Trajectory Planning with Moving Obstacles

Mo Chen
Informatics 6 - Robotics, AI , Real-time Systems
Advanced Deep Learning for Robotics - Team 9
Munich, Germany
ge64mah@tum.de

Ragip Volkan Tatlikazan
Informatics 6 - Robotics, AI , Real-time Systems
Advanced Deep Learning for Robotics - Team 9
Munich, Germany
volkan.tatlikazan@tum.de

***Abstract*—This is a proposal for project "Trajectory Planning with Moving Obstacles" of team 9 in course "Advanced Deep Learning for Robotics". Our project focuses on motion planning in dynamic environments. It is explained more in detail in Objective(1), Related Work(2) and Technical Outline(3) sections. Section 1 describes our problem at hand and shortcomings of standard methods to solve it. Section 2 summarizes some related research papers and analyze their importance for our project. Section 3 lists all significant technical details as bullet points which we plan to do in the future.**

***Index Terms*—motion planning, moving obstacles, reinforcement learning**

I. Objective

Motion planning plays an important role in autonomous systems. For instance, a car in autonomous driving mode requires trajectory planning in free space and drones need to avoid collision when flying. With the increasing capability in autonomous systems, motion planning in dynamic environments attracts more attention, especially for autonomous vehicles for avoiding collision with moving obstacles.

Now, there are lots of algorithms for static environments. For example, the Dijkstra algorithm deals well with grid based maps. However, it still lacks deep research in motion planning in dynamic environments which require changing optimal trajectories to keep up with the changes in the environment.

Reinforcement learning (RL) is vital and suitable for dealing with dynamic environments. An agent based on RL can take a different required action after obtaining environmental information in each step. Thus, this project uses Deep-RL methods to solve motion planning problems with moving obstacles.

II. Related work

There has been some research for application of deep learning on motion planning problems. MPNet [1] provides a general neural network structure for motion planning. However, it can only deal with static environments. In addition, MP Net is trained in a supervised learning setting and target data is provided by classical motion planning algorithms. Thus, it has little difference with classical planner, but it costs less computation resources than classical planner.

In a normal RL algorithm, it is hard to design a reward function, which should guide agents to learn from the trials and errors and balance reward and punishment. Scheduled Auxiliary Control (SAC-X) [2] focuses on problems with sparse reward signals and using the key idea of "active (learned) scheduling and execution of auxiliary policies" for new complex tasks. This research can help us to deal with difficulty in reward function in a complex dynamic environment.

Meta-Reinforcement Learning (Meta-RL) is a method which can transfer experience from old tasks to new tasks. For dynamic environments, an agent cannot learn all situations. Thus, it is important for an agent to generalize its knowledge. Meta-RL can use training data efficiently and spend less time on training on new tasks than other RL methods to get a more intelligent agent. Kate et al. [3] combines meta-RL with probabilistic distribution and apply it successfully for robot grasp problems.

III. Technical Outline

We will start by assuming that the agent has full knowledge of the environment, including position and velocity of the obstacles. But we will not consider dynamic constraints on agents as we will focus more on the planning instead of the representation.

First, we establish a simple 2D motion planning problem in a dynamic environment with moving obstacles.

Second, we will apply the RL algorithms from the related work in PyTorch, with a continuous action space on this problem and compare these algorithms performances in our dynamic environment. We will start with the SAC (Soft Actor Critic) algorithm as it is a standard baseline for robots with continuous action spaces and it is sample efficient. Afterwards we will improve on the performance with Meta-RL and/or the SAC-X variant and test their results with the SAC baseline.

Lastly, we will improve on the best algorithm according to our results by considering some uncertainty of the moving obstacles, where we will add some noise on position and velocity of obstacles and optionally planning for a partially observed dynamic environment.

References

[1] Qureshi AH, Miao Y, Simeonov A, Yip MC. Motion planning networks: Bridging the gap between learning-based and classical motion planners. IEEE Transactions on Robotics. 2020 Aug 3;37(1):48-66.

[2] Riedmiller M, Hafner R, Lampe T, Neunert M, Degrave J, Wiele T, Mnih V, Heess N, Springenberg JT. Learning by playing solving sparse reward tasks from scratch. International conference on machine learning 2018 Jul 3 (pp. 4344-4353). PMLR.

[3] Rakelly K, Zhou A, Finn C, Levine S, Quillen D. Efficient off-policy meta-reinforcement learning via probabilistic context variables. International conference on machine learning 2019 May 24 (pp. 5331-5340). PMLR.