Deep Reinforcement Learning for Real-Time Obstacle Avoidance and Local Path Replanning of Robots in Complex Dynamic Scenes

Borui Zhang

{u202410496@hust.edu.cn}

School of Mechanical Sciencce & Engineering, Huazhong University of Science and Technology, 430074, Wuhan, China

Abstract. In recent years, local path planning methods have been widely used in real scenes. With the development of deep reinforcement learning technology, the local path planning method combined with various reinforcement learning algorithms significantly improves the working efficiency of the robot. This paper briefly introduces the modeling method and basic control principle of local path planning problem, reviews the research of local path planning method combined with deep reinforcement learning in recent years, and introduces the local path planning method combined with safety reinforcement learning. Finally, the paper summarizes the research results in recent years and points out the opportunities and challenges in this field, which provides optimization ideas and development guidance for future research.

Keywords: Local Path Planning, Deep Reinforcement Learning, Mobile Robot Navigation, Autonomous Navigation

1 Introduction

1.1 Background

Path planning plays an important role in the field of robots. Its core goal is to enable robots to find one or more feasible and collision free routes on a given map and start and end points. It is widely used in many fields, such as logistics, warehousing, indoor services and so on. [1].

With the increasing complexity of application scenarios, path planning algorithms are also constantly developing. According to the amount of information obtained, the current path planning methods can be divided into global path planning and local path planning. The former refers to planning a complete path on the basis of a known map. The traditional global path planning methods include graph search method, swarm intelligence method and rapid expansion random tree method. This kind of method performs well in static environment, but it can not adapt to the dynamic environment in reality. Local path planning is to generate local feasible trajectory according to the real-time input signal of the sensor. These methods often have

ICIAAI 2026, May 22-24, Singapore, Singapore

DOI 10.4108/eai.22-5-2026.2365334

excellent real-time and dynamic response capabilities, and are highly deployable in real environments. However, due to the limited information available, these methods cannot guarantee the global optimality of the path Therefore, in practical applications, the two often work collaboratively: global planning provides a globally optimal reference path, while local planning is responsible for optimizing and adjusting the reference path provided by the global planning in real time.

1.2 The current development status of local path planning technology

Traditional local path planning algorithms include Dynamic Window Algorithm (DWA) and Artificial Potential Field method (APF). The DWA method was proposed by Fox et al. in 1997 to plan the next motion state of the robot by random sampling in the velocity space, which considers the dynamic limitations of the robot and has high real-time performance, but is seriously limited by manually set parameters, weights, and evaluation functions, and lacks intelligent decision-making ability [2]. The APF method was proposed by Khatib et al. in 1986 to achieve real-time obstacle avoidance by constructing a virtual potential field, but the disadvantage is that it is easy to fall into local minima [3].

Nowadays, the increasing complexity of mobile robot application scenarios has brought many new challenges to traditional local path planning methods: first, the uncertainty caused by the introduction of dynamic scenarios has high requirements for adaptive decision-making. Secondly, complex local path planning tasks have a continuous and high-dimensional motion space, which requires high real-time performance and control accuracy of the method. In addition, as the number of task objectives continues to increase, the algorithm should have the ability to adaptively balance multiple objectives, rather than relying on manually set weights and objective functions. At the same time, the evolution of sensor technology has brought high-dimensional perceptual inputs, requiring methods to adapt to more complex state space representations.

Traditional path planning methods find it difficult to meet these requirements, but by introducing deep reinforcement learning techniques, researchers have successfully broken this impasse. Thanks to deep neural networks and end-to-end models, deep reinforcement learning achieves end-to-end learning from high-level perceptual inputs to decision outputs, and its polynomial complexity also has a natural advantage in solving such sequential decision problems. However, such methods still have deficiencies in learning efficiency, generalization, and safety.

1.3 Research motivation

In recent years, many researchers have achieved gratifying results in this field, but most studies only focus on the performance of specific algorithms or application scenarios. Therefore, this paper categorizes existing research into three types, such as value-based, policy gradient-based, and safe reinforcement learning. They are trained with different strategies and systematically summarizes each type, while also pointing out current challenges and future research directions.

