RBE550 - Motion Planning

Real-time Motion Planning With Obstacle Avoidance For Autonomous Vehicles

Pinak Jani and Krutarth Trivedi

MS Robotics 2023

WPI

The proposed project is an intrinsic component of the work - "Real-time Motion Planning for Autonomous Vehicles" and some parts are shared with the co-authors, (Wael Mohammed , and Puru Upadhyay), proposing a project - "Real-Time Motion Planning With Lane Maneuvering for Autonomous Vehicles"

Introduction | Autonomous Vehicle

- Since the last decade, autonomous vehicles are attracting considerable attention from academia, industry, and governments all over the world.

Figure 1: Google Waymo

Figure 2: Ford

- Generally, the autonomous vehicle is structured in the following main layers:
  - Perception, Sensor Fusion, Decision, and Control

- When autonomous vehicles advance toward the realistic road traffic, they are required to be capable of handling various complex maneuvers, such as lane keeping, vehicle following, lane changing, merging, avoiding both static and dynamic objects, and interacting with other traffic participants while complying with the traffic rules.

Project Overview

- The focus of this project would be to develop a motion planner for static and dynamic obstacle avoidance to navigate an ego vehicle in both, structured and unstructured environments.

- We propose to utilize the Sampling based framework for generating a probability road map.

- The global route planning level generates a global route that creates the collision-free way-points to reach the destination point.

- This technique typically lacks the dynamic path planning capabilities and hence motion planning needs to be done at multiple levels.

- Once this path is planned, a local planner implements an algorithm which dynamically modifies the path planned by the the global route level planner if either any static and/or dynamic obstacle obstruct the path or lane-maneuvering has to be performed.

Project Overview

- Structured Environment
  - Since for a structured environment, self-driving vehicles are being made to follow lane-divided roads featuring unidirectional flows.
  - The tasks that to be done by an autonomous vehicle for on-road scenarios:
    - Car Following
    - Lane Keeping
    - Lane Changing
    - Passing
    - Overtaking
    - Obstacle avoidance

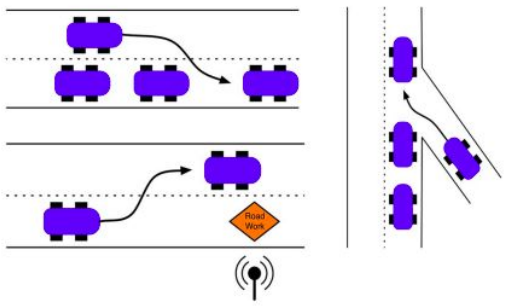


Figure 5: Lane Maneuvering

Project Overview

- Unstructured Environment
  - In absence of a lane structure, an ego vehicle needs to be navigated through static and dynamic obstacle avoidance.

