Local Path Planning

CSCE Summer Intern Project for Undergraduate Student

Path Planners

Syed Mustafa

Department of Computer Science
Texas A&M University

June 12, 2023

Contents

1 Executive Summary

This document presents a project proposal that focuses on autonomous vehicle (AV) path planning in dynamic environments. The main objective of the project is to investigate and compare different path-planning algorithms to identify the most suitable lane-changing algorithm for an autonomous non-holonomic vehicle in a dynamic environment, where the environment is only partially known and contains moving obstacles.

The identified path-planning algorithms for evaluation are: Globally Guided Reinforcement Learning, Dynamic-Window Approach, and All-in-one Switch Controller. The proposed design involves providing or calculating a world model, a kinematic model of the AV, and a global path at the beginning. Subsequently, a local planner will be implemented in a simulation environment using this information.

To analyze the selected algorithm, the project will utilize a simulation platform, specifically either OpenAI Gym with Pygame or the CARLA simulator. Both platforms offer suitable environments for testing and evaluating the algorithm's performance.

While planning algorithms have various applications, such as gaming, self-driving cars, robot manipulators, and general planning, this project specifically focuses on contextualized planning for autonomous vehicles. By successfully completing this project, it will contribute to the existing literature on path planning in dynamic environments for self-driving cars.

2 Introduction

Autonomous vehicles, or self-driving cars, operate without human intervention, relying on advanced sensors, AI, and connectivity. Recent advancements in technology have brought us closer to realizing autonomous vehicles, with companies and researchers actively working to improve road safety and transform transportation.

However, there are still significant challenges to overcome before autonomous vehicles become a common mode of transportation. Reliability and safety concerns remain, necessitating the development of a more robust and dependable path-planning module within the autonomous vehicle system.

The planning problem can be described using the moving piano problem.

Algorithm 1 Moving Piano Problem(taken from [1])

A world W in which either $W = \mathbb{R}^2 or \mathbb{R}^3$

A semi-algebraic obstacle region $\mathbb{O} \subset W$ in the world.

A semi-algebraic robot is feined in W. It may be a rigid robot A or a collection of m links, $A_1, A_2, ..., A_m$

The configuration space C is determined by specifying the set of all possible transformations that may be applied to the robot. From this, $C_{obs}\&C_{free}$ are derived.

A configuration, $q_i \in C_{free}$ designated as the *initial configuration.*

A configuration, $q_G \in C_{free}$ designated as the *Goal configuration.* Together these two are called a *query pair* and designated as (q_i, q_G)

A complete algorithm must compute a (continuous)path,$\tau : [0,1] \rightarrow C_{free}, \ni \tau(0) = q_I \; and \; \tau(1) = q_G$, or correctly report that a path does not exist.

An open research topic in path planning is path planning in a dynamic environment. The environment in such a case will be partially known, i.e. the current location of dynamic obstacles wouldn't be given.

2.1 Need Statement

A way to implement a lane-changing algorithm so that the autonomous vehicle can avoid dynamic obstacles on its path, i.e. the road.

2.2 Goals and objective

2.2.1 Goals

The literature presented in section 3 would suggest that dynamic path planning is a complex and ongoing research problem. Thus, the goal of this project is to contribute to the existing literature by comparing existing methods of dynamic path planning to:

1. Conclude what methods perform the best.
2. Analyze room for improvement in the current algorithm, if any.

2.2.2 Objectives

Before stating the objectives of the project, let us go over some assumptions of the project. The following assumptions will be followed for this project [1]:

1. The scenario takes place in a 2D environment.
2. The planned global path is fixed and given initially.
3. Static obstacles are given initially.
4. The autonomous vehicle is a 4-wheel vehicle, which can be regarded as a 4.5 x 1.8m rectangular.

The **objectives** of the project are listed below:

1. Use a reliable path planning algorithm, i.e. if a solution$s$ exists then the planner outputs at least one feasible solution.
2. Do a comparative analysis on the various planning methods listed in section 3
3. Simulate the algorithms on various simulating platforms and libraries including but not limited to:
   (a) OpenAI Gym
   (b) CARLA [7]
   (c) Pygames
   (d) Mathworks: Navigation Toolbox^{TM}
4. To get a good understanding of the difficulties involved in the planning problem.

[1] These assumptions might change based on feedback from Dr. Song and Sir Guo.

2.3 Design and Feasibility

The constraints are listed below:

1. **Time**: Given that this project is intended for a 10-week summer research program, the biggest constraint is with time. However, with enough dedication and hard work, I believe that the objectives of the project can be met.

2. **Processing power**: The type of algorithms listed in section 4.1 considerable processing power. Furthermore, processing power would be required to work on the CARLA simulator. As stated in its documentation the simulator requires an Adequate GPU. A GPU with 6 GB is the minimum requirement and 8 GB is recommended.

