Group 53 - Navigation
Alessandro Commodari 260636932
Asher Wright 260559393

Lab 3: "Navigation" Report

Data

Below is the data collected from the initial path (part 1). Note that all measurements are relative to the end point of (0,60). So if the odometer X = -0.26cm, it means the robot was at 59.74cm.

Odometer X (cm)	Odometer Y (cm)	Measured X (cm)	Measured Y (cm)	Error X (cm)	Error Y (cm)
-0.26	-0.09	-0.20	-0.40	-0.31	0.06
-0.01	0.00	-0.80	-0.10	-0.10	-0.79
0.03	0.03	-0.75	0.50	0.47	-0.78
0.02	-0.01	-0.15	0.25	0.26	-0.17
-0.32	-0.14	1.20	0.60	0.74	1.52
0.17	0.10	0.60	1.10	1.00	0.43
-0.31	-0.09	-1.05	-0.30	-0.21	-0.74
0.01	-0.02	-1.40	0.30	0.32	-1.41
-0.39	-0.15	-0.15	-0.20	-0.05	0.24
-0.31	-0.13	-1.10	-0.35	-0.22	-0.79

STDEV X (cm)	STDEV Y (cm)	Mean X (cm)	Mean Y (cm)
0.44	0.84	0.19	-0.24

Data Analysis

The formula for standard deviation is (taken from Wikipedia):

$$s = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{x})^2}.$$

To find the mean, it is only necessary to sum all elements, and then divide by the number of elements.

The formula is as follows (taken from Wikipedia):

$$A = \frac{1}{n}\sum_{i=1}^{n} a_i$$

Calculating Standard Deviation:

For X Error:

First, calculate the mean:

$A_x = \frac{1}{10}(-0.31 - 0.1 + 0.47 + 0.26 + 0.74 + 1.0 - 0.21 + 0.32 - 0.05 - 0.22)$

$A_x = \frac{1}{10}(1.9) = \mathbf{0.19\ cm}$

Then calculate the standard deviation:

Note that x-bar in the standard deviation formula is the mean, i.e. A_x.

Thus:

$$S_x = \sqrt{\frac{1}{10-1}\sum_1^{10}(x_i - A_x)^2} = \sqrt{\frac{1}{9}[(-0.31-0.19)^2 + (-0.1-0.19)^2 + (0.47-0.19)^2 + (0.26-0.19)^2 + (0.74-0.19)^2 + (1.0-0.19)^2 + (-0.21-0.19)^2 + (0.32-0.19)^2 + (-0.05-0.19)^2 + (-0.22-0.19)^2]}$$

$S_x = \sqrt{\frac{1}{9}(1.7786)} = \sqrt{0.1976}$

S_x = 0.44 cm

For Y Error:

First, calculate the mean:

$A_y = \frac{1}{10}(0.06 - 0.79 - 0.78 - 0.17 + 1.52 + 0.43 - 0.74 - 1.41 + 0.24 - 0.79)$

$A_y = \frac{1}{10}(-2.43) = \mathbf{-0.24\ cm}$

Then calculate the standard deviation again:

$$S_y = = \sqrt{\frac{1}{10-1}\sum_1^{10}(y_i - A_y)^2} = \sqrt{\frac{1}{9}[(0.06+0.24)^2 + (-0.79+0.24)^2 + (-0.78+0.24)^2 + (-0.17+0.24)^2 + (1.52+0.24)^2 + (0.43+0.24)^2 + (-0.74+0.24)^2 + (-1.41+0.24)^2 + (0.24+0.24)^2 + (-0.79+0.24)^2]}$$

$S_y = \sqrt{\frac{1}{9}\ (6.3873)} = \sqrt{0.7097}$

S_y = 0.84 cm

The error is mainly the result of the odometer. The navigator is not travelling perfectly: The measured x and y should be zero, but they actually have an average of 0.19cm and -0.24 cm, and a standard deviation of 0.44cm and 0.84cm respectively. If the navigator was working perfectly, then the result would always be zero. However, the error is the result of the odometer, since, regardless of how the navigator controls the robot, the odometer **should** read the correct measurement. This would mean that, although the robot may not end up in the correct position (due to the navigator), it would still display the correct position (which it does not). This is assuming that "the errors" to which the question refers, are the errors in the chart above. Of course, it can be argued that the robot not traveling to the correct position is an error (navigator). This is not shown in the chart, which just shows error between where the robot was, and where it thought it was.