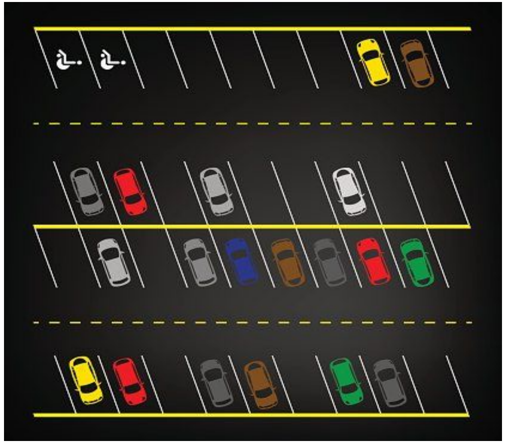


Figure 3: Autonomous Parking Scenario

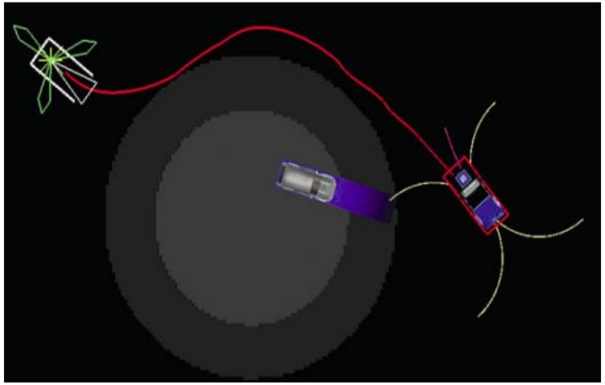
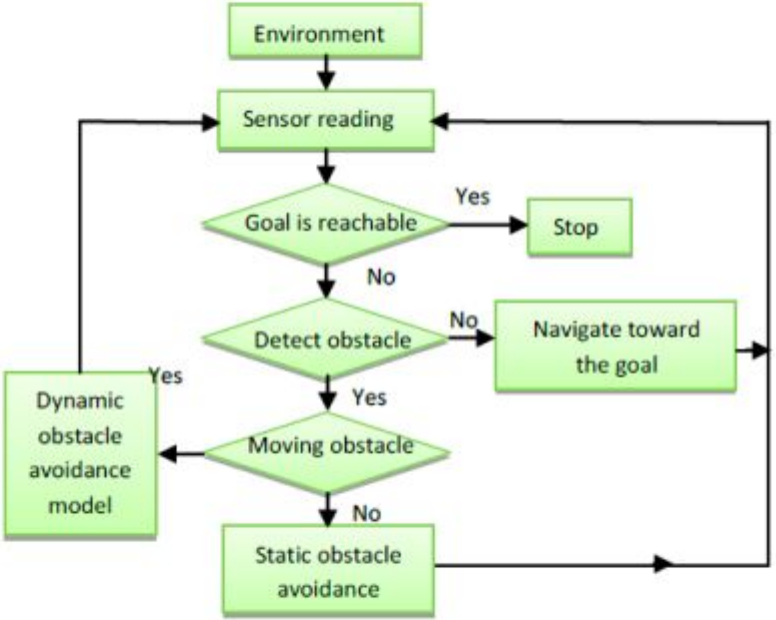


Figure 4:Dynamic Obstacle Avoidance

Methodology | Overview



Methodology | Motion Planner

- The motion planner provides a plan to achieve a particular desired goal considering the current and predicted vehicles states.

- The desired goal could be anything like traversing on a particular lane, avoiding a particular obstacle or following another vehicle.

- The motion planner can be further divided into two different layers:
    a. Global Planner
    b. Local Planner

Methodology | Global Route Planning Level

- The global planner calculates the shortest route from initial position to goal position using the given or generated map of the surrounding environment.
- In order to do this, we sample the entire environment with random points or (nodes of graph) and then we draw edges between these nodes.
- Selected the edges which are collision free, lastly using the Dijkstra discrete Search algorithm, we find the shortest path using these sample nodes while considering the collision checker algorithm on each step.

Methodology | Local Route Planning Level

- The local planner predicts the future states based upon the control inputs to an ego vehicle to perform out the global map, and simultaneously takes care of close-in obstacle avoidance, even if it was not provided by the global plan.

- We develop a 6D state lattice planner and then check for the predicted future states that are in collision or close enough to collision to avoid them. This planner functions in accordance with an obstacle map of all the surrounding obstacles created by collision checker algorithm.

- If the obstacle is close to the proximity of an ego vehicle or if the predicted future trajectory of an ego vehicle lies in the close proximity of the obstacle then that trajectory is encoded as a hard constraint and fed to the local planner to strictly avoid.

Methodology | Collision Checker

- In order to avoid static and/or dynamic obstacles during both, global and local trajectories exploration, we develop a collision checker algorithm based on the convex hull.

- To make the collision-free way-points search more efficient in terms of obstacle avoidance, we pad all the obstacles with our ego vehicle’s dimensions using Minkowski Sum method.

Implementation | Structured Planning

- The structured planning algorithm is responsible for navigating an ego vehicle in the structured environment such as on-road driving scenarios.

- The main objective of this algorithm is to drive an ego vehicle in the desired lane avoiding obstacles, keeping a keep-safe distance from the surrounding cars, and if need arises, plan a safe and collision-free trajectory to overtake the front vehicle.

- This involves use of a multilayer motion planner having the vehicle dynamics, Ackerman steering and dynamic feasibility curve.

Implementation | Unstructured Planning

- The need of an unstructured planning motion goal arises when the vehicle encounters anomalies in the environment which may be a collision with obstacles or a complex maneuver during off-road driving scenario.

- The main aim of this algorithm is to navigate an ego vehicle in given environment while considering static and dynamic obstacle avoidance.

- This planning also prevents the vehicle from reaching undesirable pose or areas considering the vehicle dynamics and thereby allowing local maneuvering and dynamic obstacle avoidance.

