Motion Planning of an Autonomous Vehicle under Uncertainty

Radha Saraf
Robotics Engineering
Worcester Polytechnic Institute
Worcester, MA
rrsaraf@wpi.edu

Amey Deshpande
Robotics Engineering
Worcester Polytechnic Institute
Worcester, MA
aadeshpande2@wpi.edu

Alexander Jensen
Mechanical Engineering
Worcester Polytechnic Institute
Worcester, MA
ajjensen@wpi.edu

***Abstract*—Autonomous vehicles can get subjected to a lot of uncertainties. These can be external or internal- Dynamic obstacles, roads with sharp turns, bad weather conditions that alter the terrain or affect the sensing capabilities of the vehicle to name a few. The ability of the vehicle to sense, analyse, propose alternate maneuvers and carry them out all in real-time poses quite the challenge. This is an area of active research and various approaches have been proposed to tackle the problem depending on the nature of the uncertainty. As part of our project, we plan to study the existing literature in this direction and aspire to build a system (in simulation) that effectively plans an optimal path under a dynamic obstacle uncertainty.**

***Index Terms*—Real-time motion planning, obstacle avoidance, dynamic and uncertain environment**

I. Introduction

Autonomous Vehicle/robots have improved dramatically over the last few decades and have come a long way from the beginnings in the late 80's, to the DARPA challenges, to the autonomous cars of today, that can autonomously navigate public roads.

These great advances are not only owing to improved hardware but also because of the software stack of the autonomous car. It can be roughly split into four layers: localization, perception, planning, and control. In this paper, we focus on the planning layer, and more specifically on the motion planning problem under uncertainty.

II. Background

Although the field is nascent, there have been significant developments in literature that are focused towards solving this challenge. A varied number of approaches have been taken to deal with uncertainties of different kinds.

Xu, et al in their paper [1] have considered the uncertainty that arises from localization and control of an autonomous vehicle, and sensing and prediction of traffic participants. They have modeled these uncertainties using known frameworks. Because of the direct uncertainty information feedback, their planner has been shown to avoid unsafe situations more efficiently as compared with other safety assessment methods.

Kuwata et al [2] dealt with the uncertainty involved in urban autonomous driving by introducing the CL-RRT algorithm, a sampling-based motion planning algorithm that they specifically developed for large robotic vehicles with complex dynamics and operating in uncertain, dynamic environments such as urban areas. The uncertainty in the environment is captured in the form of a risk penalty in the tree. The novelty comes from the use of closed-loop prediction in the framework of RRT. The proposed algorithm was at the core of the planning and control software for Team MIT's entry for the 2007 DARPA Urban Challenge, where the vehicle demonstrated the ability to complete a 60 mile simulated military supply mission, while safely interacting with other autonomous and human driven vehicles.

Alexander et al [3] have considered the road as an adversary, representing scenarios wherein the autonomous vehicle experiences sudden path changes owing to the path not being visible in the camera horizon. They present a game-theoretic path-following formulation where the opponent is an adversary road model. This formulation has allowed them to compute safe sets using tools from viability theory, that can be used as terminal constraints in an optimization-based motion planner. The planner is guaranteed to follow a path which is not known in advance, but has a bounded curvature, all of that, while guaranteeing that the car is staying within road limits and fulfilling comfort constraints.

Ferguson et al [4] have presented an approach for robust detection, prediction, and avoidance of dynamic obstacles in urban environments. After detecting a dynamic obstacle, their approach exploits structure in the environment where possible to generate a set of likely hypotheses for the future behavior of the obstacle and efficiently incorporates these hypotheses into the planning process to produce safe actions. Based on the literature review and understanding the importance of solving the problem, we are motivated to implement motion planning for an autonomous vehicle under uncertain scenarios.

