ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-7281-6327-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICASSP49357.2023.10096457

UAV LOCAL PATH PLANNING BASED ON IMPROVED PROXIMAL POLICY OPTIMIZATION ALGORITHM

Jiahao Xu[1]*, Xufeng Yan*[1]*, Cui Peng*[3]*, Xinquan Wu*[1]*, Lipeng Gu*[1]*, Yanbiao Niu*[1]

[1]Nanjing University of Aeronautics and Astronautics, Collaborative Innovation Center of Novel Software Technology and Industrialization [2] Dalian Naval Academy

ABSTRACT

Recently, more and more researchers have used deep reinforcement learning (DRL) to solve the UAV local path planning problem. However, existing DRL didn't consider the importance of recent experience for path planning, such as proximal policy optimization (PPO). Moreover, the Actor-Critic framework of PPO suffers from the problem of high variance. To address the above issues, we proposed a Delayed-policy-update PPO with a Prioritized Reply of Recent experience (DPPO-PR2) for local path planning. Firstly, we designed an adaptive parameter to calculate the probability of resampling. By limiting the range of resampling, the possibility of resampling the recent experience is increased. Secondly, a parameter experimentally is picked to make the Actor Network have fewer updates than the Critic Network. Finally, we verified our algorithm has better convergence results and faster execution in six test scenarios.

Index Terms— UAV local path planning, Proximal Policy Optimization, Prioritized Reply, Delayed-policy-update, Recent Experience

1. INTRODUCTION

The UAV local path planning as using available information to adjust attitude and reach a given position in real-time. During navigation, the UAV avoids collisions with obstacles or entering dangerous areas based on the data obtained from the sensors [1]. Traditional path planning algorithms include the dynamic window approach (DWA)[2], the Artificial Potential Field (APF) [3], and the Rapidly-exploring Random Tree (RRT) [1]. However, these algorithms may fall into local optimal points making it difficult to reach the optimal path.

Reinforcement learning (RL) uses the interaction between the agent and the environment to maximize the reward value and improves itself through trial-and-error learning. It is suitable for solving problems with complex models, such as target detection and Continuous Markov Decision Problems [4, 5]. With the development of artificial intelligence techniques, more and more researchers are applying RL techniques to the motion planning of UAV [6, 7, 8]. But for large devices like UAV, each experiment requires a lot of human and material resources. It is difficult to generate large amounts of data for real-time planning, so there is an urgent need to improve the sample efficiency of the DRL. To enable agents to remember and reuse past experiences and improve sampling efficiency, Schaul [9]was the first to propose Preferential Experience Replay (PER), which allows agents to replay experiences based on importance and accelerates algorithm convergence. Various DRL combined with PER are derived: PER-DDPG [10], PER-TD3 [11], PER-PPO [12], etc.

The existing PER measures the importance of all experiences in the current archive and doesn't focus on the importance of new experiences. In addition, DRL based on the Actor-Critic (AC) framework such as PPO have the problem of high variance estimation [13].

To address the above issues, DPPO-PR2 is proposed for local path planning. The main contributions of this paper are as follows: With the characteristics of UAV path planning, we first designed an adaptive parameter to calculate the probability of resampling. By limiting the range of resampling, the possibility of resampling the recent experience is increased, which improves data utilization and sample diversity and faster algorithm convergence. Then, a parameter experimentally is picked to make the Actor Network have fewer updates than the Critic Network. To reduce per-update errors and further improve performance. Finally, Experiments on six test scenarios showed that the performance of the DPPO-PR2 outperformed other DRL on benchmark tasks.

2. PROBLEM FORMULATION

2.1. Deep Reinforcement Learning (DRL)

UAV local path planning problem can be transformed into a policy search problem in the Markov Decision Process (MDP) and is defined by a four-tuple $(S, A, p(s,a), r(s,a))$. where the state transition probability $p(s,a) : S \times A \rightarrow [0, +\infty)$ represents the probability of the agent's next state, $s \in S$, occurring under the current action $a \in A$. The environment emits a reward $r(s,a) : S \times A \rightarrow [r_{min}, r_{max})$ at each transition [11].