Results | Global Route Planner | CARLA

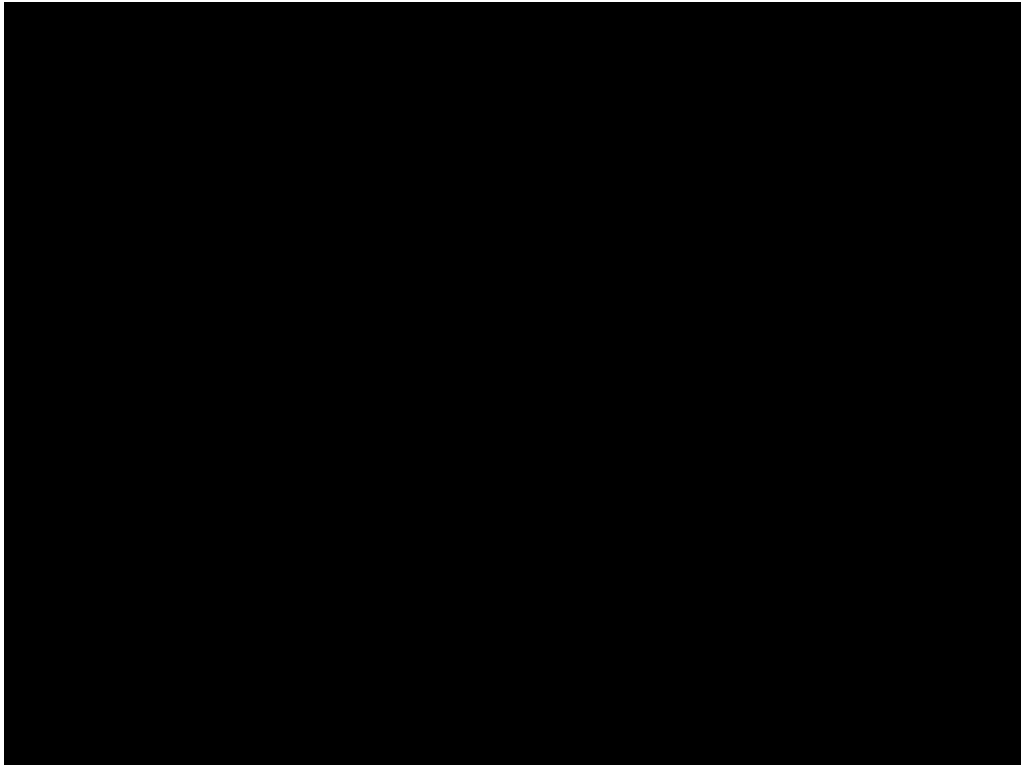
- As we can see in the figures, the global route planner is using Dijkstra algorithm to generate an optimal path from the initial position to the final position.

- Given:
  - A default world map in CARLA.
  - A Non-holonomic ego vehicle with preset vehicle dynamics.

Results| Global Route Planner | CARLA

- As we can see here, for the given start and goal positions, an ego vehicle is following the obstacle-free path generated by the global route planner.

- As we can see, in absence of the behavioral planner, an ego vehicle is not able to follow the traffic signals.



Results | Static and Dynamic Obstacle Avoidance

- In order to validate the robustness of the developed algorithms, we performed exhaustive experiments for the following test cases in pygame:

  1) Navigation in an unstructured environment with randomly placed static and dynamic obstacles

