Lab 1 – Open Loop Control- Part 1

Overview:

Two ways to categorize robot control is open and close-loop control. In open-loop control, the robot system receives no feedback or information about the environment. As it is performing its actions, it is essentially blind. The advantage of open-loop control is that the robot does not need sensors and the controller does not require much logic. In closed-loop, the robot has access to sensor information and feedback to inform its actions. Closed-loop control is theoretically more robust, but also requires more components (sensors) and programming logic.

For Lab 1 and 2, we will be focusing on open-loop control. Open-loop control requires that the environment and the correspondence between robot commands and the physical movement is known. In the Labs you will explore different ways of obtaining that correspondence.

Objectives:

In this Lab, you will experiment with different motor commands and timing in order to move and rotate in precise ways. You will also analyze how accurate and repeatable does robot movements are. With an understanding of the robot's movements and errors, you will program it to drive a simple course.

- Gain familiarity programming and working with the EV3 robots.
- Obtain the correspondence between motor commands and robot actions.
- Perform statistical analysis on how close those commands produce the expected movement.
- Use the calibrated motor commands to program the robot to traverse a course with known dimensions.

Procedure

1. Setup
    a. Follow the Installation, EV3 Brick, and Creating and running programs instructions found on the ev3-micropython webpage (https://pybricks.com/ev3-micropython/). You can ignore the instructions for the SD card as that is already done for you. Follow the instructions up to and including running a single motor.
    b. Build the driving base (https://education.lego.com/en-us/product-resources/mindstorms-ev3/downloads/building-instructions).
    c. The micropython documentation (https://pybricks.com/ev3-micropython/ev3devices.html) provides documentation on what commands are available to control the robot.
2. Task 1 - Forward movement
    a. Using python's builtin time.sleep(*duration*) and the run(*speed*) and stop motor commands, write a program that drives the robot forward (any amount) based on the duration of the sleep command and the speed set in run(). Note, the speed parameter in the run() command is in deg/s.

b. Derive an equation that specifies how far the robot moves forward based on the duration of the run command, the speed, and circumference of the wheels. Hint, convert the speed into rotation per minute (rpm).

c. Come up with three sets of duration and speed parameters that, based on your equation, should make the robot move 25 cm.

   i. For each duration and speed pair, run the robot 5 times and record the actual distance traveled.

d. Repeat part c but this time do it on a different floor surface (e.g. tile, wood, carpet).

3. Task 2 – Rotation

   a. Create a new script that causes the robot to rotate in place.

   b. When navigating spaces meant for humans, mobile robots will often need to rotate 90 degrees. For our robots, a 90 degree turn should involve one wheel doing a full rotation in one direction and the other wheel doing a full rotation in the opposite rotation.

   c. Using your most accurate duration and speed parameters from the prior Task, program the robot to turn 90 degree clockwise by having the wheels do full rotations in opposite directions. Measure the error (distance off) in the rotation for 5 trials.

      i. Hint, to measure the error in the rotation you can use a setup like the one shown in Figure 1. Note, that the center of rotation is between the robot's axles so you will need to line that up with the cross drawn on the paper.

   d. Do part C again but this time have the robot rotate counter-clockwise.

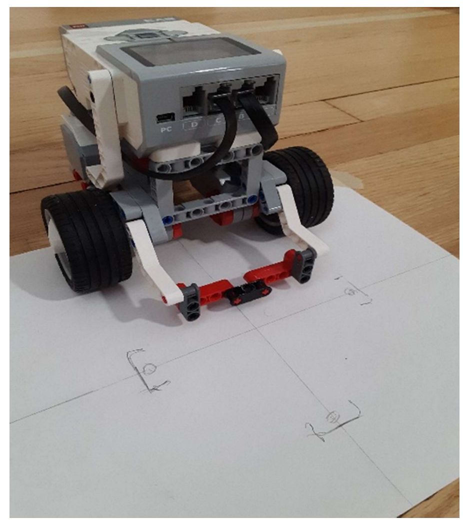

Figure 1 Setup for measuring error in rotating 90 degrees. Note, the robot axle should be placed over the center cross.

4. Task 3 – Path traversal

   a. Create the three paths shown in Figure 2.

   b. Program the robot to traverse each of the three paths.

   c. For each path, record the error in the end locations for 5 trials

   d. For one of the paths, record a short video

1. Robots must pass over all the line segments.
2. At the start and end, the robot should face the same direction.
3. Each line segment is 25 cm

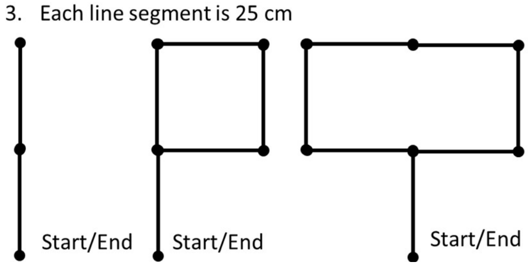

Figure 2: Path layouts for Task 3.

Questions

1. For the forward movement, how accurate where the different speed and duration sets?
2. How would the floor type affect the distance the robot moved?
3. For the forward movement, what are some factors, aside from floor type, that could cause differences between the predicted and the actual distance?

4. For the rotation movement, was rotating clockwise the same as rotating counterclockwise?
5. Which of the three paths had the smallest end location error? Why?

Write up notes/reminders

1. Methods
    a. Any equations you derived.
    b. Brief description of task and measuring setups. Include pictures.
2. Results
    a. Data and explanation for each task.
        i. Includes plots of the error measurements with error bars.
    b. Include a screenshot and discussion of the video you took for Task 3. Submit the video via Canvas.
3. Discussion
    a. See questions
4. Supplementary Material
    a. Pdfs of your code

Rubric – Lab 1

General formatting (10)

- Title, date, name, section headings
- Miscellaneous style and formatting

Abstract (15)

- Gives a brief intro or overview of the Lab and/or topics covered.
- Gives a brief summary of the Results and main points from the Discussion.
- Well written and concise without typos or grammar issues.

Methods (10)

- Possess all relevant equations and algorithm explanations
- Looks nice and is easy to follow.

Results (40)

- Figures and text for all experiments.
- Figures are properly formatted (labels, legends, captions, etc.).

Discussion (25)

- All questions are answered.
- Well written and concise without typos or grammar issues.