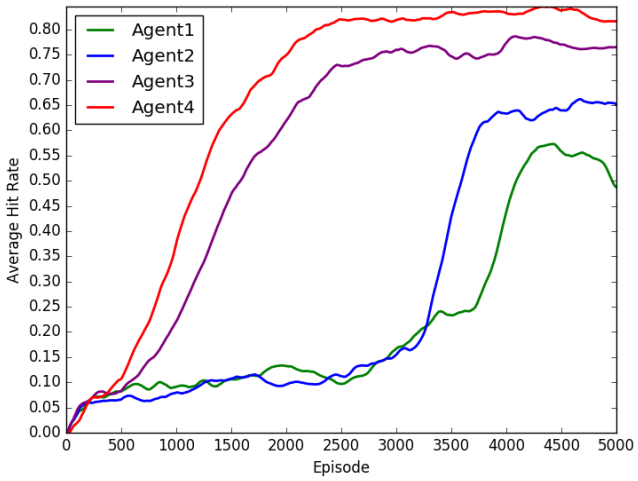

Figure 3. Average hit rate curves of the four different agents.

As we can see in figure 3, agent 4 achieves the fastest convergence, the highest hit rate compared with other three agents. This is because it integrates both intermediate reward and our RDA scheme, which continuously provide the agent more incentive signals to reinforce the learning. Agent 3 works better than agent 1 shows the importance of the intermediate incentive signals when facing with practical applications. If we select the hit rate at final convergence as the indicator, we can see that our RDA scheme brings conventional DRL algorithms higher hit rates. To be specific, RDA brings an effectiveness promotion of 9.22% and 44.44% respectively by comparing agent 4 (0.83) to agent 3 (0.76) and comparing agent 2 (0.65) to agent 1 (0.45).

As the trainings are done, we employ the trained planners to solve the autonomous motion planning problem and evaluate the effect of the proposed algorithms from the perspective of applications. In the testing experiments, we let the four agents start from the same location (-10,180) and travel through the same dynamic environment until they reach the same target (-120,-120). We take three screenshots

for each agent at T=5s, T=10s and the time collision or arrival. All the screenshots can be seen in figure. 4.