The structure of this paper is organized as follows: Section 2 mainly introduces modeling methods for local path planning problems and the basic framework of deep reinforcement learning in local path planning. Section 3 summarizes and analyzes recent research progress in discrete action spaces, continuous action spaces, and safe reinforcement learning. Section 4

further discusses the challenges currently faced in research and future directions, including issues of sample efficiency, generalization and Sim-to-Real problems, as well as safety concerns. Finally, Section 5 concludes the paper and presents an outlook for future work.

2 Basic principles and framework

2.1 MDP formulation of local path planning

The local path planning problem is essentially a state-driven sequential decision-making problem, with the objective of maximizing long-term accumulated benefits, and is therefore suitable to be modeled as a Markov Decision Process (MDP). Due to the dynamic nature of real-world environments and the limitations of sensor perception, it is theoretically represented as a seven-tuple:

$$\mathcal{M}_{\mathcal{P}} = \langle S, A, O, R, P, Z, \gamma \rangle \tag{1}$$

where S is the state space, A is the action space, and O is the observation space; $R(s, a, s')$ is the reward function; $P(s'|s, a)$is the state transition probability; $Z(o|s', a)$is the observation probability; $\gamma \in (0,1]$is the discount factor. Considering computational complexity, it is often simplified in engineering practice to a five-tuple:

$$\mathcal{M}_{\mathcal{P}} = \langle S, A, \mathrm{R}, \mathrm{P}, \gamma \rangle \tag{2}$$

it is assumed that the observed o corresponds to the full states. Most existing studies implement local path planning based on this simplified MDP framework.

2.2 State–Action–Reward design

In local path planning problems, researchers mainly focus on the design of the state space S, action space A, and reward function R.

The state space S is a collection of all possible states of the surrounding environment of the robot, which generally contains the start and end points, current positions, obstacles and other information in the local path planning scene. The expression of the state space is an important part of the local path planning problem. The expression of the state space in different task scenarios may be very different. The rationality of the expression and its adaptability to the task scenario directly affect the quality and generalization ability of the final strategy.

Action space A represents the set of all possible actions that the robot can perform in the current state, which is usually divided into discrete actions and continuous actions. Discrete action control usually abstracts local actions into a limited set of discrete actions, which is suitable for learning using value-based methods, and is easier to connect with traditional methods, with the disadvantage of low control accuracy. Continuous motion control directly outputs continuous control signals such as angular velocity and linear velocity from the planner, which is more in line with the motion characteristics of the robot, but requires more samples.

The reward function R measures the feedback obtained by the robot after performing a specific action. In the process of learning strategies, the model need to evaluate each step of its actions and determine the learning direction through the reward function. Therefore, the setting of the reward function will greatly affect the convergence direction of the strategy and the behavior

characteristics of the model. According to the task requirements, the total reward function is generally designed as the weighted sum of rewards for multiple goals, which can determine the focus of the model on different indicators through the weights of different rewards.

2.3 Application of reinforcement learning in local path planning

When using reinforcement learning methods to solve path planning problems, the core objective is to maximize the long-term cumulative return of the policy:

$$J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}[\sum_{k=0}^{\infty}\gamma^k r_{t+k}] \tag{3}$$

where τ denotes the trajectory under a given policy π_θ, and $\gamma \in (0,1)$, with values closer to 1 indicating a stronger emphasis on long-term objectives.

The reason Reinforcement Learning (RL) can learn effective policies through sample interaction in unknown environments is that its value function definition satisfies the Bellman recursion structure. For a given policy π, the action-value function is generally defined as:

$$Q^\pi(s,a) = \mathbb{E}_\pi[G_t \mid s_t = s, a_t = a\ ] \tag{4}$$

According to the recursive definition of the return$G_t = r_t + \gamma G_{t+1}$, the policy optimization problem can be expressed as:

$$Q^*(s,a) = \mathbb{E}\left[r(s,a) + \gamma \max_{a'} Q^*(s',a')\ \right] \tag{5}$$

That is, the policy selects the action at each state that maximizes the action-value function to ensure the maximum future cumulative return. This discrete action optimization method is called the value function approach. Its advantage is that it can replace the manually designed evaluation functions in traditional local path planning methods, significantly reducing the dependency on manual tuning and making the method more intelligent.