This work is supported by the 14th Five-Year Planning Equipment Pre-Research Program under Grant No.: JZX7Y20210401001801.

2.2. Observation Space and Action Space Specification

In DRL, the state space not only represent the current state of the UAV, but also provide information about obstacles and targets in the environment. The state space $S = (P_c, P_g, obs_{pIn})$ consists of the UAV's current position P_c, the target position P_g and partial information about the environment detected by the sensors obs_{pIn}. obs_{pIn} mainly consists of obstacle positions. The environment is continuously updated based on P_c, the radius, speed, and position of each obstacle. And feedback to the nearest obstacle position to the UAV.

The action space $A = (a_x, a_y, a_z)$ consists of the vectors a_x, a_y, and a_z on the x, y, and z-axis as derived by the Actor Network, and the next position is produced by the action vector added to the previous location of the UAV.

2.3. Reward

The reward $R(a|s)$ is significantly correlated with the algorithm's performance and the UAV's behavior. We offer the idea of "non-safety zones" to achieve obstacle avoidance and prevent the scenario when the UAV is too near the surface of an obstacle. The following penalty is applied if the UAV is within a dynamic obstacle at time $t+1$ or the next route point is too near to an obstacle's surface.

$$reward_1 = \begin{cases} \frac{\|P_{t+1}-C_0\|}{R_0} - T_0, if \|P_{t+1}-C_0\| \le R_0 \\ \frac{\|P_{t+1}-C_0\|-(R_0+NS)}{R_0} - T_2, \\ if R_0 \le \|P_{t+1}-C_0\| \le R_0+NS \end{cases} \tag{1}$$

Where P_{t+1} is the position of the UAV at moment $t+1$. C_0 is the center of the obstacle, and R_0 is the radius of the obstacle. T_0 is a positive constant. $NS > 0$ is a non-safety zone and T_2 is a positive constant. If P_{t+1} doesn't lie within a dynamic obstacle, the following reward function is applied, and P_S is the starting position of the UAV.

$$reward_2 = \begin{cases} -\frac{\|P_{t+1}-P_g\|}{\|P_s-P_g\|}, if \|P_{t+1}-P_g\| > \varepsilon \\ -\frac{\|P_{t+1}-P_g\|}{\|P_s-P_g\|} + T_1, if \|P_{t+1}-P_g\| < \varepsilon \end{cases} \tag{2}$$

Following the above, the reward $R_{(a_t|s_t)}$ can be expressed as follows:

$$R_{(a_t|s_t)} = \gamma_1 \cdot reward_1 + \gamma_2 \cdot reward_2 \tag{3}$$

where $\gamma_1, \gamma_2 > 0$ is the weighting factor.

3. DPPO-PR2 FOR MOTION CONTROL

3.1. Proximal Policy Optimization (PPO)

To prevent performance crashes caused by step size selection, Trust Region Policy Optimization (TRPO) [14] uses a KL-divergence constraint to optimize policy. TRPO is theoretically able to ensure performance improvement in the policy. But TRPO is computationally inefficient and difficult to handle high-dimensional problems. The PPO is introduced to simplify the computation [15]. PPO updates the policy by clipping constraints on the ratio $r_{s,a}(\pi)$. the objective function of the PPO in clip truncation mode is given by

$$J^{clip}(\theta) = E_t[min(r_t(\theta)A_t, clip(r_t(\theta), 1-\varepsilon, 1+\varepsilon)A_t)] \tag{4}$$

$$r_t(\theta) = \frac{\pi(a_t|s_t;\theta_{new})}{\pi(a_t|s_t;\theta_{old})} \tag{5}$$

where $r_t(\theta)$ is the ratio of the current strategy $\pi(a_t|s_t;\theta_{new})$ to the previous strategy $\pi(a_t|s_t;\theta_{old})$. A_t is the generalized advantage estimation (GAE) value [16]. The bias is reduced by multi-step experience to obtain an advantage estimate of the current action value. And the following equation determines A_t.

$$A_t = \delta_t + (\gamma\lambda)\delta_{t+1} + \cdots + (\gamma\lambda)^{T-t-1}\delta_{T-1} \tag{6}$$

$$\delta_t = r_t + \lambda V(s_{t+1}) - V(s_t) \tag{7}$$

Where γ and λ are hyperparameters.