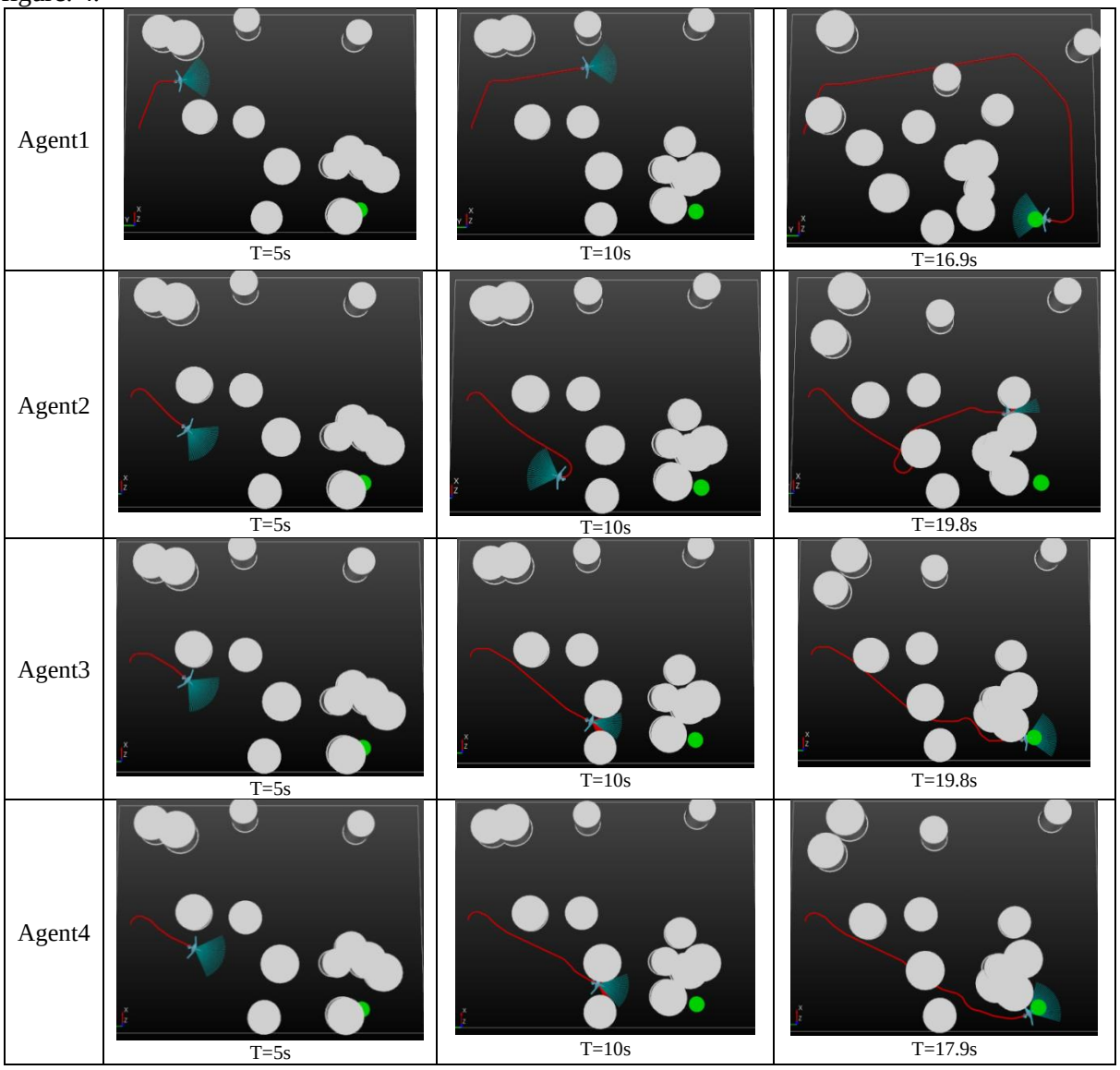

Figure 4. UAV flying trajectories based on the four agents.

From figure 4, we can see that the four trained agents fly out completely different trajectories for the same task. Agent 1 chose to avoid intensive threat areas and fly a big circle until arrives to the target at time 16.9s. Agent 2 chose to directly fly to the target at first and when it finds some threats in the front, it changes its way. But unfortunately, it collides finally at time 19.8s. Agent 3 and Agent 4 choose better paths comparing to Agent 1 and Agent 2, they successfully bypassing threats and fly directly to the targets. However, Agent 4 flies a shorter path and arrives to the target at time 17.9s, while the Agent flies 19.8s. The lengths of flied trajectories of the four Agents are 606m, 396m, 352m and 338m, respectively. Obviously, our RDA-DRL based planner (Agent 4) owns the best performance while solving the autonomous motion planning problem.

5. Conclusion

This paper presents a DRL-based planning framework for UAV to autonomously motion in dynamic unknown environments. To maintain learning efficiency, a novel reward difference amplifying scheme

is proposed to reshape the conventional reward functions. This RDA scheme is introduced into state-of-the-art DRLs to constructs novel DRL algorithms for the planner's learning. The training and testing experiments demonstrate that our RDA scheme makes great contributions to the performance improvement and provides the UAV good adaptability to dynamic environments.

Acknowledgments
This work was supported by Science and Technology on Electronic Information Control Laboratory (No. 2017KD0154)

