ECE183DA (Winter 2020)
Design of Robotic Systems I

Prof. Ankur Mehta
mehtank@ucla.edu

Lab assignment 1
Due 3pm Thu. Jan. 23, 2020

1 Lab Overview

1.1 Objectives

In this lab, you will build, characterize, and simulate a simple instrumented 2 wheeled robot.

You will generate a mathematical input-output model of its system dynamics, and build on to it data-driven sensor and actuator responses grounded in physical experimentation. You will create an interface to a simulator, and compare the results of the simulator to real life.

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 create a well documented git repository containing all your code and data. You will also create a team writeup describing your mathematical formulation, computational implementation, experimental setup, experimental results and data, and conclusions. Include in your writeup links to your code repository / documentation, as well as a complete list of references you've used and in what manner.

As an individual, you will create your own one-page overview slide summarizing the key contributions of your personal efforts, primarily using equations and/or figures. This should be a standalone document that graphically communicates your individual key contributions in the context of this lab.

For both of these deliverables, you will be assessed on both the clarity and completeness of your content. Submit pdfs on CCLE by 3pm Thu. Jan. 23, 2020.

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 overview

You will build a two-wheeled robot similar to the one shown in Fig. 1. It has two wheels of diameter $r \approx 50mm$, separated by a distance $w \approx 90mm$. Each wheel is direct driven from a continuous rotation servo. It drags a tail

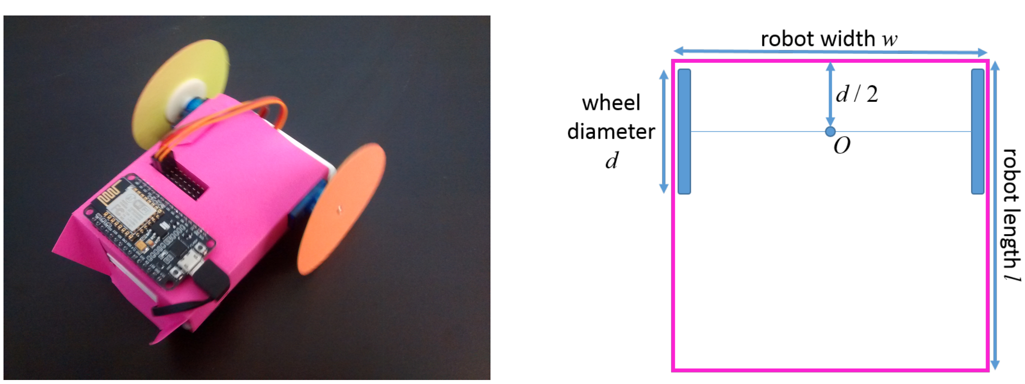

Figure 1: Two wheeled tank-drive robot, with actual dimensions to be measured from the physical artifact.

for stability, at a distance $l \approx 75mm$ behind the centerpoint of the wheels O. The position of this robot in the environment is defined relative to this centerpoint O. These dimensions are approximate, and may be different based on manufacturing tolerances.

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 relative to 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: GYVL53L0X or GYVL53L1X laser range sensors
- MPU9250 IMU
- Paper robot body + wheels
- LiPo battery

2.3 Hardware interfacing

When doing physical experimentation, you will need to issue control commands and extract sensor measurements during trials, and record these values for subsequent data analysis. Sample skeleton code for interacting with the various subcomponents (which you will need to adapt and extend) is available on CCLE and at the git repository:

`https://git.uclalemur.com/mehtank/paperbot`

2.4 Actuation model

Each wheel is powered independently by a continuous rotation servo—part number FS90R—with the angular velocity of the wheel controlled by a PWM signal from the microcontroller. The control input to the robot hardware will be the PWM values you send to each wheel, for a total of 2 input variables. This allows the robot to drive forwards or backwards at variable speed, or turn with any turning radius.

The mapping from PWM to rotational speed may include nonlinear effects including a deadzone and saturation. There may also be slippage between the wheels and the floor. Use physical experimentation to determine the actuator response and noise models.

2.5 Sensing model

You will install onto your robot two laser range sensors—either part number VL53L0X or VL53L1X—and an inertial measurement unit (IMU)—part number MPU9250. The output of these sensors will be a function of the state of the robot within its environment.

Mount the laser range sensors onto your robot such that you measure 1) the distance to a wall in a straight line in front of the robot, and 2) the distance to a wall in a straight line to the right of the robot. The IMU will return 1) a measurement of the in-plane rotational speed from a angular rate (gyro) sensor, and 2) the components of the measured magnetic field along each of the 2 in-plane coordinate axes, which can be used as a compass for absolute orientation relative to Earth's magnetic field. We will ignore the out-of-plane gyro and magnetometer axes, as well as the accelerometer on the IMU. Thus the robot hardware will produce 5 output values.

Use the data sheets for these specific sensors along with physical experimentation to determine their sensor response and noise models.

2.6 Mathematical formulation

The state of your robot will satisfy the Markov property, capturing the complete history of actuator inputs (and noise!) to the robot hardware, allowing for computation of the dynamics update as well as all of the sensor measurements. Define this state, and then write out analytic mathematical models for the system dynamics and measurement processes, starting with an ideal theoretical model based on fundamental principles then including terms for noise where appropriate. Use empirical data from your experiments to quantitatively characterize the noise terms that you have included.

Be sure to clearly define and describe all variables and equations, and produce illustrative diagrams as necessary. Clearly describe the experiments that were run, the data that was gathered, and the process by which you use that data to mathematically formulate the robotic system. Include pictures and links to videos.

2.7 Simulation

Implement this mathematical model as a computational simulation, then compare the results of your simulator to the real life robot that you've built. The best comparison will involve sending identical control signals to both the simulation and the real robot—now as well as in future labs—so you will likely want to devise a common programmatic interface to both, and invoke both simultaneously.