

EE209AS (Fall 2019)

Computational Robotics

Prof. Ankur Mehta
mehtank@ucla.edu

Lab 2
Due 4pm Mon. Nov. 4, 2019

Objectives

The goal of this lab is to explore build autonomy into a simple two-wheeled non-holonomic robot. 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 in teams on this assignment. As a team, you will create a well documented git repository containing all your code and data. You will also create a team writeup describing the mathematical formulation, experimental setup, and experimental results.

As an individual, you will create your own one-page slide summarizing the key contributions of your personal efforts primarily using equations and/or figures. You will be assessed on both the clarity and completeness of your content.

Include a link to your code repository / website in your report, as well as a complete list of references you've used and in what manner, and submit pdfs by 4pm Mon. Nov. 4, 2019.

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 was the aggregate % contributions of each team member?

1 Robot model

You will consider a two-wheeled robot similar to the one shown in Fig. 1. It has two wheels of radius $r = 25mm$, separated by a distance $w = 90mm$. It drags a tail for stability, at a distance $l = 75mm$ behind the centerpoint of the wheels. The position of this robot in the environment is defined relative to the centerpoint of the wheels.

Each wheel is powered independently by a continuous rotation servo, with the angular velocity of the wheel controlled by a PWM signal from the microcontroller. Assume each input prescribes the respective angular speed of the wheel, from -60 to +60 RPM. This allows the robot to drive forwards or backwards at variable speed, or turn with any turning radius.

- 1(a). What is the dimensionality of the input (action) space, robot state (configuration) space, and operational space?
- 1(b). Define the (continuous space) system model. Assume a fully observable system, i.e. the state itself is directly accessible with no sensor model.

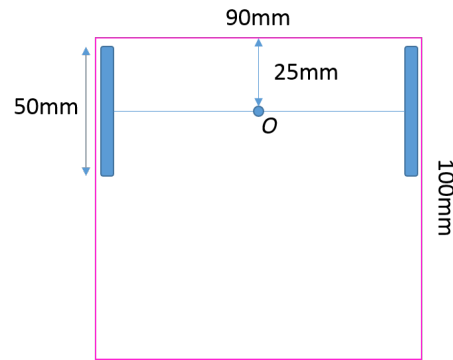
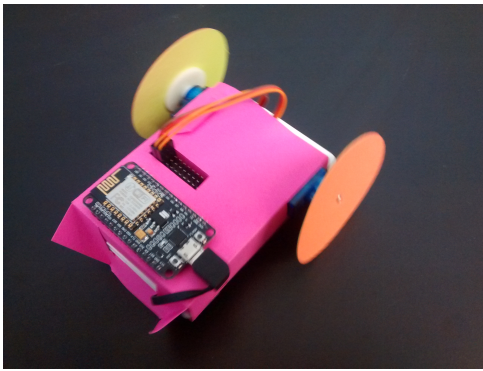


Figure 1: Two wheeled tank-drive robot, with dimensions as specified

2 Trajectory planning

The robot must move through a cluttered rectangular environment representing a parking lot, ultimately achieving a prescribed goal, i.e. attaining a state within a desired region in C-space. You will consider geometric obstacles defined by rectangles specified within the operational space. For an additional challenge, you may also include “one-way” regions in the space that only permit motion in one direction.

Implement an RRT based algorithm to take the robot from a given initial state to the desired goal state while avoiding obstacles, and use that to simulate 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 target point x_t , define and justify a metric that can be used to determine which of the points in V is closest to x_t .
- 2(b). Given arbitrary initial robot state x_i and target robot state x_t (in C-space), generate a smooth achievable trajectory (that matches the metric you defined above) from the x_i towards x_t lasting 1 second. What are the control inputs for this trajectory?
- 2(c). Given a smooth robot trajectory in C-space and obstacles defined in operational space, determine whether this trajectory is collision free.
- 2(d). Combine the above to implement an RRT planner generating a trajectory from a specified initial state to a desired goal region. Visualize the evolution of the RRT.

3 Evaluation and Extensions

- 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). Improve on your planner using the RRT* algorithm, and compare to your original RRT planner using the above metrics.
- 3(d). Qualitatively describe some conclusions about the effectiveness of your planner for potential tasks your robot may encounter. For example, what happens to your planner in the presence of process noise, i.e. a stochastic system model? How might you modify your algorithm to better handle noise?