EE209AS (Fall 2018)

Computational Robotics

Prof. Ankur Mehta
mehtank@ucla.edu

Problem set 4
Due 2pm Thu. Nov. 15, 2018

Objectives

The goal of this lab is to explore build autonomy into a simple 2 wheeled robot. Using your description of the system dynamics from the previous lab, you will develop an RRT-based planner designed to get your robot to a desired goal state.

Deliverables

This project will require you to write code. Make sure the **well commented** code for this lab is committed and pushed, and submit a link to the repository. For some possible resources on git, see the previous pset.

You may work individually or in pairs on this assignment. Each person needs to submit their own solutions, but the team can submit common code. Indicate on your solutions who you worked with, and for each person identify 1) the specific contributions made by each, and 2) an aggregate percentage of the total work done.

Upload your solutions to gradescope.

Preliminaries

0(a). What is the link to your (fully commented) github repo for this pset?

0(b). Who did you collaborate with?

0(c). What other resources did you use?

0(d). What were the specific contributions of each team member?

0(e). What was the aggregate % contributions of each team member?

1 Robot model

You will again consider the 2 wheeled robot from lab 3, this time ignoring sensors and noise terms. For simplicity, you can consider the robot to be a circle of diameter 115mm. For realism, you can optionally consider the true shape of the robot to be a rectangle of length 80mm and width 85mm; the center of the axle is 10mm from the front of the robot (see Fig. 1).

2 Trajectory planning

The robot must move through a cluttered 2D rectangular environment representing a parking lot. You will want to consider geometric obstacles defined by rectangles specified within the space. For additional complexity, you may want to include "one-way" regions in the space that only permit motion in one direction (with motion in the other directions represented by obstacles in C-space).

The robot must achieve a prescribed goal, i.e. a desired x, y, θ for the robot state. Note that this robot is non-holonomic: you only have two input controls for the 3-DOF state.

Compute a plan to take the robot from a given initial state to the desired goal state while avoiding obstacles, and use that to implement a planner / controller specifying the inputs to the actuators as a function of time.

2(a). Given a set of points V in C-space and a single other target point, determine which of the points in V is closest to the target.

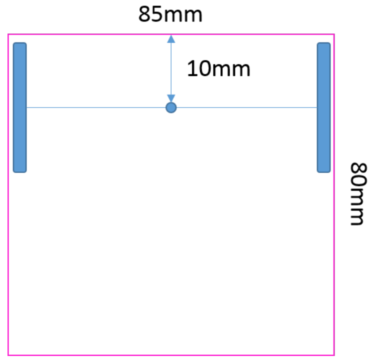

Figure 1: Two wheeled tank-drive robot dimensions

2(b). Given arbitrary initial and target robot states (in C-space), generate a smooth achievable trajectory from the inital state towards the target lasting 1 second. What are the control inputs for this trajectory?

2(c). Translate the map of your environment (with obstacles) into C-space, and create a visualization of this map with initial and goal states indicated.

2(d). Given this C-space map and an arbitrary robot trajectory, determine whether this trajectory is collision free.

2(e). Implement an RRT planner on this map to generate a trajectory from a specified initial state to the desired goal state. Visualize the evolution of the RRT.

2(f). Optionally, improve on this planner using the RRT* algorithm.

3 Evaluation

As in lab 3, you should implement and debug your experiments in simulation.

3(a). Run some examples that demonstrate the performance (in terms of computational efficiency, trajectory efficiency, and obstacle avoidance) of your planner as your robot tries to achieve various goals (such as head-in parking and parallel parking between other such parked vehicles). Clearly describe the experiments that were run, the data that was gathered, and the process by which you use that data to characterize the performance of your planner. Include figures; you may also refer to animations uploaded to your git repo.

3(b). How much relative computational cost is associated with the various operations of the RRT planner?

3(c). Qualitatively describe some conclusions about the effectiveness of your planner for potential tasks your robot may encounter. How might you improve it?