Lab 3
Zeta Group:

Justin Evaristo
Aditya Malik
Ben Poreh
Andrew Wood

This lab was focused on running code developed in our third programming assignment in the real world on a turtlebot platform. Our code involved basic path following using the ROS client and service paradigm, it works as follows:

1) The client waits to make connection with the service. When a connection is made, the client uploads a path defined as a series of poses (position coordinates and orientation), and waits for the service to execute that path i.e. move the turtlebot through the desired sequence of poses.

2) The service accepts connections from clients. When a connection is made, the service receives a sequence of poses, computes the rotation and movement commands to get the turtlebot from one point in the sequence to the next point. It performs these computations at once, and then sends these commands down to an execution environment that will, for each command, and given the speed parameters of the turtlebot, execute the commands for certain amounts of time until it is assumed the turtlebot is at the desired location.

In practice, this style of execution does not work well. While running on a turtlebot, we tasked the turtlebot to move in a square, but quickly found that the turtlebot did not follow these commands correctly. This occurs for a series of reasons which all boil down to the assumptions made when controlling a robot in a “bind” manner (i.e. no check to the success of commands) does not match the real world very well. There are several sources for this error;

1) robots in the real world do not move at commanded velocities instantaneously, and do not stop moving instantaneously

2) Motors have error, just because they are told to spin at a specific rate, the may or may not actually spin at that rate.

3) Robots do not move exactly as intended in the real world. Phenomena like wheel slip, or non-flat surfaces contribute to robots behaving differently than the code governing their behavior assume.

This resulted in our turtlebot not tracing out a square at all, missing both commanded distances as well as desired turning angles drastically. Our video can be found at https://www.youtube.com/watch?v=Xq3F3SB3rWM and our code can be found at https://github.com/aew61/eecs376.