3.2. Proximal Policy Optimization with Prioritized Experience Replay

Experience Replay [17] samples from historical experience, solving the problem of agents having to learn online in the sequence of samples. PER is an improvement on Experience Replay that gives preferences to critical samples by increasing the sampling probability of the data. And priorities are measured by TD error. This paper combines the characteristics of the PPO to measure priority through a dominance function with an error calculated as

$$|\delta_i| = |r_i + \gamma V(s_{i+1}) - V(s_i)| \tag{8}$$

For the i-th data point, the prioritized value p_i is defined as $p_i = |\delta_i| + \varepsilon$. the probability of sampling the data point $P(i)$ is calculated as follows.

$$P(i) = \frac{p_i^\alpha}{\sum_j p_j^\alpha} \tag{9}$$

In [9] a fixed value of the index α is used. The bias due to prioritized replay is corrected by using importance-sampling (IS) weights w_i. However, this priority in UAV path planning cannot be permanent. As training moves on, the strategies employed by the UAV are continuously improving. For instance, when training at the beginning, UAV are unaware of their environment, thus executing actions without

much consideration of the obstacles lying ahead. As a result, the percentage of dangerous experiences is higher. With further training, UAV gradually find an excellent policy to avoid obstacles, and the percentage of dangerous experiences decreases. In this case, if experience replay is still used, the neural network will become unstable or even non-convergent [18].

Therefore, We designed an adaptive parameter α to calculate the probability of resampling, which removes the bias by changing the index, and α is calculated as follows.

$$\alpha = \beta^{\frac{episode_{Max} - episode_i}{episode_{Max}}} \tag{10}$$

β is a hyperparameter that controls how much the priority value affects the sampling probability. $episode_{Max}$ is the total episodes and $episode_i$ indicates the number of rounds.

3.3. Prioritized Reply of Recent Experience

In this section, a prioritized replay of recent experience (PR2) is proposed. The core idea is to sample from all data in the replay buffer during the parameter update phase, gradually narrowing down the sampling range from the first to the last batch in order to sample more aggressively from the nearest data points. There are two key points to this scheme: (i) recent data is sampled at a higher frequency; (ii) updates of older data do not override with fresher data. Assume that the Kth mini-batch update is occurring. Let N be the maximum capacity of the replay buffer. For the kth update, $1 \leq k \leq K$, uniformly sample from the nearest s_k data point, where

$$s_k = max\{N \cdot \mu^{k \frac{m}{actualRunNum}}, s_{min}\} \tag{11}$$

Where $\mu \in (0, 1]$ is a hyperparameter that determines the importance given to recent data. $actualRunNum$ is the actual number of runs. The smallest sampling range allowed is s_{min}. To prevent sampling from few near-term data. With the k growing, the older data are excluded. The fresher the data, the more likely to be sampled. This process allows us to make better use of recently visited data and maintain an acceptable range to reuse past data. We take values for μ by simulated annealing methods. At moment t, $\mu_t = \mu_0 + (\mu_T - \mu_0) \cdot \frac{t}{T}$, where $\mu_T = 1$ is annealing to uniform sampling. m is a constant, we have experimented intensively and concluded that the best results are achieved when $m = 10$.

3.4. DPPO-PR2

The PPO is a DRL based on the Actor-Critic (AC) framework, where the value estimate is a function of the future reward and the estimation error, and the estimation variance is related to the future reward and the variance of the estimation error [19]. If the error at each update is not controlled, the variance will grow significantly with each update. High variance estimates provide a noisy gradient for policy updates. This will reduce the learning speed and impair the performance of the algorithm [13].

