# ECE183DA (Winter 2020)
## Design of Robotic Systems I

Prof. Ankur Mehta
mehtank@ucla.edu

**Lab assignment 2**
**Due 3pm Thu. Feb. 6, 2020**

# 1 Lab Overview

## 1.1 Objectives

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

The goal of this lab is to explore Kalman Filtering (KF) to estimate the state of your two-wheeled robot. You will build upon the results of your previous lab: you will use your mathematical model to derive a KF-based state estimation algorithm, then implement and validate that estimator against both simulation and physical experimentation.

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. Feb. 6, 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

The hardware system as shown in Fig. 1 is the same system as in Lab 1.

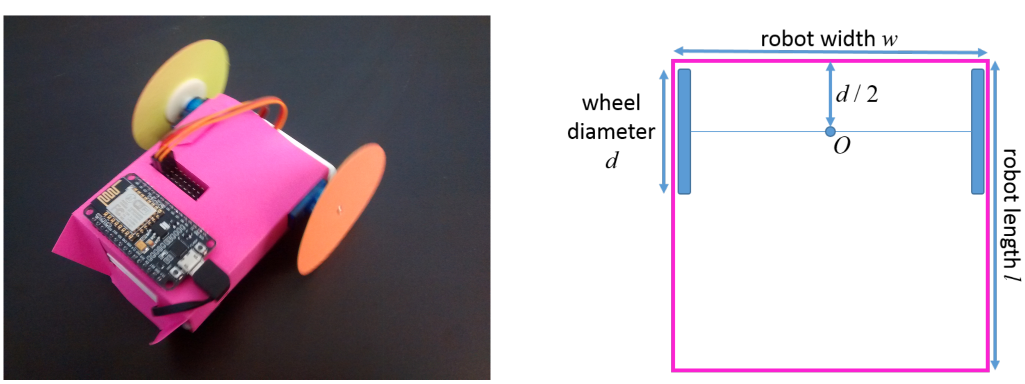


Figure 1: Two wheeled tank-drive robot, with actual dimensions to be measured from the physical artifact.

## 2.2 Extended Kalman Filter

You will build a state estimator, which is a computational system that takes as input both the input and output of your robot system, and produces an output which is the estimated state of the robot system. Since the robot system is nonlinear, you will need to use a variant of the original linear KF, namely the Extended Kalman Filter (EKF), for the state estimation.

Show the mathematical framework for your EKF, then implement the algorithm in code. You may not use any predefined robotics or state estimation packages—the goal is to implement the EKF from mathematical fundamentals.

The EKF procedure employs a computational model of the robot system for its calculations, so you may want to modularly reuse components of your computational simulator from before. Keep in mind the key differences between the model as it exists within a KF and the model as it is used for simulation purposes. Also note that the EKF requires generating linearized system matrices at the setpoint in addition to the other mathematical model quantities, so your model code will likely need to be augmented to handle that.

## 2.3 Evaluation

Run some examples that demonstrate the performance of your computed state estimate over time as the robot is issued various commands. That is, drive your robot around and collect time series data of both the input commands as well as the resulting sensor output for a variety of trajectories. Run these data streams through your EKF to get the resulting state estimate.

Compare the resulting state estimates to the true state as measured by an external observer, such as a birds-eye camera. Include an analysis of the uncertainty as quantified by the estimate covariance. Also compare those values to the open-loop evolution of the modeled state (based on your mathematical model/computational simulator from Lab 1), both as a check on your algorithms and simulations, but also to evaluate the value of the closed loop state estimator.

Consider the case where you have perfect knowledge of the initial robot state as well as the case where the initial state is entirely unknown.

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 figures; you may also refer to videos and animations uploaded to your git repo.

Qualitatively describe some conclusions about the effectiveness of your state estimator for potential tasks your robot may encounter. How might you improve it, why would that generate an improvement, and what are the tradeoffs?