Planning and Navigation of Mobile Robots
Independent Study Report
Spring 2022 - Under Prof. Madhava Krishna

Avneesh Mishra
avneesh.mishra@research.iiit.ac.in *

May 8, 2022

This report describes the things learned in the Independent Study during the Spring 2022 session. The material is available on GitHub at TheProjectsGuy/IS_RPN22-EC9.404.

Project Description

This project aims to study mobile robots in the context of planning and navigation in known and unknown environments. The aim is to explore aspects of mobile robotics: Motion models, trajectory generation, optimization, planning, and tracking. Overall, the study aims to discover the underlying mechanics of autonomous mobile systems. The study seeks to emulate the course outcomes of Robotics: Planning and Navigation (EC4.403) while exploring as much additional material as possible.

Contents

*M.S. by research - CSE - 2021701032

1 Summary

1.1 Robotic Planning and Navigation - EC4-403

This is a course that the independent study tried to follow. The following was completed from this course

- Assignment 1: Motion planning using RRT. Official submission at Robotics-Planning-Navigation.
- Assignment 2: Bernstein polynomials. Official submission at Robotics-Planning-Navigation.
- Assignment 3: Time scaling, collision cone. Official submission at Robotics-Planning-Navigation.

Additionally, lecture videos were also seen (as long as they were online and recorded). Lectures videos from `Lecture 3` through `Lecture 14` were seen for this study. The following concepts were covered through watching the recorded lectures

> Visibility Graph - Forward and Inverse Kinematics of differential drive robots - UAV motion control - Model Predictive Controller - Collision checking as an optimization problem - Bernstein polynomials - Time scaling - Collision cone and Velocity obstacle - Time scaled collision cone - Inverse Velocity obstacle

1.2 Extra Material

The following was covered by referring to online sources [1]

> Unicycle and bicycle model - wheel encoders - creating a dashboard using Plotly - RRT - **Dijkstra** from scratch - navigating indian cities using **OSMNX** - Trajectory generation using cubic and quintic spirals - tracking trajectories using **pure pursuit and stanley trackers** - Collision checking using circles and convex hulls (swath) - **dynamic window avoidance**

The above topics are also included in the repository.

[1] AP102 - Motion planning and Path tracking - Naveen Arulselvan

2 Learning Objectives

The following learning objectives were achieved through this independent study

- Modeling and control of different vehicle models
- Global path planning using sampling and search-based planners
- Local trajectory planning using third and fifth order spirals, Bezier curves (Bernstein polynomials), and other methods
- Local trajectory tracking using optimization techniques and various trackers
- Collision avoidance using time scaling, optimization constraints, and other methods
- Solve the assignments of course `Robotic Planning and Navigation - EC4.403`

3 Robotics Planning and Navigation

3.1 Assignment 1: Motion planning using RRT

RRT Algorithm

The RRT algorithm has the following steps

1. **Define map**, free space, and the problem: This will be a map having the start, stop and the free space information of the environment (including a list of obstacles).

2. Keep **sampling** in free space. Store nodes in a tree (can be modeled as a list with links). Each node in the list has x, y, a parent (index), children (indices), and a v, ω command (to reach it from the parent). We start at the starting point, with zero commands and NULL parent.

3. When a new point in free space is sampled, the closest point from the nearest node (in the tree) and in the allowed distance is obtained. The control commands to this point is obtained. The node is added (as a child and to the list) if the path from the nearest node to this node is free of collisions.

4. We repeat the above step will the most recently added node (at the bottom of list) is within a certain bounds of the goal.

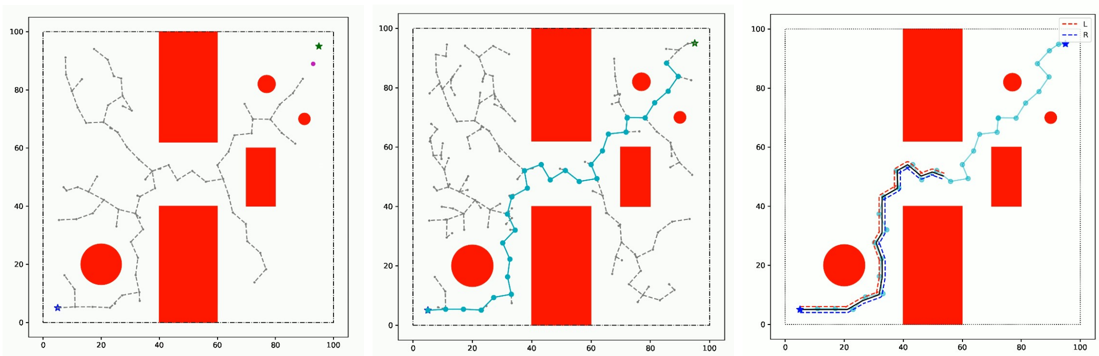
The following implementations were done

- Generating a path assuming that a holonomic robot is going to travel it (do not store v, ω), and then make a non-holonomic robot travel the path. The robot will either go straight or turn at nodes.

  This does not exploit the non-holonomic structure of the vehicle and may not be practical for larger robots. See figure 1.

- Exploiting the non-holonomic structure of the vehicle when adding nodes. This has the advantage of creating paths and routes that can inherently be traversed by the vehicle.

  This involves either directly sampling in the vehicle's control space or running inverse kinematics on the points sampled in the state space (free map). See figure 2.

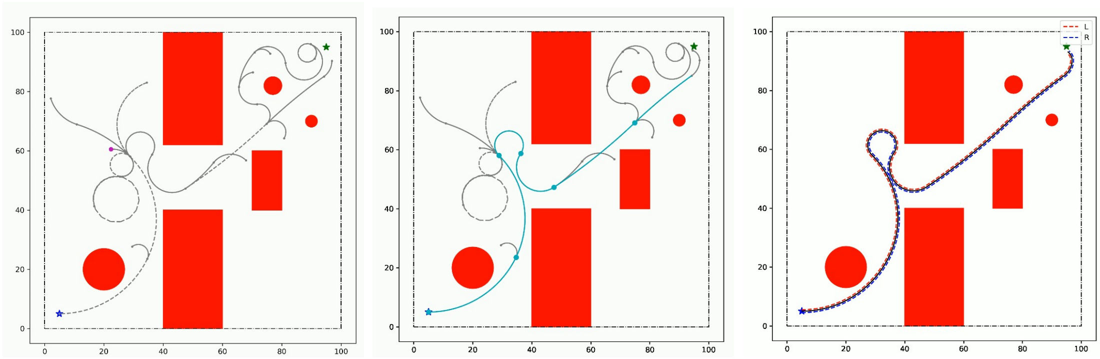


(a) Sampling (b) Path (c) Traveling

Figure 1: RRT: Holonomic sampling and non-holonomic travel
The sampling and path are that of a holonomic model. The non-holonomic robot navigates this using a sequence of straight line travels and turning commands.

3.2 Assignment 2: Bernstein Polynomials

3.3 Assignment 3: Time scaling



(a) Sampling (b) Path (c) Traveling

Figure 2: RRT: Non-holonomic sampling and travel
The sampling, path generation and traveling is of a differential drive robot (non-holonomic robot).

4 Extra Material

4.1 Modeling

4.2 Path Planning

4.3 Path Tracking

4.4 Collision Checking