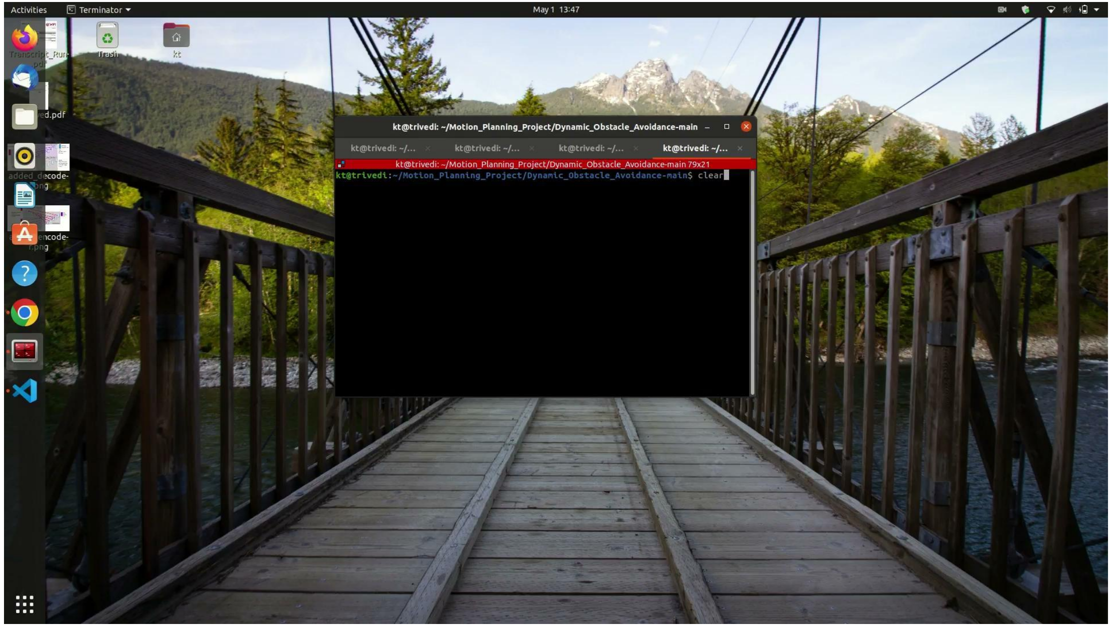
  2) Lane maneuvering with static obstacle

  3) Lane maneuvering with both static and dynamic obstacle

  4) Navigation in an unstructured environment replicating the parking space.

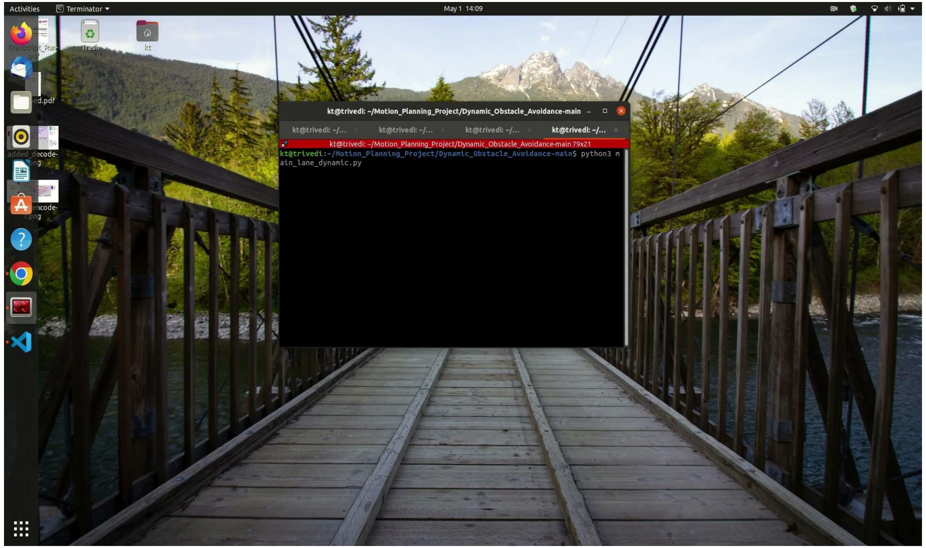
Results | Static and Dynamic Obstacle Avoidance

- Test-case: A structured environment where an ego vehicle plans an optimal global path to reach the goal state avoiding static obstacles.



