Lab 1 Report

Getting used to the environment is the first step in designing with new software, this is no different in our pursuit of controlling mobile robots. With the option to use a simulator technology rather than the *expensive* physical robot hardware known as the USF "Robobulls-2018" I decided to work with the simulator early on. From experience I know that understanding and getting comfortable with any unfamiliar technology takes time and patience and with that in mind I press forward. The purpose of this first lab was to familiarize ourselves with our robot system of choice as well as understand how our robot deals with motor control and encoder feedback. To understand the way in which robots control their movement I will be using a software called Webots, an open-source robot simulator by Cyberbotics of Switzerland. There are many robots already set up and designed in the software available to new users to show what it is capable of, but I simply worked with a small mobile robot called an e-puck for Lab 1.

The e-puck is a small 2-wheeled differential drive robot with one left and one right rotational motor. Despite getting to play with several high-speed robots and even a flying drone, the e-puck we're working with has weak little motors with a maximum velocity rotation of 6.28 rad per sec. The encoders for the robots are what can tell us how fast the robot is moving by reading feedback from the spinning motors. In the case of the physical robot there are small holes on the side of each wheel that measures how many times the light flicks through. For the simulator's e-puck things are done with function libraries using position sensors at the hinge joints. Using the motor. SetVelocity() function allows for control of motor velocity and the sensor.getValue() function reads off data collected by the PositionSensor.

[Task 1 – Motion Control]

The first given task of the lab is to test our knowledge of motor function. We are tasked with sending the robot forward or backward in a straight line at user supplied distances and speeds. The user should be able to choose the robot Velocity(V) and the Distance(X) in inches that they wish the robot to travel, and the robot should comply. The trouble I had with this task is understanding how Webots uses their functions and attempting (and failing miserably) to convert inches to meters. The conversions are simple enough but in practice the robot was not driving nearly as far as I wanted. In the end I changed the floor map a 1 x 1 meter square and when I told the e-puck to drive 39.3700787 inches straight, it stopped right at the edge of the floor. Fishy. Either my imported world was tweaked to use inches for the sensor feedback, or my calculations were pure luck with unit cancellations.

[Task 2 – Encoder Reading]

Task 2 is no longer a simple backward or forward movement rather it requires the e-puck to move in a circle. Given a Radius(R1) and Velocity(V) the e-puck should be programmed to loop in a circle at the given radius and velocity. The trouble I had with this task was stopping the robot after a full circle was driven to calculate a specific time. Having one motor faster than the other and keeping those velocities constant makes a perfect circle whose radius can be calculated. Using the given radius, it is easy to find the velocity of each individual motor, but it was important to make sure that the program knew when the new calculated velocities were within the possibilities of the e-puck motor. Anytime the velocity exceeds the range [-6.28, 6.28] I should break to an error message.

As with any new software the grind to understand its nuances can be tricky and is never quite over. Months on I imagine I will look back at Lab 1 and chuckle at the lack of knowledge I had of the power of the software at my fingertips! Already being able to run simulations of drones, a complex lizard robot, and even a famous Boston Dynamics 'creature' I understand there is a lot further we can go with these technologies when given the time, patience, and creativity. I am still having trouble understanding what units of measurement the sensors and motors use because based on the calculations I did in converting from inches to meters the robot was not doing what was expected at all. Velocity functions are in radian per second for rotational motors and meter per second for linear motors. For the inputs we are given inches and inches per second.

Calculations