However, discrete action control sacrifices smoothness and precision, and high-frequency action switching can lead to instability in the agent's movement in real-world scenarios. To achieve continuous action control, the policy gradient theorem is introduced:

$$\nabla_\theta J(\theta) = \mathbb{E}[\nabla_\theta\, log\, \pi_\theta(a_t \mid s_t) A^{\pi_\theta}(s_t,a_t)] \tag{6}$$

where $A^{\pi_\theta}(s_t,a_t) = Q^{\pi_\theta}(s_t,a_t) - V^{\pi_\theta}(s_t)$is the advantage function; the action-value function Q represents the long-term cumulative return of action a_t, and $\nabla_\theta \log \pi_\theta(a_t \mid s_t)$ indicates the direction to increase the probability of sampling the current action in that state. Through the policy gradient theorem, the model can control continuous parameterized actions, improving control precision and smoothness. However, in practical applications, high variance can lead to unstable training. Therefore, combining value function methods with policy gradient methods has been a trend in recent research.

3 Research progress of local path planning based on DRL

3.1 Value function-based methods

In recent years, many researchers have made breakthroughs in reducing Q-value overestimation in Deep Q-Network (DQN) methods and in end-to-end state space representation. Kim et al.

proposed an improved algorithm combined DQN with GRU and action skipping, which enhances the decision-making capability of the algorithm in Partially Observable Markov Decision Process (POMDP) scenarios through gated units, while reducing the update frequency of new actions to strengthen learning of key actions. This approach achieved good performance in automatic parking robot navigation scenarios [4]. Chen et al. proposed an end-to-end navigation method based on local cost maps, and by combining Double Dueling DQN, prioritized experience replay, and curriculum learning, enabled robots to perform obstacle avoidance, navigation, and real-world transfer in complex environments [5]. Lee et al. also proposed an end-to-end DRL navigation framework, which combines object detection and deep reinforcement learning, allowing robots to autonomously identify targets and avoid obstacles in unknow indoor environments [6]. As shown in Fig. 1, Li et al. proposed a method combining the PER-D3QN algorithm with an artificial potential field reward function optimization. By employing a knowledge transfer training strategy, this method improved the navigation efficiency and learning speed of robots in unknown environments [7].

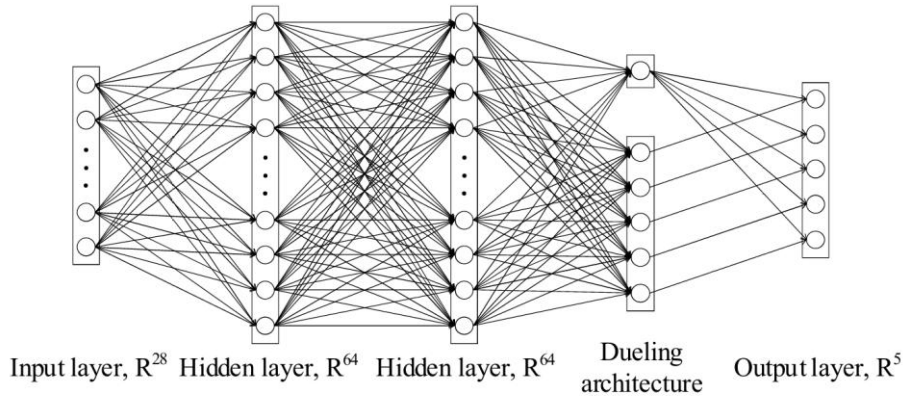

Fig. 1. Network architecture of the PER-D3QN algorithm [7].

