EE183DA (Winter 2018)

Design of Robotic Systems I

Prof. Ankur Mehta
mehtank@ucla.edu

Lab assignment 3
Due 5pm Tues Feb. 27, 2018

1 Lab Overview

1.1 Objectives

In this lab, you will implement the computational infrastructure necessary to command a simple 2 wheeled robot. You will derive a mathematical input-output model of the system dynamics, and build on to it sensor and actuator responses. You will then implement this model within a computational environment, and use it to build a state estimator.

You will be working in your project teams. You will be responsible as a team for dividing the various tasks of this project between all members. Your grade will be based both on team and individual performance.

1.2 Deliverables

As a team, you will submit a documented git repository of your code, along with further documentation describing your mathematical formulation, algorithms, experimental setup, experimental results, and final performance. This additional documentation can take the form of either a lab report as in lab 1 or academic poster as in lab 2. You will be assessed on both the clarity and completeness of your content. Upload a link to your code repository / website along with a pdf of your report / poster on CCLE by 5pm Tues Feb. 27, 2018.

As an individual, you will submit a participation questionnaire indicating the contributions of each team member to the team results.

Submissions that are up to 24 hours late will be accepted for a 10 percentage point reduction in final grade. No submissions will be accepted more than 24 hours late.

2 Lab specification

2.1 System model

You will consider a 2 wheeled robot similar to the one shown in Fig. 1. It has two wheels of radius $r = 20mm$, separated by a distance $w = 85mm$. It drags a tail for stability, at a distance $l = 70mm$ behind the centerpoint of the wheels. These dimensions are approximate, and may be different based on manufacturing tolerances.

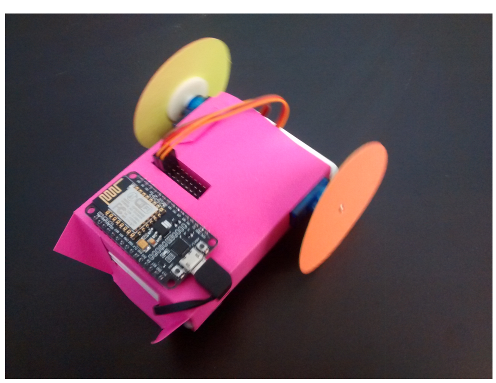

Figure 1: Two wheeled tank-drive robot

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. This allows the robot to drive forwards or backwards at variable speed, or turn with any turning radius. There may be slippage between the wheels and the floor.

You will add onto the robot laser range sensors and an IMU for extrinsic position sensing. The output of these sensors will be a function of the positional state of the robot within its environment.

The robot will be driving within a rectangular environment consisting of 4 walls bounding an open space.

2.2 Hardware

- ESP8266 wifi microcontroller + motor driver breakout
- 2x: FS90R continuous rotation servos
- 2x: GYVL53LOX laser range sensors
- MPU9250 IMU
- Paper robot body
- LiPo battery

2.3 Mathematical setup

The state of your robot will consist of the 3DOF pose of the robot in 2D space. You may also want to include velocity terms. In addition, your state may include sensor bias terms if necessary. Thus, your robot state will have at least 3 variables, but may have more.

Your control input will be the PWM values you send to each wheel, for a total of 2 input variables.

Your sensor output from the laser range sensors will consist of the distance to a wall in a straight line in front of the robot and the distance to a wall in a straight line to the right of the robot. The IMU will return an absolute bearing indicating the angle of the robot with respect to magnetic north, and can also return an angular rate measurement. Thus the robot system will produce 3 or 4 output values.

You will need to derive and present the mathematical equations describing this robot plant, including the appropriate sensor and actuator models and system dynamics. Be sure to clearly define and describe all variables and equations, and produce illustrative diagrams as necessary. Include realistic noise terms into the model as appropriate.

2.4 State estimation

You are free to choose any state estimation algorithm you'd like to compute a state estimate from the time series of PWM commands and sensor measurements. You may want to implement a model based state estimator, or perhaps implement an Extended Kalman Filter (EKF). Other options include an Unscented Kalman Filter (UKF) or a particle filter, or even a neural network. Be sure to explain which algorithm you chose and why, and describe the mathematics behind it in your writeup. Identify any algorithm parameters that need to be set, and explain how you selected the values of those parameters.

2.5 Experimentation

You should plan out and debug your experiments in simulation, but then test and analyze results on a real life robot.

Within the simulation, you should implement approximate models of your sensor and actuator response (especially including noise) to generate simulated sensor measurement data given arbitrary control inputs. Generating these models may require some data collection from physical sensors and actuators.

When doing physical experimentation, you will need to come up with a communication scheme to exchange control commands and sensor measurements between the robot and your state estimator. Sample skeleton code can be obtained from: `https://git.uclalemur.com/mehtank/paperbot/`.

Run some examples that demonstrate the performance (in terms of accuracy and efficiency) of your computed state estimate over time as the robot is issued various commands. 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 state estimator. Include pictures and links to videos.

What can you conclude about the usability of your state estimator for potential tasks your robot may encounter?