ME 132B - Odometry

Released: Wednesday 04/19/2017 Due: Wednesday 04/26/2017

In part b of the first lab for ME 132B you will be writing a script to command a TurtleBot (Fig. 1) to trace out a simple shape and return to its starting position. During the test the TurtleBot will be logging readings from the encoders on each of its wheels, which you will convert into Cartesian position using the differential vehicle equations and compare with the path you intended it to follow. At the end of the test we will physically measure the error in the TurtleBot's position in order to gauge the accuracy of the odometry.

Lab Preparation

Appointments for the hardware demonstration can be made via the doodle poll that will be sent out via email and under announcements on the course website. Before your scheduled demo time you will need to add code to a script to make the TurtleBot trace a simple pattern that brings it back to rest in the same location as it started within 20 seconds. The template for this script can be downloaded here. Try to keep the TurtleBot from venturing more than 1m away from the starting point, along with linear motion $\leq$ 0.3 m/s and angular $\leq \pi/2$ rad/s.

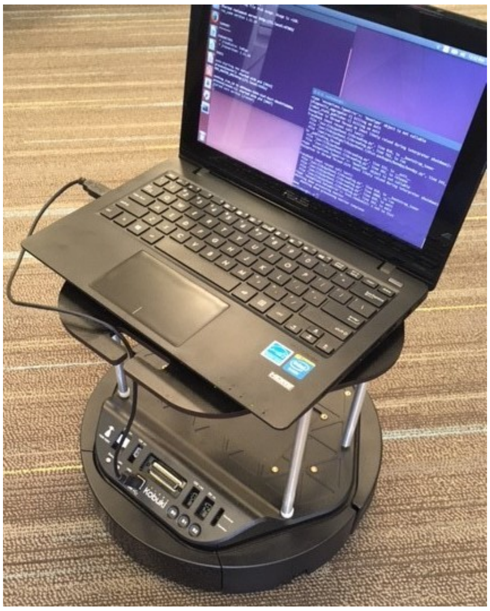

Figure 1: TurtleBot 2 based on the Kobuki platform

Hardware demonstration

Show up to your scheduled time slot with your modified python script ready to be transferred to the TurtleBot's notebook. The TurtleBot will be placed on a marked starting position and then your script will be run, moving through whatever pattern you designed.

After the program finishes, we will measure the X and Y offset of the TurtleBot from its initial position for you to compare to your odometry results. You will take a copy of the csv file on the

TurtleBot notebook which contains encoder readings from the two wheels over the course of the program.

Odometry Data processing

Use the kinematic equations from the class handout on differential drive vehicles to calculate the XY position and orientation of the TurtleBot over the course of your program, and plot the results as you did in lab 1a. Equation 27 on page 6 will allow you to calculate the new XY position and orientation at each timestep.

The values in the csv you receive after testing give the number of 'ticks' seen by the encoder on each wheel. There are 2578.33 ticks per wheel revolution which can then can be multiplied by 2π to give ϕ, with a wheel radius of $\rho =$ 3.5cm. The encoder values are 16-bit numbers which means if they exceed 65535 they will roll back to zero, which you may need to account for. The distance between the wheels or 'wheel base' is 2W = 23cm.

The encoder readings are taken at a rate of 50Hz, but when using the backward difference equation the time step is not needed as the relation is purely kinematic.

Homework Deliverables

Submit plots of the XY position and orientation of the TurtleBot along with the code/equations you use to calculate them. Comment on how the path plotted by the odometry compares with the path the TurtleBot actually traced out, and the accuracy of the position measured at the end of the program.