Overall, these studies improved the challenges of Q-value overestimation and end-to-end control in existing methods by combining different DQN variants, enhancing state space, and refining reward function representation, providing technological breakthroughs for navigation and obstacle avoidance of mobile robots in complex environments.

3.2 Continuous control methods based on policy gradient

Policy-based reinforcement learning methods can select strategies by calculating the gradient of policy parameters with respect to expected returns and are suitable for continuous action control. On this basis, researchers introduced the Actor-Critic framework, which is divided into two parts. The Actor is responsible for learning and outputting the policy, while the critic evaluates the policy output by the actor and provides update signals to the actor. This structure reduces the estimation variance of traditional policy gradient methods while improving training stability. Many control methods are built upon this framework, such as proximal policy optimization

(PPO), Twin Delayed Deep Deterministic Policy Gradient (TD3), and Soft Actor-Critic (SAC) have emerged.

Improving control methods based on the Actor-Critic framework has been a research focus in recent years. To address the utilization of temporal information, Liu et al. proposed an improved PPO based on Long Short-Term Memory (LSTM) network structure, solving the problem of insufficient memory for temporal information in traditional PPO, thereby enhancing the robot's decision-making ability in the face of complex obstacles; at the same time, by designing virtual target points, the issue of deadlocks during training was reduced [8]. Regarding sample efficiency, Li improved the original TD3 algorithm by introducing TD-error based prioritized experience replay to increase the probability of learning from high-informative experiences, while improving the policy updates through a dynamic delayed update mechanism [9]. As shown in Fig.2, Zhang et al. introduced LSTM networks and prioritized experience replay into the SAC method, and through a burn-in training mechanism, solved the problem of memory degradation caused by hidden state resets during LSTM training [10]. In terms of enhancing algorithm practicality, Zhang designed an end-to-end local path planner based on SAC and adjusted the light detection and ranging (LiDAR) input according to the robot's physical model, giving the sensor information actual physical significance, thereby improving the robot's deployability and obstacle avoidance capability [11]. These studies enhance the decision-making capability of traditional algorithms and improve their applicability in different environments by strengthening historical state information, improving experience sampling mechanisms, modifying network structures, and adjusting training methods.

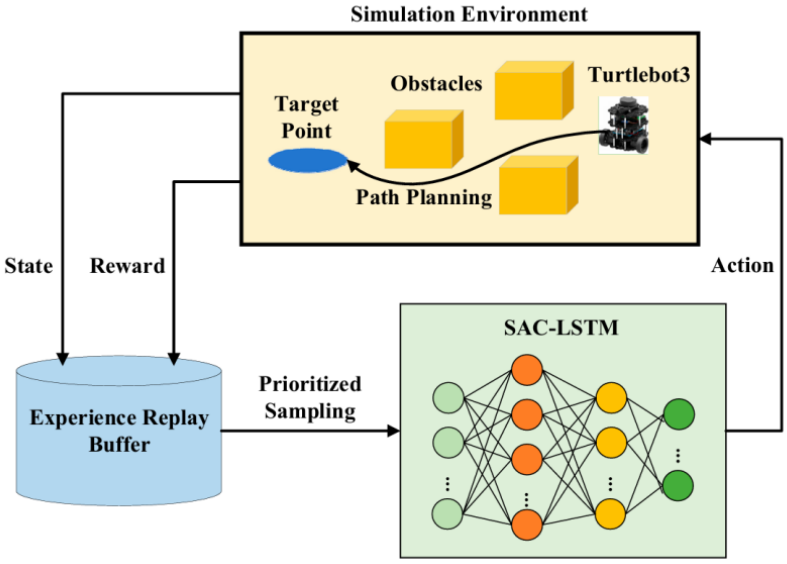

Fig. 2. Path planning framework based on SAC-LSTM algorithm [10].

