CSE 360 - Programing the FreeNove Robot

Karen Li - https://github.com/kkl224/MobileRobotics

I. Introduntion

The objective of this lab is to work with the FreeNove Robot and write a Python script that instructs the robot to form a square trajectory, display different colors of LED, and beep the buzzer. The final program will consist of 4 main functions: *forward(t)*, *rotate()*, *light(index, g, b, r)*, and *buzzer(t)*, and some of them will be called multiple times. The goal is to form a ”perfect” square where the final destination is as close as possible to the starting location. The foreseen challenges include moving forward in a straight line and rotate exactly 90 degrees.

II. Description of the functions

- *forward(t)* - Moves forward when all 4 wheels are set to the same positive speed. Motion lasts for *t* seconds.
- *rotate()* - Rotates approximately 90 degrees, to the direction of the side of wheels that are set to the same negative speed whereas the other side of the wheels are set to the same positive speed. Default direction of rotation is set to left.
- *light(index, g, b, r)* - Displays the indexed LED with the color of the (g, b, r) value.
- *buzzer(t)* - Beeps the buzzer for *t* seconds

III. Answers to the Questions

1. Does your robot form a square trajectory? (yes/no)

Yes, my robot does form a squarish trajectory, but not a perfectly enclosed square.

2. How far is your robot from the starting location?

My robot is approximately 0.2 meters away from the starting location.

3. Describe three ways to improve the accuracy of the robot trajectory. It can be improvements in hardware or software.

- Hardware improvements
  - The 4 DC speed reduction motors connecting to the car board could be loosely screwed. That will introduce errors when the car is moving at a fast speed. And the error will be magnified as the displacement of the car increases. Therefore, a possible hardware solution could be checking the connection of the DC motors to the board and tighten the screws if needed.
- Software improvements
  - One of the software solutions could be decrease the motor speed so that the car move slower and have less displacement within the same time frame. Like the reasoning stated in the hardware solution, reduced displacement means reduced magnitude of error. Therefore, moving at a slower speed will be an easy and intuitive way of increasing the accuracy of the robot trajectory.
  - Another software solution would be adjusting the speed of each motor to offset the errors to achieve a relative straighter forwards motion. It is rather difficult to determine the hardware issues of the robot, since there are to many parts and components to exam. Thus, allocating the parameters of *setMotorModel()* is a more reliable and feasible solution to reduce hardware errors.

IV. Link to the recorded video

For further reference see url: https://drive.google.com/drive/u/0/folders/10FM9CwagYL-kOuy7-5ChBe9Kex7KyDwf