David Schulz & Andrew Ruswick

Lab 1: Open Loop Control

CS-4981-021

9/13/2021

**Lab 1: Open Loop Control 1**

**Abstract**

In this lab, we gain familiarity with programming the EV3 robot on open-loop tasks by learning the basics of making it go forwards and backwards and rotate for specified distances or degrees. The code only takes rotations per second and time to wait as inputs, so to do this, we have to derive equations that find distance and turn degrees, respectively, from a given motor speed, wait time, and robot wheel circumference and track width. After finding the correct equations and combinations of inputs that resulted in a 25 cm straight run and 90 degree turn, we tested the combinations by running them on the robot and comparing the actual movement measurements to the expected values. From our measurements, we are able to draw two conclusions. The first is that a smoother surface allows the robot to go farther due to less friction, taking more time to come to a full stop. The second is that higher motor speeds take more time to come to a full stop without proper control, making the robot go farther than intended.

**Methods**

Wheel circumference: 5.6cm \* π = 5.6cm \* 3.14 = 17.6cm

Track width: 11.7 cm

Distance = (Motor speed \* Sleep time) \* (17.6 cm / 360 degrees)

Wheel rotations = (Degrees of turn / 360) \* (Track width \* π / Wheel circumference)

**Results**

**Task 1**

For task 1, we put a strip of tape on both a table in Diercks and the carpet. Then we marked a starting line at one end of the tape, put the robot at the starting line, ran the program we coded that’s supposed to make it go forward for 25 cm, and marked the point it stopped at. We then recorded the distance it travelled and repeated that whole process 4 more times for each surface type.

25 cm = (Motor speed \* Sleep time) \* (17.6 cm / 360 degrees)

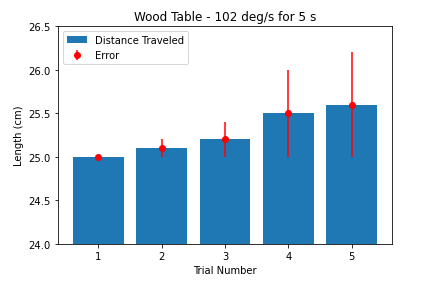
25 cm = (Motor speed \* Sleep time) \* 0.0489 cm/degree

511.364 degrees = Motor speed \* Sleep time

**Wood Table Trials:**

* 102 degrees/s for 5 s

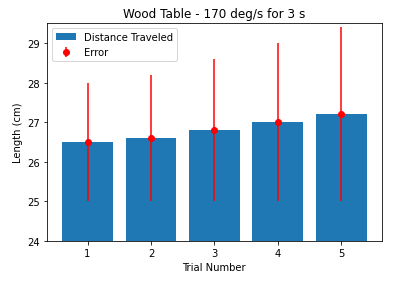
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 25 cm | 25.1 cm | 25.2 cm | 25.5 cm | 25.6 cm |



**Figure 1:** The above graph shows the distance, measured in centimeters, that the robot traveled for each trial. The error corresponds to the distance that the robot was intended to travel (25 cm).

* 170 degrees/s for 3 s

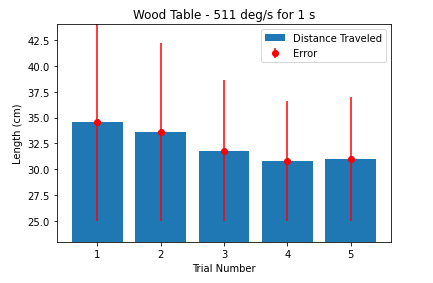
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 26.5 cm | 26.6 cm | 26.8 cm | 27 cm | 27.2 cm |



**Figure 2:** The above graph shows the distance, measured in centimeters, that the robot traveled for each trial. The error corresponds to the distance that the robot was intended to travel (25cm).

* 511 degrees/s for 1 s

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 34.6 cm | 33.6 cm | 31.8 cm | 30.8 cm | 31 cm |

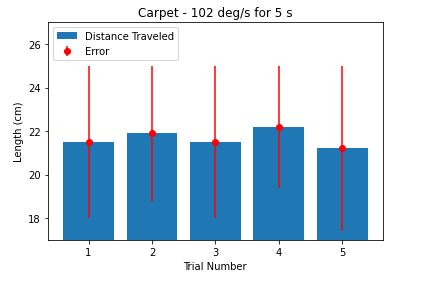


**Figure 3:** The above graph shows the distance, measured in centimeters, that the robot traveled for each trial. The error corresponds to the distance that the robot was intended to travel (25 cm).