3.3 Safe reinforcement learning methods

In recent years, with the enhancement of people's safety awareness, safe reinforcement learning methods have gradually been introduced into local path planning tasks. These methods not only aim to maximize cumulative rewards but also require the system's behavior to always satisfy certain safety constraints. Balancing the efficiency and safety of these methods is a key focus of such research.

Classical safe reinforcement learning (RL) methods, such as the Control Barrier Function (CBF) based on RL framework, ensure that robots remain within human-defined safe sets during the learning process by overlaying a CBF controller to correct the output of the RL controller in real time, although it has limitations regarding the affine barrier functions [12]. As shown in Fig.3, Emam et al. improved the original CBF by introducing a Robust Control Barrier Function (RCBF) safety layer, which can encode more complex obstacle shapes and explicitly consider system disturbances; they also proposed a differentiable safety layer, which encodes the output of the safety layer into the RL loss function, significantly improving sample efficiency in car following environments [13]. Recent research has mainly focused on how to balance safety and work efficiency: Dawood and colleagues proposed a dynamic safety barrier method that combines the robustness of an optimized controller with the long-term predictive capability of reinforcement learning. By adaptively adjusting controller parameters in different environments, it successfully improves exploration efficiency based on safety and significantly outperforms baseline methods in multiple metrics [14]; Ugurlu and others defined a novel reward function inspired by Lyapunov stability theory, which penalizes the robot for moving quickly toward obstacles and introduces an early termination mechanism to limit the robot's exploration range. In experiments, this method demonstrated caution when moving near obstacles in a quadrotor drone scenario [15]. Recent research has ensured the safety of system learning and output strategies by introducing explicit safety constraints or constructing safety layers, as well as designing reward functions that consider safety penalties.

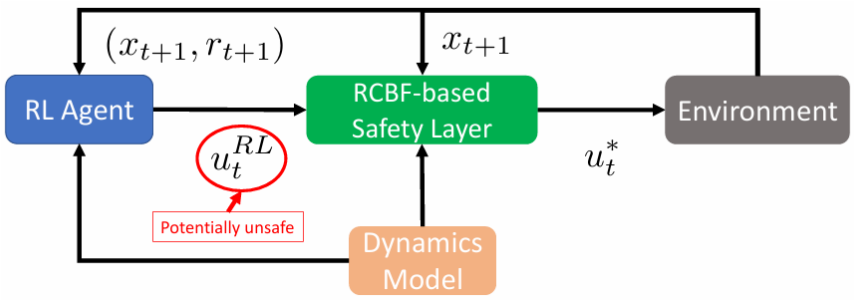

Fig. 3. A diagram of the RCBF framework [13].

4 Current challenges

4.1 Sample efficiency issue

Reinforcement learning learns the optimal strategy through a large number of interactions with the environment, that is, trial and error learning. In the context of local path planning, robots often make random explorations, which means that the algorithm needs many attempts to find high-value actions. And a large number of data will also lead to inefficient sample learning, especially when facing multi-dimensional learning samples in the state space, such as sensor readings, robot position and obstacle information. In order to speed up the efficiency of sample learning, some studies use curriculum learning to gradually increase the complexity of the environment to speed up convergence, or use expert data to strengthen the initial learning to reduce the random exploration of methods. These methods alleviate the severity of the problem to some extent, but there is still a huge room for improvement.

4.2 Generalization ability and sim-to-real issue

Similarly, the over fitting risk of deep reinforcement learning will also lead to the lack of generalization ability of such methods. The expression method of state space and the complexity of the environment will affect the quality of the strategy. Once the complexity of the environment changes, or the researchers' expression design of the state space is not appropriate, the generalization of the model will be reduced. In recent years, researchers have tried to strengthen the generalization ability of the model by randomly simulating the obstacles or environmental parameters during training and increasing the training map samples. Some progress has been made on this issue, but there is still a lot of research space. Other researchers focus on the Sim-to-Real problem during the transformation from simulation training to real deployment. For example, some researches have successfully improved the robustness of the algorithm by randomly setting the sensor noise and environmental friction coefficient in each training through the domain randomization method. However, how to solve the differences of robot dynamics, sensor noise and complex environment between virtual and real environments is still one of the current research focuses.

