COMP470: Mobile Robotics

Spring 2020

Lab 3: Dead Reckoning

The purpose of this lab is to explore the idea of Dead Reckoning and the errors that we can expect if we use this navigation technique. Each team will receive perform a series of experiments to quantify the dead reckoning accuracy of their robots. We will then compare the results of these experiments across all the teams.

Per Wiki Article: "In navigation, dead reckoning is the process of calculating one's current position by using a previously determined position, or fix, and advancing that position based upon known or estimated speeds over elapsed time and course."

Lab Procedure: Part 1

1. Create a new Arduino sketch that by duplicating the sketch used in Lab1. Rename this sketch YourNames_Lab2.ino.

2. Modify your sketch to make your robot drive straight forward for 1 meter then stop. Note that the functions available to you allow you the set the voltage to the motor to a value between 0 and 100 percent. It is up to your team to pick a voltage and determine how fast your car drives at that voltage (m/sec). Also note that there is wheel slippage when you suddenly start or stop the motors. You may want to consider ramping up your speed. How will that change your calculation for stopping at 1 meter distance.

3. Create another Arduino sketch that makes your robot rotate by 90 degrees. To get this to work you will need to control the voltage to each side of the robot. Look at the header file Hercules.h and you should see two functions called `setSpeedDir1` and `setSpeedDir2` which allow you to set the voltage to each side of the robot.

4. Run a few experiments to tune your code from steps 2 and 3 until you are confident that your code can perform these tasks consistently.

5. Combine these behaviors to performa sequence of movements.

```
1. Drive Forward 1 meter
2. Stop for 5 seconds
3. Rotate 90 degrees
4. Stop for 5 seconds
5. Drive Forward 2 meter
6. Stop for 5 seconds
7. Rotate 90 degrees
8. Stop for 5 seconds
9. Drive Forward 1 meter
10. Stop for 5 seconds
11. Rotate 90 degrees
12. Stop for 5 seconds
13. Drive Forward 2 meters
14. Stop
```

1. Run an experiment where you repeat this sequence of movements from the exact same starting point 10-15 times. Be sure to always start the robot from the same position and orientatin each trial.
2. Each time the robot stops place a sticker on the ground where the robot is sitting.
3. After completing the 10-15 trials measure the position of each sticker.

Lab Procedure: Part 2

In future labs we will have competitions were each group will need to perform a desired task in a time trial race format. Take make your life easier later on I would like for you to create a couple functions that will allow you to quickly modify your speed to perform basic driving tasks.

1. Write a function that takes two arguments (speed and distance) that will make the robot drive forward for the provided distance at a speed of the provided speed.

2. Write a function that takes two arguments (speed and angular distance) that will make the robot rotate for the provided angular distance at an angular velocity of the provided speed.

3. Test your code for longer distances. In your writeup you should discuss the following questions. Does the time to accelerate up to "speed" influence your errors in positioning? If so, how could you adjust your function to deal with this issue?

Lab Writeup

Write a lab report detailing your procedures. Include plots (I recommend MATLAB) of the positioning error at each step of the sequence. Include analysis of this data and any conclusions that you realized about dead reckoning. Also include your source code in your lab writeup.