3 Literature and Technical Survey

This section presents the literature and technical Survey done for this proposal. Note that this section does not encapsulate all the literature that was reviewed. Rather, the papers that are relevant to path planning in a dynamic environment are presented.
The authors of [**?**] presented a novel approach to the problem of path panning in a dynamic environment. implemented a hierarchical path-planning algorithm using the Deep Reinforcement Learning paradigm known as Double Deep Q-Network (DDQN). The presented algorithm combines *global guidance* and *local RL-based planner*. The experiment results showed that the presented planner outperformed A* based global and local re-planning algorithms in terms of moving cost and detour percentage. However, in terms of computing time, the algorithm did not perform well.
The approach that [4] took was that it first identified the strength and weaknesses of the Deep Reinforcement learning approach. The authors realized that the DRL approach works substantially better than traditional methods when it comes to dynamic obstacle avoidance. The authors noted that the tendency of DRL algorithms to become trapped in local minima can result in suboptimal performance in certain scenarios, such as long corridors, corners, and dead ends. To address this limitation, the authors proposed an innovative system that leverages a DRL actor-critic approach to switch between local planners, a strategy aptly named the All-in-One approach. Specifically, the proposed system integrates the model-based Time Elastic Bond (TEB) planner with a DRL-based approach, and the switching mechanism is trained using Lidar data. The underlying logic behind this approach is that the DRL-based planner is activated when the mobile robot operates in environments with dynamic objects, while the TEB planner takes the lead in other situations.
The authors of the [5] introduced a new approach called Real-Time RRT (RT-RRT) that builds upon the Randomly Exploring Random Tree (RRT) algorithm, incorporating elements from RRT* and Informed RRT*. RRT, initially proposed by LaValle, is a popular sampling-based planning algorithm in robotics known for its fast computation. In their work, the authors presented a modified version of RRT that preserves the entire tree structure and performs rewiring of tree nodes based on the tree root's location and changes in dynamic obstacles. This modification allows for more efficient path planning. The paper highlighted that real-time capability was achieved by interleaving the process of path planning with tree expansion and rewiring. A notable feature of the RT-RRT algorithm is that the tree root is moved along with the agent, ensuring that the tree structure is retained instead of constructing it from scratch in each iteration. This approach reduces computation time and allows for better utilization of previously explored regions. By incorporating the principles of RRT* and Informed RRT*, the authors aimed to enhance the exploration and optimization aspects of RRT, ultimately improving the efficiency and effectiveness of the algorithm in dynamic environments. The reported results indicated successful real-time planning by integrating path planning, tree expansion, and rewiring

processes, leading to faster and more adaptive motion planning for robotic systems.
The paper [6] makes the following key contributions to the literature: Global Dynamic Window Approach: The authors propose the GDWA as a planning and control strategy for mobile robots to navigate at high speeds in dynamic environments. The GDWA combines a dynamic window algorithm with global path planning to ensure real-time collision avoidance while maximizing robot velocity.
Dynamic-Window Algorithm: The paper enhances the traditional dynamic window algorithm by incorporating a lookahead mechanism that allows the robot to anticipate potential collisions and select optimal velocities accordingly. By considering both current and predicted future states, the algorithm dynamically adjusts the feasible velocities available to the robot.
Real-time Collision Avoidance: The GDWA ensures real-time collision avoidance by continuously updating the robot's velocity commands based on the dynamic window and the global path plan. The algorithm adapts to changes in the environment and allows the robot to respond swiftly to avoid obstacles while maintaining high velocities.
Thus the identified approaches used in the literature are:

1. Globally Guided reinforcement learning from [3].
2. All for one controller from [4].
3. Real-time RRT algorithm from [5].
4. Dynamic-Window approach from [6]

4 Proposed Work

4.1 Evaluation of alternative solutions

Section 3 identified four solutions to the need statement. One of the objectives of this project is to perform a comparative analysis of the planning methods identified above. To do that all the algorithms must be run on the same environment. Thus, the algorithms will be tested on either OpenAI Gym using PYGAMES or on the CARLA simulator. Of course, the CARLA simulator would be preferred since it has more options to model the environment more accurately.
Moreover, an interesting takeaway from the papers/methods reviewed was that they presented a purely geometric approach to path planning, i.e. they weren't concerned with the configuration of the vehicle. Thus modeling the autonomous vehicle as a non-holonomic differential drive vehicle would add-on to the literature.
Lastly, the planners would be implemented as a lane-changing algorithm, as opposed to generalized obstacle avoidance algorithms (as was the case in the papers mentioned in section 3).