III. Methods

As our project is regarding Motion Planning of an Autonomous Vehicle under uncertainty, we studied various types of uncertainty an autonomous robot may perceive in its environment as well as in it's system. Path planning and motion planning for a vehicle/robot under uncertainty may face the risk of failure due to unexpected events. To reduce the risk of failure the vehicles has to stay away from failure states. Designing a path and motion planner for an autonomous vehicle/robot in uncertain environments while avoiding static and dynamic obstacles and staying away from risky states is a challenging problem. So, we studied few of the approaches to tackle this problem. Ongoing research topic in the motion planning is identification of feasible paths for autonomous system under many forms of uncertainty which can be categorized into following groups:

- Uncertainty in system configuration.
- Uncertainty in the system model.
- Uncertainty in environmental situations
- Uncertainty in Future environment state.(Dynamic obstacles)

Uncertainty can be linear as well as non-linear. Thus, uncertainty can be modelled by Gaussian processes which helps modelling the uncertainty.

Path planning consists of graph based search methods and sampling based methods. Graph based methods include A*, Dijkstra, D* etc where as sampling based methods include Rapidly exploring random trees, RRT, RRT*.

Sampling based approaches have shown to have several advantages for complex motion planning problems, including efficient exploration of high dimensional configuration space, paths that are dynamic and trajectory wise checking of possibly complex constraints. The RRT algorithm has been demonstrated as successful planning algorithm for complex real world systems, such as autonomous vehicle. However, RRT do not explicitly incorporate uncertainty, they have to extend RRT for incorporating uncertainty of dynamic obstacles and other type of uncertainty.

Thus, we studied following RRT variants that deal with uncertainty:

RRBT

RRT method was extended to Rapidly-exploring Random Belief Trees considering uncertainty. It contains multiple belief nodes with different uncertainty level and path lengths in a graph.

RRBT can preserve individual uncertainty estimates along multiple trajectories passing a particular location. The algorithm builds a graph with a set of vertices, G, and each vertex can have multiple belief nodes.

CC-RRT

CC-RRT* is similar to RRT* in that it is a random sampling-based algorithm that has rewiring for path optimality. However, instead of simply connecting points to one another with straight lines (like RRT), a dynamic model and controller are taken into account. The dynamics model uncertainty in the state and uncertainty in measurements through tracking the state mean and covariance under a Gaussian distribution assumption.

To achieve robustness against uncertainties, we grow a tree of state distributions and ensure that the probability of failure is less than a pre-defined value.

RRT-FND

RRT* FND allows computationally efficient motion planning in uncertain and dynamic environments by preserving valid parts and damaged trees. It incorporates the ”Reconnect” and ”Regrow” methodology for ting trees and sub-trees when a dynamic obstacle comes up.

RRT with motion primitives

Motion primitives are pre-computed motions that the robot can take. By superimposing this on the graph at the robot's position, the adjacent states (or what are usually called successors) are not necessarily spatially adjacent. Instead, they represent what the robot can smoothly transition to.

For RRT with motion primitives instead of extending the tree towards the random configuration along a straight line, we use motion primitives to generate possible new nodes and trajectories which follow the non-holonomic constraints. Motion primitives are generated from the tree node closest to the random configuration, then the end point of motion primitive which is closest to the random configuration is added as the new node and the trajectory as the edge.

RRT* with dynamic window approach for dynamic obstacles

The dynamic window approach is used for reactive collision avoidance for mobile robots equipped with synchro-drives. The approach is derived directly from the motion dynamics of the robot and is therefore particularly well-suited for robots operating at high speed. It differs from previous approaches in that the search for commands controlling the transnational and rotational velocity of the robot is carried out directly in the space of velocities. The advantage of this approach is that it correctly and in an elegant way incorporates the dynamics of the robot. This is done by reducing the search space to the dynamic window, which consists of the velocities reachable within a short time interval. Within the dynamic window the approach only considers admissible velocities yielding a trajectory on which the robot is able to stop safely. Among these velocities the combination of transnational and rotational velocity is chosen by maximizing an objective function. The objective function includes a measure of progress towards a goal location, the forward velocity of the robot, and the distance to the next obstacle on the trajectory.

After review of few methods for motion planning algorithm for dealing with dynamic obstacle uncertainty, we decided to proceed with RRT*+ Dynamic window approach. RRT* is used as global planner and Dynamic window is used as local planner which is very effective in avoiding dynamic moving obstacles. We used Pygame for simulating our results.