The odometer error is mainly a result of the parameters for the robot, such as wheel radius and track width. It is also a result of slipping (although this was minimized by setting the motor acceleration very low). The robot "thought" it was turning and moving perfectly, and thus usually ended with a very close number to the result.

Also, since the odometer "controls" the navigation, any error in odometry is an error in navigation.

Observations and Conclusions

Q: In three to four sentences, explain the operation of your controller(s) for navigation.
Non-avoiding controller:

This controller gets a list of points to which the robot should travel, finds the correct angle to turn to face towards the next point, and then travels the hypotenuse ($\sqrt{x^2 + y^2}$) to get to this point. It repeats this for each point in the list.

(2 sentences)

Avoiding-controller:

This controller gets a list of points to which the robot should travel, turns the correct angle to face towards the next point, and starts to travel straight at a constant speed unless it detects an object with its ultrasonic sensor. If it detects an object, it marks the object's position, moves past it, turns back onto the route, and continues until it reads the desired position (and repeats for all points).

(2 sentences)

Q: How accurately does it move the robot to its destination?

Both controllers move the robot very accurately. The first is simply rotating and straight movement, and since the robot's parameters are well defined, and the acceleration is small, there are not many sources of error. The second actually waits for the odometer to read the correct value, and so, it is very accurate too.

Q: How quickly does it settle (stop oscillating) on its destination?

It settles very quickly. With the first controller, it immediately settles upon reaching the position (very little rotation after reaching). With the second controller, it takes a bit for the oscillation to finish, as it may be coming to the point from a different position (not straight from the last point.

Q: How would increasing the speed of the robot affect the accuracy of your navigation?

Increasing the speed of the robot would decrease the accuracy of the robot's navigation. For one, it would result in more potential slipping, which results in worse odometry readings. Second, for the avoiding controller, it would result in less accurate movements. Since the robot is checking its odometer to see if it reaches the correct point, the faster the robot moves over this point, the more the robot could be off (since there is a set amount of ultrasonic polls per second).

Q: What is the main source of error in navigation (and odometry)?

The main source of error is incorrect robot parameters, i.e. the track width and wheel radii. Slipping, in our case, is non-existent, since the acceleration of the robot is so low. Instead, having an incorrect track and/or wheel radii results in turns that are not exactly the desired theta, and distances that are not desired as well.

Since our navigation is dependent on our odometry, these odometer problems affect the navigation as well.

Further improvements

A hardware improvement that could be implemented to help correct errors with navigation and odometry would be **to add a second ultrasonic sensor.** With one directly in front of the robot that could pick up objects in front, and a second on the side of the robot at 90 degrees that could track the wall after turning to "follow" it, the robot could better navigate around objects.

Our robot could adapt to different obstacles (unlike with some hardcoded software), but this is currently at the cost of performance. The robot took turns too close to the wall, since it would constantly analyze whether it should be a wall-follower or point-driver. A software improvement would be to, upon seeing a block, **switch completely over to wall-follower (a separate thread), and, upon passing the wall, switch back to the point-driver** (rather than constantly check).

A final hardware & software improvement would be to **add an initial option for calibrating the robot parameters using a light sensor.** The robot would be placed on a gridline intersection, and the robot would spin from one black line to the other, and adjust the track to ensure that it recognized it as a 90-degree turn.

(6 sentences)