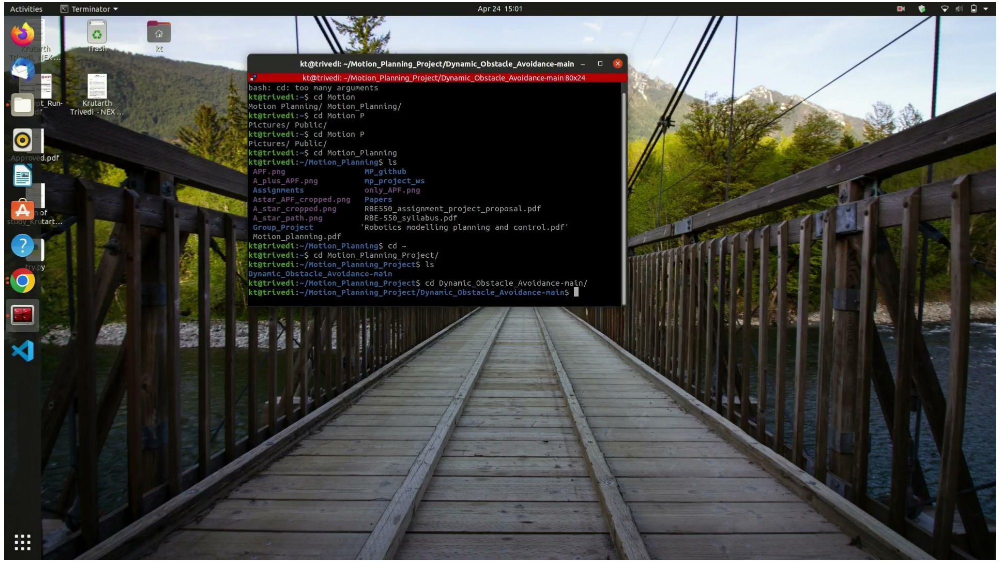
Results | Static and Dynamic Obstacle Avoidance

- Test-case: A structured environment where an ego vehicle plans an optimal global path to reach the goal state avoiding dynamic obstacles (Moving cars) as well as it considers a safe-overtaking scenario.



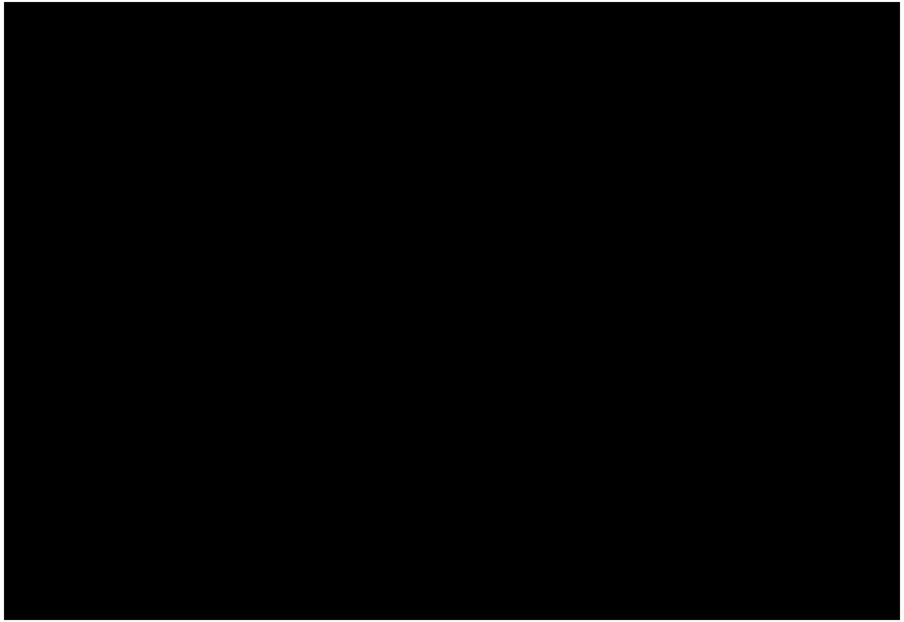
Results | Static and Dynamic Obstacle Avoidance

- Test-case: The parking lots, where an ego vehicle maneuvers itself avoiding both parked cars and moving cars.



Results | Static and Dynamic Obstacle Avoidance

- Test-case: An unstructured environment with randomly initialized static obstacles and the dynamic obstacles are spawned in real-time during the run-time simulation



Challenges

- Setting up the CARLA simulator.
- Choice of computationally efficient algorithm for global and local planners among the state-of-the-art methods.
- Environment setup for spawning dynamic obstacle.
- Designing a method for object collision detection algorithm

Key Conclusions

- Based on the performed experiments and obtained results, we conclude that using a single multi-level planner with a sampling-based global planner and a state lattice local planner, we can achieve efficient path planning in case of both structured and unstructured scenarios.

- It was interesting to observe that using PRM as the global planner, we could also achieve multi-query planning with better accuracy especially for the parking lot scenarios and a complex on-road maneuvering is also possible to achieve effectively using the state lattice as the local planner with a convex hull intersection checking algorithm.

Any Questions?