4.3 Safety and interpretability issues

Although the research in recent years has restricted the method at the security level, it is still a problem that can not be ignored. Neural network is an important part of deep reinforcement learning, but it is difficult to provide strict security by simple mathematical methods. Moreover, improper reward function may lead to reward hacking, which makes the model deviate from the expected safe behavior and make dangerous attempts. These high-risk behaviors are insignificant in simulation training, but they are likely to endanger property and even personal safety when deployed in the real world. Therefore, the balance between the safety and efficiency of the method is still the focus of current researchers.

5 Conclusion

This paper reviews the path planning methods based on deep reinforcement learning. In the theoretical part, this paper introduces the modeling method of local path planning problem including state space, action space and reward function, and then introduces the basic control principle of path planning method combined with deep reinforcement learning from two learning methods based on value and strategy gradient. Subsequently, the paper summarizes recent research. Overall, significant breakthroughs have been made in recent years in discrete and continuous action control as well as safe reinforcement learning, providing new ideas for autonomous navigation of mobile robots; however, issues such as low sample learning efficiency, poor generalization, difficulty in transferring from simulation to reality, and problems with safety and interpretability still exist. Future research should further explore these aspects to promote the application of deep reinforcement learning-based local path planning technology in the field of mobile robotics.

References

[1] Zhu, K: Deep reinforcement learning based mobile robot navigation: A review. Tsinghua Science and Technology, Vol. 26, pp. 674–691 (2021)

[2] Fox, D. et al.: The dynamic window approach to collision avoidance. IEEE Robotics & Automation Magazine, Vol. 4, pp. 23–33 (1997)

[3] Khatib, O.: Real-time obstacle avoidance for manipulators and mobile robots. Autonomous Robot Vehicles, pp. 396–404 (1986)

[4] Kim, I. et al.: Reinforcement learning for navigation of mobile robot with LiDAR. 2021 5th International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 2021, pp. 148–154 (2021)

[5] Chen G. et al.: Robot navigation with map-based deep reinforcement learning. 2020 IEEE International Conference on Networking, Sensing and Control (ICNSC), Nanjing, China, 2020, pp. 1–6 (2020)

[6] Lee, M. et al.: Mobile robot navigation using deep reinforcement learning. Processes, Vol. 10, pp. 2748 (2022)

[7] Li, W. et al.: Navigation of mobile robots based on deep reinforcement learning: Reward function optimization and knowledge transfer. International Journal of Control, Automation and Systems, Vol. 21, pp. 563–574 (2023)

[8] Liu, G. et al.: Local path planning of robot based on improved PPO algorithm. Computer Engineering, Vol. 49, pp. 119–126 (2023)

[9] Li, P. et al.: Path planning of mobile robot based on improved TD3 algorithm in dynamic environment. Heliyon, Vol. 10, pp. 32167 (2024)

[10] Zhang, Y. et al.: Path planning of a mobile robot for a dynamic indoor environment based on an SAC-LSTM algorithm. Sensors, Vol. 23, pp. 9802 (2023)

[11] Wang, Y. et al.: Costmap A* guided reinforcement learning path planning method for complex environments navigation. Journal of Shanghai Jiaotong University, (2024)

[12] Cheng, R. et al.: End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks. Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, pp. 3387–3395 (2019)

[13] Emam, Y. et al.: Safe reinforcement learning using robust control barrier functions. IEEE Robotics and Automation Letters, Vol. 10, pp. 2886–2893 (2025)
[14] Dawood, M. et al.: A dynamic safety shield for safe and efficient reinforcement learning of navigation tasks. arXiv:2412.04153 (2024)
[15] Ugurlu, H. et al.: Lyapunov-inspired deep reinforcement learning for robot navigation in obstacle environments. 2025 IEEE Symposium on Computational Intelligence on Engineering/Cyber Physical Systems (CIES), pp. 1–8 (2025)