Robot Motion Planning in Dynamic Environments: Comparisons and Applications*

1st Xiaoliang Zhang
Deparment of Computer Science
University of York
York, UK
xiaoliang.zhang@york.ac.uk

2nd Pengcheng Liu
Department of Computer Science
University of York
York, UK
pengcheng.liu@york.ac.uk

3rd Alan G. Millard
Department of Computer Science
University of York
York, UK
alan.millard@york.ac.uk

***Abstract*—Motion planning (MP) plays an important role in robotics. In this extended abstract, we provide a short discussion on different categories of MP methods. And there remains a challenge for MP in dynamic environments. We also discuss learning-based MP in dynamic environments.**

***Index Terms*—Motion planning, Dynamic environment, Machine learning**

I. Introduction

Motion planning (MP) is increasingly important nowadays, since it plays a central role in robot motion. A well-performed MP algorithm should better guarantee properties like completeness, optimality, computation efficiency, and insensitivity to environments [1], especially the first two. The most common architecture for a motion planner is a hierarchical design made with a global planner, in charge of generating global waypoints, and a local planner, in charge of generating commands for robots based on the situation of the surroundings.

To achieve such goals, numerous MP methods have been developed through past efforts in academia and industry. Generally, we can classify them into classical approaches and learning-based approaches, based on whether they use machine learning (ML) methods in the MP process or not. For example, classical methods can include Sampling-based MP (SBMP), Optimization-based MP (OBMP), Graph-search MP (GSMP) and Reaction-based MP (RBMP). And typical learning-based methods include Deep Learning (DL), Deep Reinforcement Learning (DRL) and Unsupervised Learning (UL). Each of them has its own pros and cons, which are given in Table II. In recent years, learning-based methods have become increasingly popular.

Although MP has been investigated for a long time, MP in dynamic environments remains a challenge that needs to be solved. Both classical and learning-based methods have limitations in dynamic environment settings, especially if there are some other complicated features. In the next section, a detailed introduction to dynamic environments and the limitations of MP methods in such settings will be given.

TABLE I
Comparison Among Types of Classical MP Models

Methods	Advantages	Disadvantages
GSMP	Could guarantee completeness and optimality	1. Require explicit representations of the environment 2. Suffer from dimension curse when facing high-dimension state space
RBMP	React in real-time with dynamic environments	1. Usually stuck at local optimum 2. Require explicit representations of the environment 3. Perform not well in cluttered environments
SBMP	1. Plan both locally and globally 2. Good at handling complex environments 3. Not require explicit representations of the environment	1. Just guarantee asymptotic optimality and probabilistic completeness 2. High computation cost 3. Trap space problem
OBMP	1. Good at obstacle avoidance and smoothing trajectories 2. Could take some criteria like energy consumption into consideration 3. Not require explicit representations of the environment	1. Tend to stuck at local optimum 2. Require the effort of an expert 3. Require explicitly represent the environment

II. MP in Dynamic Environments

When we mention dynamic environments, we mean that the elements are moving in position with time, and that can include moving obstacles and moving goals. In this extended abstract, we mainly focus on the situation of moving obstacles.

Since the environment is changing with time, planning in real-time is necessary to realize MP in dynamic environments. However, it is very challenging to reach this goal. Generally, MP in a dynamic environment is NP-hard, indicating

TABLE II
COMPARISON AMONG TYPES OF MACHINE LEARNING MODELS FOR MP

Methods	Advantages	Disadvantages
DL	1. Strong adaptability 2. End-to-end usage space 3. Not require explicit representations of the environment 4. Real-time planning	Just one-step prediction
DRL	1. Making sequential decisions 2. Other similar benefits in DL	1. Sparse reward problem 2. Hard to converge 3. Reward shaping is challenging
UL	1. Take uncertainty and noise into consideration 2. Not require explicit representations of the environment 3. Real-time planning	Have difficulty in avoiding obstacles

it is computationally expensive [2]. Moreover, for many MP models, especially the classical ones, rapid re-planning is indispensable due to the changing surroundings, but it will consume a lot of computational resources to do this [3]. Among the classical MP methods, SBMP is more powerful since its high efficiency compared with GSMP and does not require an explicit representation of the environment, and it can guarantee a certain degree of optimum, compared with local planning approaches like RBMP. So it is commonly used in manipulation tasks in dynamic environments, but it still struggles in such situations.