Algorithm 1: DPPO-PR2

Initialize learning rate γ, parameter β, max episodes M_E, max steps M_{step}, prioritized memory P_{list}, repeat times Rt, Number of delayed-policy-update m
while episode $< M_E$ **do**
  Initialize single obstacle environments, get the state $s_0 = [P_c, P_g, obs_{pIn}]$
  Initialize temp memory $M_{list} = []$
  while s $< M_{step}$ **do**
    Select action $a \sim \pi_\theta(a_t|s_t)$, action probability p_t
    next state s_{t+1}, Observe reward r_t, done d_t
    if d_t *is True* **then**
      Reset the environment
      Learning Rate $\gamma = 0$
    end
    Store transition tuple $(r_t, d_t, s_t, a_t, p_t)$ in buffer M_{list}
    Update learning rate, UAV position, obstacle position
  end
  $K = actualRunNum$
  while k $< K$ **do**
    /* PR2 */
    compute the sampling range using Eq. 11
    sample a mini-batch with probabilities $P(i)$ using Eq. 9
    /* Delayed-policy-update */
    while t $< R_t$ **do**
      Update the critic network
      Update priorities and store them in P_{list}
      if t *mod* m **then**
        Update the actor network
      end
    end
  end
end
Output the best solution found.

This paper makes the variance as small as possible by delaying policy updates. Limiting the possibility of repeated updates to the unchanged critic. Fewer policy updates produce a smaller variance value estimate [19]. We have experimented extensively and found that the best results are produced when the number of Critic-Network updates is three times higher than the number of Actor-Network updates.

Combined with the PR2 described in Section 3.3, The DPPO-PR2 is proposed. The DPPO-PR2 first divides the sampled empirical data into multiple mini-batches and then

performs PR2. Finally, the problem of high variance estimation in the AC framework is solved by deferring policy updates to produce a smaller variance. The DPPO-PR2 is detailed in Algorithm 1.

4. SIMULATION RESULTS

Each scenario in the training phase begins with a random obstacle. After the episode training ends, the test is conducted in a setting with lots of dynamic obstacles. As the random character of RL may produce different results, different random number seeds are set for multi-training during the experiment.

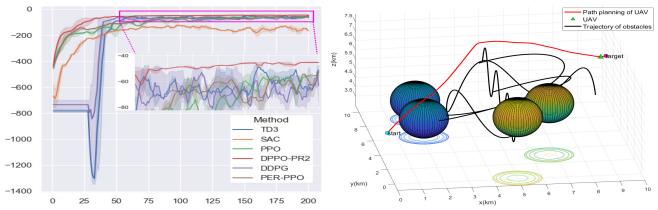
Figure 1(a) makes it clear that DPPO-PR2 has a noticeable advantage in convergence speed in this scenario and has a smaller fluctuation range with the highest reward. Figure 1(b) shows that AUV can safely reach the target area from the starting position with almost the shortest path in this scenario.

To verify the DPPO-PR2's efficacy in solving the path planning problem, six indicators are used to evaluate the planned paths [20]: reward value r, path maximum detour LS, path global detour GS, path length L, obstacle threat index T, and the time needed to track the path ts. For each environment, six indicators are calculated, and the results are recorded in Table 1. The parameters of the other five algorithms are consistent with those set in the corresponding literature (PPO [15], TD3 [11], DDPG [21] SAC[22], and PER-PPO [12]).

Table 1 shows that the DPPO-PR2 significantly outperforms the other algorithms in the three indicators r, T, and ts. The reward is a comprehensive assessment indicator of path performance. Based on the experimental data, DPPO-PR2 outperforms other algorithms in all environments, demonstrating a significant performance increase for DPPO-PR2. Except for DPPO-PR2 and DDPG, which can design pathways devoid of dangers ($T = 0$), all algorithms in Env-1 are unable to create paths that are non-intersecting with obstacles ($T = +\infty$). Using the DPPO-PR2, the UAV can arrive at the target point much more quickly because the paths planned by DPPO-PR2 always take significantly less time to execute than the paths planned by the other algorithm. In most scenarios, The pathways devised by the other algorithms often need twice as long to perform. In terms of LS and GS, as well as path length L, the paths planned using the DPPO-PR2 are also nearly with or optimal.