**Carpet Trials:**

* 102 degrees/s for 5 s

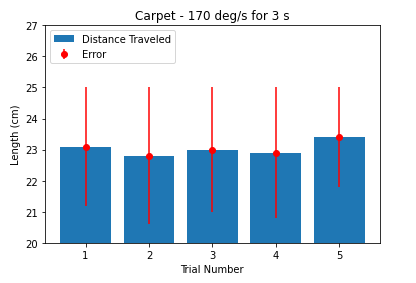
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 21.5 cm | 21.9 cm | 21.5 cm | 22.2 cm | 21.2 cm |



**Figure 4:** The above graph shows the distance, measured in centimeters, that the robot traveled for each trial. The error corresponds to the distance that the robot was intended to travel (25 cm).

* 170 degrees/s for 3 s

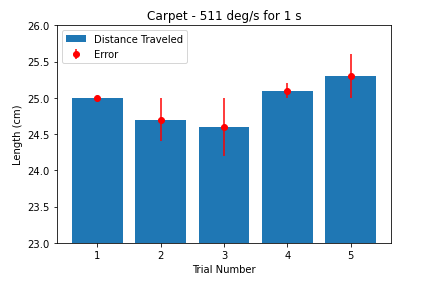
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 23.1 cm | 22.8 cm | 23.0 cm | 22.9 cm | 23.4 cm |



**Figure 5:** The above graph shows the distance, measured in centimeters, that the robot traveled for each trial. The error corresponds to the distance that the robot was intended to travel (25 cm).

* 511 degrees/s for 1 s

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 25.0 cm | 24.7 cm | 24.6 cm | 25.1 cm | 25.3 cm |



**Figure 6:** The above graph shows the distance, measured in centimeters, that the robot traveled for each trial. The error corresponds to the distance that the robot was intended to travel (25 cm).

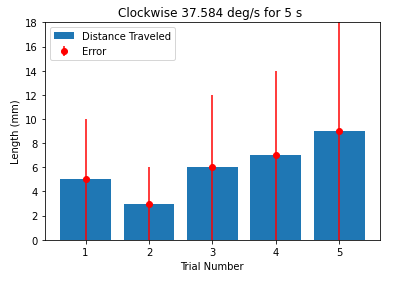
**Task 2**

For task 2, we used the same method of measuring error as Figure 1 shows in the lab directions. In the code, we chose the turning time that produced the most accurate results from task 1. We centered the robot on the + on the paper and then ran the code to turn the robot, turning clockwise and counter-clockwise 5 times each. After each run, we marked the point in the hole of the front robot measuring assembly and then recorded the distance from the point to the center of the original mark for that side.

Wheel rotations for 90 degrees of turn = (90/360) \* (11.7π cm / 17.6 cm) = 0.522 rotations (187.92 degrees)

* Clockwise 37.584 degrees/s for 5 s

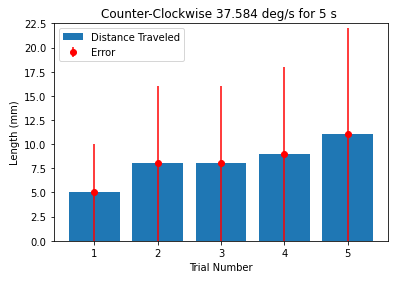
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 5 mm | 3 mm | 6 mm | 7 mm | 9 mm |



**Figure 7:** The above graph shows the distance, measured in millimeters, that the robot deviated from the expected final position for each clockwise trial, also called the error.

* Counter-clockwise 37.584 degrees/s for 5 s

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 5 mm | 8 mm | 8 mm | 9 mm | 11 mm |



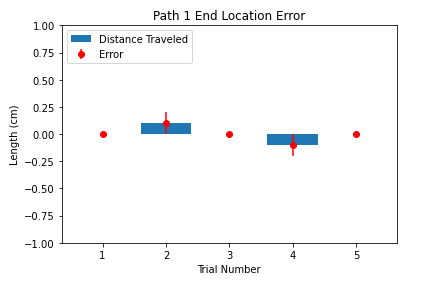
**Figure 8:** The above graph shows the distance, measured in millimeters, that the robot deviated from the expected final position for each counter-clockwise trial, also called the error.

**Task 3**

For task 3, we used masking tape to create three paths for the robot to follow. We also used a piece of tape to mark the starting location to make it easier to line up the robot in the same spot for each trial. Using EV3’s built-in *DriveBase* class, we utilized it’s *straight()* and *turn()* methods in order to successfully traverse each of the three paths. Once the robot returned and stopped, we measured the distance between the starting location and ending location (in centimeters) in order to calculate the error. We did this five times for all three of the paths.