Learning-based MP methods are more applicable in dynamic environments, they are trained offline thus their computation consumption is offloaded when being set to online usage [4]. They also have strong generality in different environments. Moreover, they still work in unknown or partially known environments. The usage of learning-based MP can be roughly classified into two categories, *end-to-end usage*, referring to using only ML models to do MP, and *hybrid usage*, referring to combining ML models and classical planners.

A. End-to-end Learning-based MP

The most attractive motivation for end-to-end ML-based MP models is to save engineering efforts. Also, the powerful data processing capability makes such models could directly process high-dimensional inputs.

For example, Wang et al. [5] applied a two-stream DQN to separately deal with navigation and obstacle avoidance for a mobile robot in a dynamic environment. Johnson et al. [6] also applied a hierarchical structure, with neural networks processing the input costmap and mapping it into the latent space, then the planning network generates the next state.

B. Hybrid Learning-based MP

End-to-end models have limitations. They are limited in long-horizon tasks: DL does not incorporate temporal information and thus can only generate one-step prediction; DRL suffers from high cost when search space is large [7] [8]. Moreover, specifying goals in end-to-end settings is proved to be challenging [9]. Thus, hybrid approaches might be better.

Patel et al. [10] used DWA to make the trajectory generated by DRL kinodynamic feasible. Want et al. [**?**] combined RRT^* with GMR to increase the sampling efficiency. So we can see, in this model, each of chosen models can make up for another one's disadvantage.

III. CONCLUSIONS

In this extended abstract, we introduced the comparisons between commonly used MP methods, and provided an overview of the application of learning-based MP in dynamic environments, which remains a challenge in the research of MP. The state-of-the-art approaches are commonly applying hybrid method combining DRL and classical methods, to make use of both of their benefits. And in this way, we believe hybrid models with more stability, and timely responding capability is still needed.

REFERENCES

[1] A. H. Qureshi, Y. Miao, A. Simeonov, and M. C. Yip, "Motion planning networks: Bridging the gap between learning-based and classical motion planners," *IEEE Transactions on Robotics*, vol. 37, no. 1, pp. 48–66, 2020.

[2] M. Mohanan and A. Salgoankar, "A survey of robotic motion planning in dynamic environments," *Robotics and Autonomous Systems*, vol. 100, pp. 171–185, 2018.

[3] L. Petrović, I. Marković, and I. Petrović, "Mixtures of gaussian processes for robot motion planning using stochastic trajectory optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 12, pp. 7378–7390, 2022.

[4] S. H. Semnani, H. Liu, M. Everett, A. De Ruiter, and J. P. How, "Multi-agent motion planning for dense and dynamic environments via deep reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3221–3226, 2020.

[5] Y. Wang, H. He, and C. Sun, "Learning to navigate through complex dynamic environment with modular deep reinforcement learning," *IEEE Transactions on Games*, vol. 10, no. 4, pp. 400–412, 2018.

[6] J. J. Johnson, L. Li, F. Liu, A. H. Qureshi, and M. C. Yip, "Dynamically constrained motion planning networks for non-holonomic robots," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 6937–6943.

[7] C. Zhou, B. Huang, and P. Fränti, "A review of motion planning algorithms for intelligent robots," *Journal of Intelligent Manufacturing*, vol. 33, no. 2, pp. 387–424, 2022.

[8] S. Nasiriany, H. Liu, and Y. Zhu, "Augmenting reinforcement learning with behavior primitives for diverse manipulation tasks," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 7477–7484.

[9] X. Xiao, B. Liu, G. Warnell, and P. Stone, "Motion planning and control for mobile robot navigation using machine learning: a survey," *Autonomous Robots*, vol. 46, no. 5, pp. 569–597, 2022.

[10] U. Patel, N. K. S. Kumar, A. J. Sathyamoorthy, and D. Manocha, "Dwa-rl: Dynamically feasible deep reinforcement learning policy for robot navigation among mobile obstacles," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 6057–6063.