IV. Challenges

- CARLA turned out to be an unsuitable environment due to resource constraints.
- While writing planner implementations was straight forward, we had difficulty integrating controls to steer the vehicles and it ended up taking a lot of our exploration time.
- Likewise, matplotlib served to be a very useful tool for rapid testing of planner implementations however, there was a learning curve involved in simulating the (control) motion in pygame.

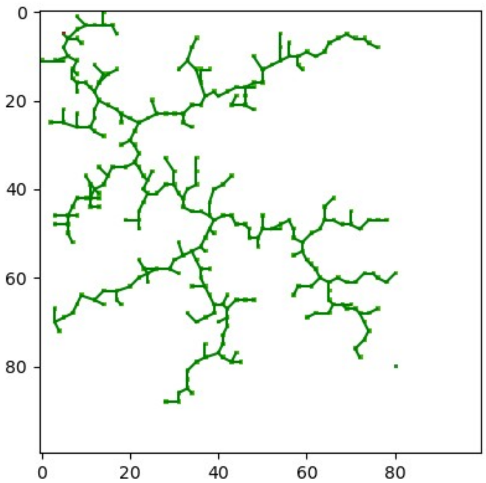

Fig. 1: RRT implemented on a 2D grid

- Integrating motion primitives with RRT/RRT* for road like scenarios was difficult and had to be tuned since sometimes the tree nodes would completely miss the goal as shown in Fig. 2
- We implemented CCRRT and RRBT, but there was no mean of simulating it so, we shifted our approach to dynamic window approach.
- CC-RRT and RRBT were not so easy concepts and were hard to understand and implement.
- CARLA turned out to be an unsuitable environment due to resource constraints.
- While writing planner implementations was straight forward, we had difficulty integrating controls to steer the vehicles and it ended up taking a lot of our exploration time.

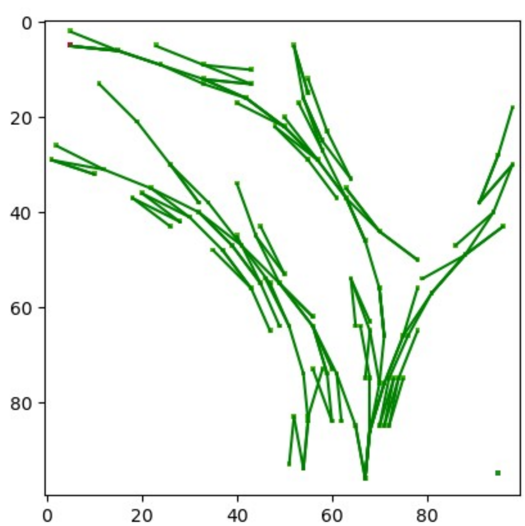

Fig. 2: RRT with motion primitives missing the goal

V. Division of Labor

We divided the labor amongst team members as shown in the table I.

Amey Deshpande	Incorporating and modelling the uncertainty,Research and implementation of Global and local planner, Final simulation, Analyze results, Report
Alexander Jensen	Incorporating and modelling vehicle constraints
Radha Saraf	Global and local planner implementations for simulation, incorporate uncertainty,Research and implementation of planning algorithms, Final simulation, Analyze results, Report

TABLE I: Division of Labor

VI. Results

After trying out few algorithms like hybrid A*, PRM, RRT, RRT*, we decided with fixing RRT* as a base algorithm and implemented vanilla RRT and a couple of variants in the matplotlib environment. We were able to successfully generate an RRT path using motion primitives for road like scenarios and steer a vehicle avoiding static obstacles.

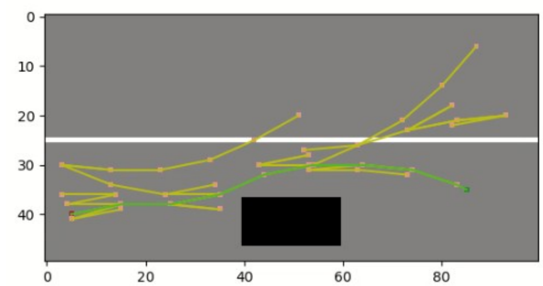

Fig. 3: RRT with motion primitives for a road scenario

For dealing with the uncertain nature of dynamic obstacles we utilized the dynamic window approach. We implemented RRT* as a global planner which returned us the path. We fed the path to Dynamic Window local planner which took care of avoiding uncertain dynamic obstacle and the control for the robot. For simulating uncertain dynamic moving obstacles, we incorporated randomly moving obstacles and made them to collide with our robot/vehicle. We tested our algorithm and our robot with RRT+ DWA, which never collided with any uncertain moving obstacle and was able to reach the goal node successfully.