Reference

[1] Stevens RC, Sadjadi FA. Small unmanned aerial vehicle real-time intelligence, surveillance and reconnaissance (ISR) using onboard pre-processing. *Proceedings of SPIE*, 2008; **6967**(1):1-8.
[2] Darrah M, Niland W, Stolarik B. UAV cooperative task assignments for a SEAD mission using genetic algorithms. *Proceedings of AIAA guidance, navigation, & control conference & exhibit*. 2006 Aug 21-24; Colorado, New York: AIAA; 2012.
[3] Tomic T, Schmid K, Lutz P. Toward a fully autonomous UAV: research platform for indoor and outdoor urban search and rescue. *IEEE Robotics & Automation Magazine*, 2012; **19**(3):46-56.
[4] Shakhatreh H, Sawalmeh A, Al-Fuqaha A. Unmanned aerial vehicles (UAV): a survey on civil applications and key research challenges. *IEEE Access*, 2018, **7**(1):1-58.
[5] Wan K., Gao X., Hu Z et al. Robust motion control for UAV in dynamic uncertain environments using deep reinforcement learning. *Remote sensing*, 2020; **12**(4): 640-661.
[6] Imanberdiyev N, Fu C, Kayacan E. Autonomous navigation of UAV by using real-time model-based reinforcement learning. *Proceedings of 14th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. 2016 Nov. 13-15; Phuket, Thailand; IEEE; 2017.
[7] Chee KY, Zhong ZW. Control, navigation and collision avoidance for an unmanned aerial vehicle. *Sensors and Actuators A: Physical*, 2013, **190**(1):66–76.
[8] Panagou D. A distributed feedback motion planning protocol for multiple unicycle agents of different classes. *IEEE Transactions on Automatic Control*, 2016, **62**(3):1-1.
[9] Gee T, James J, Mark W. Lidar guided stereo simultaneous localization and mapping (slam) for UAV outdoor 3-D scene reconstruction. *Proceedings of International Conference on Image and Vision Computing*, 2016 Nov21-22; Palmerston North, New Zealand; IEEE; 2017.
[10] Liang X, Wang H. Three-dimensional path planning for unmanned aerial vehicles based on principles of stream avoiding obstacles. *Chinese Journal of Aeronautics*, 2013, **34**(7):1670-1681.
[11] Lin X, Yu Y, Sun CY. Supplementary reinforcement learning controller designed for quadrotor UAVs. *IEEE Access*, 2019:1-1.
[12] Hu Z, Wan K, et al. A dynamic adjusting reward function method for deep reinforcement learning with adjustable parameters. *Mathematical Problems in Engineering*, 2019; **2019**(1):1–10.
[13] Zhang B, Mao Z, Liu W, et al. Geometric reinforcement learning for path planning of UAVs. *Journal of Intelligent & Robotic Systems* 2015; **77**(2): 391-409.
[14] Mohammad J, Hao X. Intelligent control for unmanned aerial systems with system uncertainties and disturbances using artificial neural network. *Drones*, 2018, **2**(3):30-37.
[15] Pham H X, La H M, Feil-Seifer D. Reinforcement learning for autonomous uav navigation using function approximation. *2018 IEEE International Symposium on Safety, Security, and Rescue Robotics*; 2018 Aug 6-8; Philadelphia, US. IEEE, 2018: 1-6.
[16] Wan KF, Gao XG, Li B. Using approximate dynamic programming for multi-ESM scheduling to track ground moving targets. *Journal of Systems Engineering and Electronics*, 2018, **29**(1): 74 – 85.
[17] Yang QM, Zhang JD, Shi GQ. Modeling of UAV path planning based on IMM under POMDP framework. *Journal of Systems Engineering and Electronics*, 2019, **30**(3):545-554.
[18] Loquercio A, Maqueda A I, Del-Blanco C R. DroNet: Learning to fly by driving. *IEEE Robotics*

and Automation Letters 2018; **3**(2): 1088-1095.
[19] Kjell K. Deep reinforcement learning as control method for autonomous UAV [dissertation]. Universitat Politecnica de Catalunya; 2017.
[20] Mnih V, Kavukcuoglu K. Human-level control through deep reinforcement learning. *Nature*, 2015,**353**(7540):529–533.
[21] Van HH, Guez A, Silver D. Deep reinforcement learning with double Q-learning. *Proceedings of the 30th AAAI Conference on Artificial Intelligence*. Menlo Park, United States; AAAI; 2016.
[22] Wang Z, Freitas N, Lanctot M. Dueling network architectures for deep reinforcement learning. *Proceedings of the 33rd International Conference on Machine Learning*. New York; ACM; 2016.
[23] Samuelson W, Zeckhauser R, Status quo bias in decision making. Journal of Risk and Uncertainty,1988:7-59
[24] Wu GF, Gao XG, Fu XW. Mobility control of unmanned aerial vehicle as communication relay in airborne multi-user systems. *Chinese Journal of Aeronautics*, 2019, **32**(6): 1520–1529
[25] Dixon C. Controlled mobility of unmanned aircraft chains to optimize network capacity in realistic communication environments [dissertation]. Colorado: University of Colorado; 2010.