MeEn 595R – Autonomous Systems
Team Homework #1

We have taken a data set of the TurtleBot robot driving around the motion capture room in the MAGICC Lab. The TurtleBot is equipped with a camera and software to produce range and bearing measurements to ArUco tag markers that were strategically placed along its path around the room. You will be provided with the locations of the ArUco landmarks as well as the initial conditions for the TurtleBot. We also will post a movie of the TurtleBot's run that will give you a sense of the ArUco range and bearing measurements available during the run. From the motion capture system, we also have truth data for the robot pose provided in the truth_data.mat file.

The task for your team is to take one of the localization methods that we've studied and to apply that method to the TurtleBot experimental data to localize the TurtleBot in the provided MAGICC Lab map.

The TurtleBot odometry and measurements data file (processed_data.mat) has the following data:

Variable name	**Description**
landmarks	x and y locations of landmarks
l_time	time vector corresponding to landmark measurements
l_depth	range (depth) measurements from vehicle to landmark
l_bearing	bearing measurements from vehicle to landmark
odom_t	time vector corresponding to odometry and velocity commands
pos_odom_se2	x, y, and heading position odometry from robot
vel_odom	velocity and angular velocity of robot

The TurtleBot pose truth data file (truth_data.mat) has the following data:

Variable name	**Description**
t_truth	time vector for pose truth data
x_truth	x position truth
y_truth	y position truth
th_truth	heading angle truth

Initial conditions for the robot are x_0 = 2.5 m, y_0 = –1.7 m, and θ_0 = 10 deg = 0.175 rad.

Tasks:

1. Plot the x-y position odometry of the robot with the landmarks. Watch the posted video and confirm that the video and odometry correlate.
2. Implement a localization filter of your choice to estimate the position of the robot. The input to your filter should be the velocity and angular velocity of the robot as determined from the robot's motor encoders. Choose from EKF, UKF, MCL, or EIF. Let's try to get a diversity of solutions from the five teams in the class.
3. Create plots of your state variables versus time as well as a plot of the robot trajectory on the landmark map.

Hints:

- Use the time information from the odom_t vector as the clock for your simulation. In this way, your velocity commands will be applied at the appropriate times. Your time steps will not be uniform in size so you will have to calculate and apply a different dt at each time step.

- Your landmark data will not by synced in time to the commands, so you will have to sort through the data to determine when to apply the landmark measurements. Data in the l_bearing and l_depth matrices is organized with one column for every time step in the l_time vector and 9 rows, one for each landmark. At times corresponding to when a landmark was seen, range and bearing measurements show up accordingly in the l_depth and l_bearing matrices. If a landmark was not seen at a given sample time, a NaN shows up in its place.
- Instead of the velocity motion model in the book, I used a standard unicycle model. I was simpler, more numerically stable (no problems with zero angular velocity), and gave identical results due to our relatively small step sizes. I made modifications to the *G* and *V* Jacobians of the EKF accordingly.