* Path 1 End Location Error

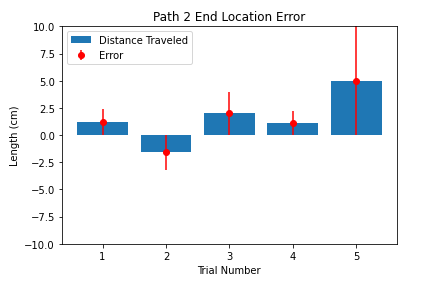
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0.0 cm | 0.1 cm | 0.0 cm | -0.1 cm | 0.0 cm |



**Figure 9:** The distance, measured in cm, of the ending location to the starting location for Path 1. The positive values indicate the robot stopped in front of the starting location and the negative values indicate the robot went past the starting location.

* Path 2 End Location Error

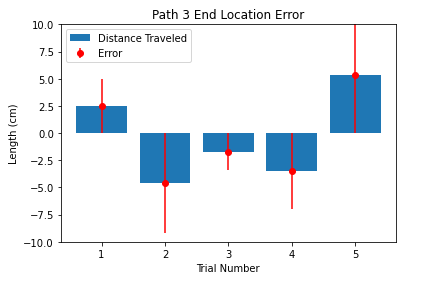
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| +1.2 cm | -1.6 cm | +2.0 cm | +1.1 cm | +5.0 cm |



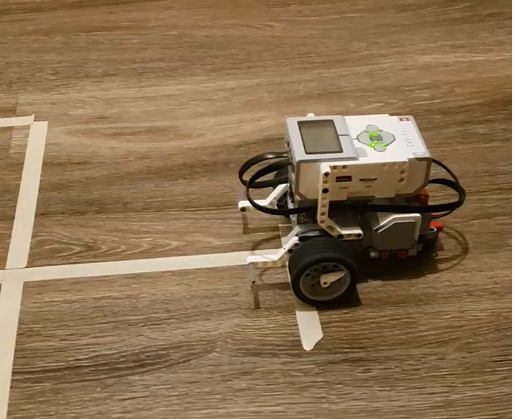
**Figure 10:**The distance, measured in cm, of the ending location to the starting location for Path 2. The positive values indicate the robot stopped in front of the starting location and the negative values indicate the robot went past the starting location.

* Path 3 End Location Error

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| +2.5 cm | -4.6cm | -1.7 cm | -3.5 cm | +5.3 cm |



**Figure 11:** The above graph shows the distance, measured in cm, of the ending location to the starting location for Path 3. The positive values indicate the robot stopped in front of the starting location and the negative values indicate the robot went past the starting location.



**Figure 12:** A screenshot from the video of Path 1.

* Video Discussion

We decided to take a video of our robot traversing Path 1 because of how surprised we were when it was able to consistently produce little-to-no error. We learned that *DriveBase*’s *straight()* method is extremely precise, especially when compared to *Motor*’s *run()* method. The video can be found within the submitted .zip file.

**Discussion**

1. For the forward movement, how accurate were the different speed and duration sets?

* Surprisingly, they were pretty inaccurate. By comparing Figures 1-3, you can see that the inaccuracy was directly proportional to the increase in speed for the wood table trials. And by comparing Figures 4-6, you can see that the inaccuracy was directly proportional to the decrease in speed for the carpet trials.

1. How would the floor type affect the distance the robot moved?

* When we switched the floor type from wood to carpet, the force of friction decreased. This caused a loss of traction which resulted in the robot not being able to travel as much distance. This is especially apparent when comparing Figure 3 to Figure 6. You can see that even at the highest speed, the robot was just able to reach the 25 cm target point when on carpet. But when using that same speed on wood, the robot overshot the target point in each of the five trials.

1. For the forward movement, what are some factors, aside from floor type, that could cause differences between the predicted and the actual distance?

* Some factors that could cause a difference between the predicted and actual distance include, the starting point for each trial not being equal, the motors starting and stopping at different times, and the accuracy of *time.sleep()* for each trial.

1. For the rotation movement, was rotating clockwise the same as rotating counterclockwise?

* No. The ending position of the robot was consistently too far to the right while going both directions, meaning it overshot one direction and undershot the other.

1. Which of the three paths had the smallest end location error? Why?

* Path 1 had the smallest end location error because it was the least complex path out of the three due to not having any turns. The turns added a layer of complexity that introduced more end location error.