5. CONCLUSION AND FUTURE WORK

For the UAV local path planning problem, based on the characteristics of UAV path planning, PR2 is firstly proposed, which can significantly improve the learning speed of PPO. Then, to address the problem of high variance estimation in the AC framework of PPO, the variance is made as small as possible by delayed-policy-update. As a result, the PPO performs better and learns more quickly in challenging environments. The trained model is tested in six environments. Results show that the DPPO-PR2 has an excellent online planning capability with faster planning speed. For future work, we also plan to test PR2 on other DRL (e.g., TD3) and other benchmarks (e.g., Atari games) to see whether the method is applicable to other algorithms and environments.



(a) cumulative return curve (b) path planning results

Fig. 1: Cumulative return curves and path planning results under En-5

Scenario	Algorithm	r	Ls	Gs	$L(Km)$	T	$ts(s)$
Env-1	PPO	-55.81	**26.56**	**1.78**	**13.92**	Inf	53.63
	DDPG	-57.17	101.61	14.76	44.47	**0.00**	90.79
	TD3	50.98	104.49	11.5	19.17	inf	46.18
	SAC	-92.53	122.31	37.42	28.57	inf	90.88
	PER-PPO	-56.87	51.44	3.36	15.96	23.86	73.05
	DPPO-PR2	**-45.94**	33.23	3.66	14.93	**0.00**	**24.12**
Env-2	PPO	-49.58	31.35	**1.06**	**13.42**	**0.00**	38.74
	DDPG	-64.31	124.89	11.35	33.39	**0.00**	81.00
	TD3	-47.10	90.05	5.23	18.78	**0.00**	38.85
	SAC	-153.9	137.26	38.45	55.55	inf	239.6
	PER-PPO	-48.93	76.86	2.10	14.50	**0.00**	39.66
	DPPO-PR2	**-45.22**	**17.74**	1.22	13.77	**0.00**	**23.01**
Env-3	PPO	-52.69	76.75	1.98	**13.80**	9.98	58.63
	DDPG	-48.62	47.04	5.52	19.18	8.50	32.66
	TD3	-50.11	85.55	8.66	21.73	**0.00**	53.30
	SAC	-98.07	139.44	40.24	27.01	5.73	72.95
	PER-PPO	-49.7	60.34	**1.79**	14.27	**0.00**	44.36
	DPPO-PR2	**-45.62**	**30.48**	2.22	14.36	**0.00**	**22.57**
Env-4	PPO	-50.38	30.90	**0.79**	**13.25**	20.88	37.76
	DDPG	-50.40	94.09	9.26	21.91	**0.00**	42.45
	TD3	-46.81	78.68	6.75	19.06	**0.00**	45.47
	SAC	-135.08	148.37	37.86	55.79	inf	229.1
	PER-PPO	-50.01	**19.00**	1.00	14.81	**0.00**	46.07
	DPPO-PR2	**-45.63**	30.91	2.40	14.34	**0.00**	**22.45**
Env-5	PPO	-55.96	**15.50**	1.15	**13.75**	inf	55.65
	DDPG	-47.35	86.36	8.47	18.34	**0.00**	32.21
	TD3	-50.18	89.12	7.91	21.76	**0.00**	59.29
	SAC	-113.14	-113.14	44.66	39.96	inf	140.2
	PER-PPO	-51.89	50.33	1.66	14.10	15.43	44.10
	DPPO-PR2	**-45.56**	17.44	**1.14**	14.15	**0.00**	**22.55**
Env-6	PPO	-57.00	81.55	2.49	16.32	**0.00**	121.8
	DDPG	-48.57	76.18	9.88	25.74	**0.00**	41.76
	TD3	-49.51	79.90	9.53	19.16	15.56	40.39
	SAC	-112.50	138.48	36.48	50.03	27.51	192.9
	PER-PPO	-51.66	**10.34**	**1.54**	**14.19**	**0.00**	47.36
	DPPO-PR2	**-47.43**	22.39	3.48	14.36	9.76	**25.08**

Table 1: Comparison of algorithm indicators in six test environments

6. REFERENCES

[1] Yu Wu and Kin Huat Low, "An adaptive path replanning method for coordinated operations of drone in dynamic urban environments," *IEEE Systems Journal*, vol. 15, no. 3, pp. 4600–4611, 2020.