VII. Conclusion

Thus, we are able to successfully motion plan of an autonomous robot/vehicle under dynamic obstacle uncertainty. The RRT*+Dynamic window approach scaled well with highly dynamic moving obstacles and did not collide with any obstacles. In the future, we would like to incorporate CCRT+ Dynamic window, in which both model and environment uncertainties will be mapped. Moreover, we can use Deep Learning based approach for predicting future trajectories of

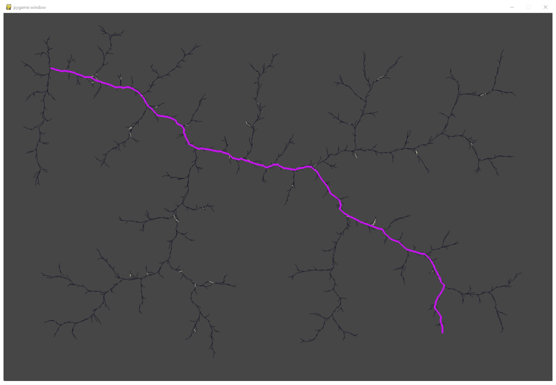

Fig. 4: RRT* as Global Planner

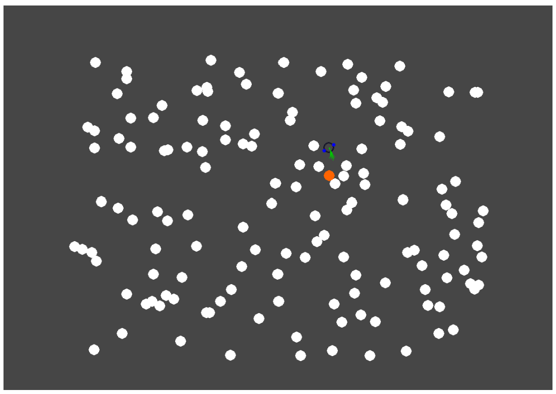

Fig. 5: Dynamic Window as Local Planner under uncertain dynamic moving obstacles

obstacles and plan the motion of the autonomous vehicle so as to reduce the uncertainty.

- Adapt the dynamic window approach for a car like vehicle(Ackerman Steering) to predict other vehicle trajectories.
- Shift the simulation setup to a 3D environment like gazebo/CARLA depending on the drive mechanism.

References

[1] Xu, Wenda, et al. "Motion planning under uncertainty for on-road autonomous driving." 2014 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2014.

[2] Kuwata, Yoshiaki, et al. "Real-time motion planning with applications to autonomous urban driving." IEEE Transactions on control systems technology 17.5 (2009): 1105-1118.

[3] Liniger, Alexander, and Luc Van Gool. "Safe motion planning for autonomous driving using an adversarial road model." arXiv preprint arXiv:2005.07691 (2020).

[4] Ferguson, Dave, et al. "Detection, prediction, and avoidance of dynamic obstacles in urban environments." 2008 IEEE intelligent vehicles symposium. IEEE, 2008.

[5] S. Thrun, W. Burgard, D. Fox, et al., Probabilistic robotics, vol. 1.MIT press Cambridge, 2005

[6] J. Van Den Berg, P. Abbeel, and K. Goldberg, "Lqg-mp: Optimized path planning for robots with motion uncertainty and imperfect state information," The International Journal of Robotics Research, vol. 30, no. 7, pp. 895–913, 2011.

[7] A. Broadhurst, S. Baker, and T. Kanade, "Monte carlo road safety reasoning," in Intelligent Vehicles Symposium (IV), pp. 319–324, IEEE,2005.