4.2 Design specification

The system design is given in figure 1. The figure shows the high-level modules and how they would interact with one another The high-level modules are:

- Global planner
- Local planner
- Kinematic Model
- Environment
- Simulator

- Evaluation

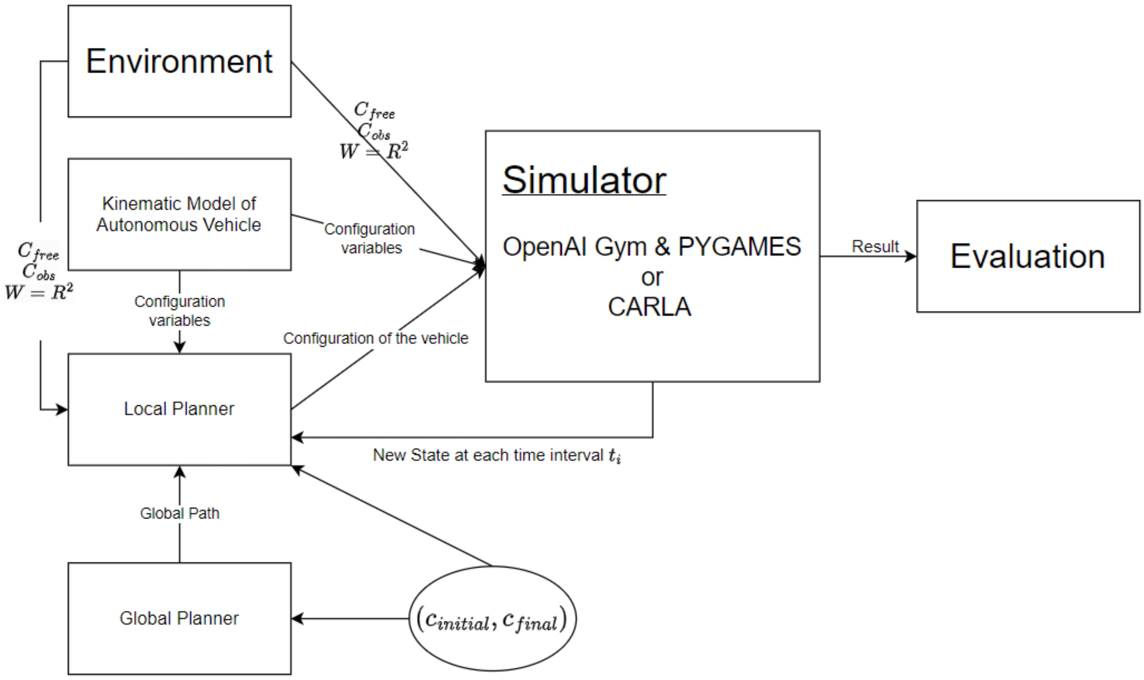

Figure 1: Proposed System Diagram

4.2.1 Environment

The environment would be partially known. The map, the location of static obstacles, and the robot location would be known. The location of a dynamic obstacle would be unknown unless it is detected. Thus using the notation given in 1. The world $W = \mathbb{R}^2$, and C_{free} would contain the autonomous vehicle and dynamic object. And, $A(q_t) \cap d_i(t) = \ \ni A(q_t)$ is the position of the robot at time t, and $d_i(t)$ is the position of the dynamic obstacle. The environment module would be responsible to model the world, i.e. the workspace (W) and configuration space (C).

4.2.2 Simulator

This would be either OpenAI Gym or CARLA. The simulator module would get its inputs from the environment, a kinematic model of the AV, and the local planner. That will input $C_{free}, C_{obs} \& W$, the kinematic model of the vehicle, and the configurations of the vehicle (done by the local planner at each time step t_i). The simulator module would output a new state at each time interval.

4.2.3 Kinematic Model

The vehicle will be modeled as a non-holonomic differential drive vehicle. This module will output the model parameters and input them into the local planner and simulator.

4.2.4 Evaluation

The results will be evaluated on several metrics. Computation time is one example of such a metric. For other metrics, a further literature review needs to be done. The metrics must be such that they can gauge the utility of each algorithm. One possible metric could be the smoothness of the path, which could be measured through some heuristics.

4.3 Approach for design validation

Please refer to section 4.2.4.

5 References

References

[1] LaValle, S. M. (2006). Planning algorithms. Cambridge university press.

[2] LaValle, S. M. (1998). Rapidly-exploring random trees: A new tool for path planning.

[3] Wang, B., Liu, Z., Li, Q., & Prorok, A. (2020). Mobile robot path planning in dynamic environments through globally guided reinforcement learning. IEEE Robotics and Automation Letters, 5(4), 6932-6939.

[4] Kastner, L., Cox, J., Buiyan, T., & Lambrecht, J. (2022, May). All-in-one: A DRL-based control switch combining state-of-the-art navigation planners. In 2022 International Conference on Robotics and Automation (ICRA) (pp. 2861-2867). IEEE.

[5] Naderi, K., Rajamäki, J., & Hämäläinen, P. (2015, November). RT-RRT*: a real-time path planning algorithm based on RRT. In Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games (pp. 113-118).

[6] Brock, O., & Khatib, O. (1999, May). High-speed navigation using the global dynamic window approach. In Proceedings 1999 IEEE international conference on robotics and automation (Cat. No. 99CH36288C) (Vol. 1, pp. 341-346). IEEE.

[7] Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., & Koltun, V. (2017, October). CARLA: An open urban driving simulator. In Conference on robot learning (pp. 1-16). PMLR.