[2] Xunyu Zhong, Jun Tian, Huosheng Hu, and Xiafu Peng, "Hybrid path planning based on safe a* algorithm and adaptive window approach for mobile robot in large-scale dynamic environment," *Journal of Intelligent & Robotic Systems*, vol. 99, no. 1, pp. 65–77, 2020.

[3] Zheng Wang, Guangfu Li, and Jia Ren, "Dynamic path planning for unmanned surface vehicle in complex offshore areas based on hybrid algorithm," *Computer Communications*, vol. 166, pp. 49–56, 2021.

[4] Yihao Luo, Xiang Cao, Juntao Zhang, Leixilan Pan, Tianjiang Wang, and Qi Feng, "Multi-scale reinforcement learning strategy for object detection," pp. 2015–2019, 2022.

[5] Ekaterina Tolstaya, Alec Koppel, Ethan Stump, and Alejandro Ribeiro, "Nonparametric stochastic compositional gradient descent for q-learning in continuous markov decision problems," pp. 6608–6615, 2018.

[6] Daniel Bonilla Licea, Edmond Nurellari, and Mounir Ghogho, "Energy-efficient 3d uav trajectory design for data collection in wireless sensor networks," pp. 8329–8333, 2020.

[7] Muleilan Pei, Hao An, Bo Liu, and Changhong Wang, "An improved dyna-q algorithm for mobile robot path planning in unknown dynamic environment," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2021.

[8] Chao Wang, Jian Wang, Yuan Shen, and Xudong Zhang, "Autonomous navigation of uavs in large-scale complex environments: A deep reinforcement learning approach," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 2124–2136, 2019.

[9] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver, "Prioritized experience replay," *arXiv preprint arXiv:1511.05952*, 2015.

[10] Yuenan Hou, Lifeng Liu, Qing Wei, Xudong Xu, and Chunlin Chen, "A novel ddpg method with prioritized experience replay," pp. 316–321, 2017.

[11] Sitong Zhang, Yibing Li, and Qianhui Dong, "Autonomous navigation of uav in multi-obstacle environments based on a deep reinforcement learning approach," *Applied Soft Computing*, vol. 115, pp. 108194, 2022.

[12] Xingxing Liang, Yang Ma, Yanghe Feng, and Zhong Liu, "Ptr-ppo: Proximal policy optimization with prioritized trajectory replay," *arXiv preprint arXiv:2112.03798*, 2021.

[13] Richard S Sutton, "Learning to predict by the methods of temporal differences," *Machine learning*, vol. 3, no. 1, pp. 9–44, 1988.

[14] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz, "Trust region policy optimization," pp. 1889–1897, 2015.

[15] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov, "Proximal policy optimization algorithms," 2017.

[16] Scott Fujimoto, Herke Hoof, and David Meger, "Addressing function approximation error in actor-critic methods," pp. 1587–1596, 2018.

[17] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel, "High-dimensional continuous control using generalized advantage estimation," *arXiv preprint arXiv:1506.02438*, 2015.

[18] Richard S Sutton and Andrew G Barto, "Reinforcement learning: An introduction," 2018.

[19] HU Zijian, GAO Xiaoguang, WAN Kaifang, ZHAI Yiwei, and WANG Qianglong, "Relevant experience learning: A deep reinforcement learning method for uav autonomous motion planning in complex unknown environments," *Chinese Journal of Aeronautics*, vol. 34, no. 12, pp. 187–204, 2021.

[20] Yunfei Zhang and Honglun Wang, "Adaptive interfered fluid dynamic system algorithm based on deep reinforcement learning framework," pp. 1388–1397, 2021.

[21] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[22] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," pp. 1861–1870, 2018.