ECSE 211 Lab 2

Odometry Lab Report

Group 65

Kathy Yim 260741794

Hanwen Wang 260778557

Design Evaluation

The robot in this lab is composed of three main components: two motors and a light sensor. The light sensor is placed between the two motors at approximately the point of rotation of the robot in the middle. Due to the fact that ambient light can cause errors to line detection, we placed the sensor fairly close to the ground to minimize the possible errors. In addition, the two motors are placed on the sides of the EV3 brick rather than directly underneath to increase its stability when turning corners. We also made use of the metal ball fixed at the back of the EV3 for increased stability and mobility.

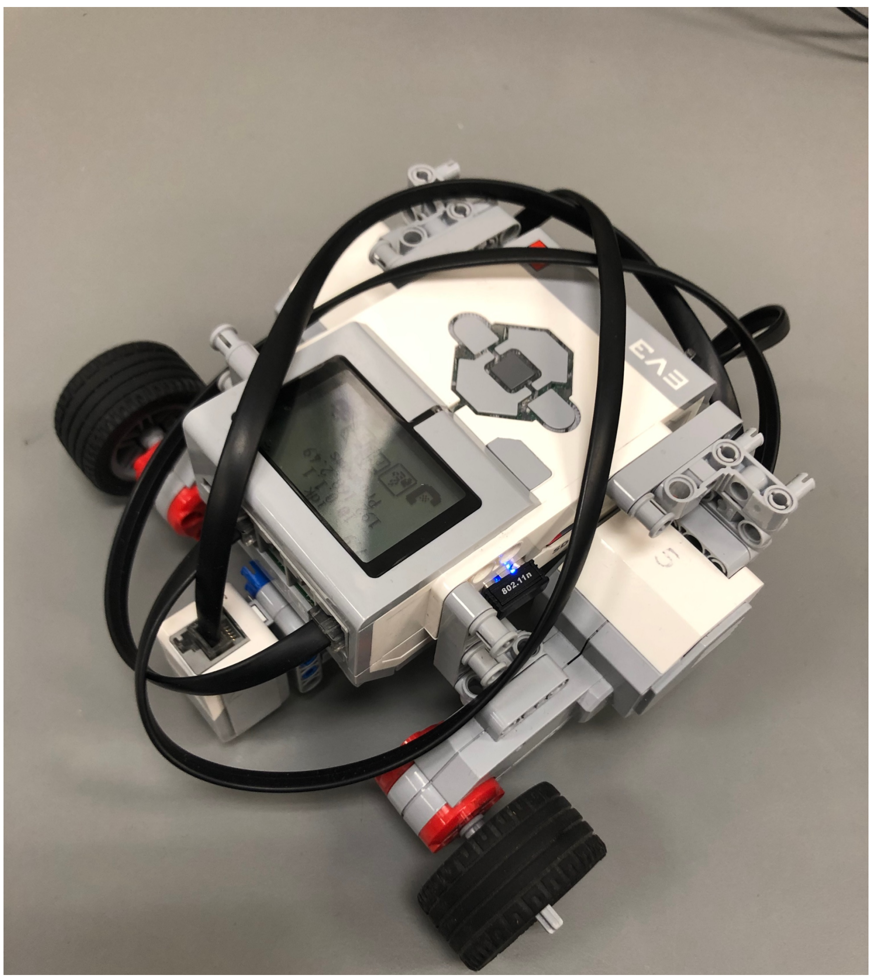

Figure 1: Hardware design overview

For the software design, the predefined constants in the given code had to be tweaked often. Increasing the value of track will increase the turning angle of the robot, while decreasing the track caused the robot to turn at a smaller angle. We ended up with a track value of 17.1cm and a wheel radius of 2.2cm, which allows the robot to move in a square path.

In the OdometerCorrection class, the working process is illustrated as the following:

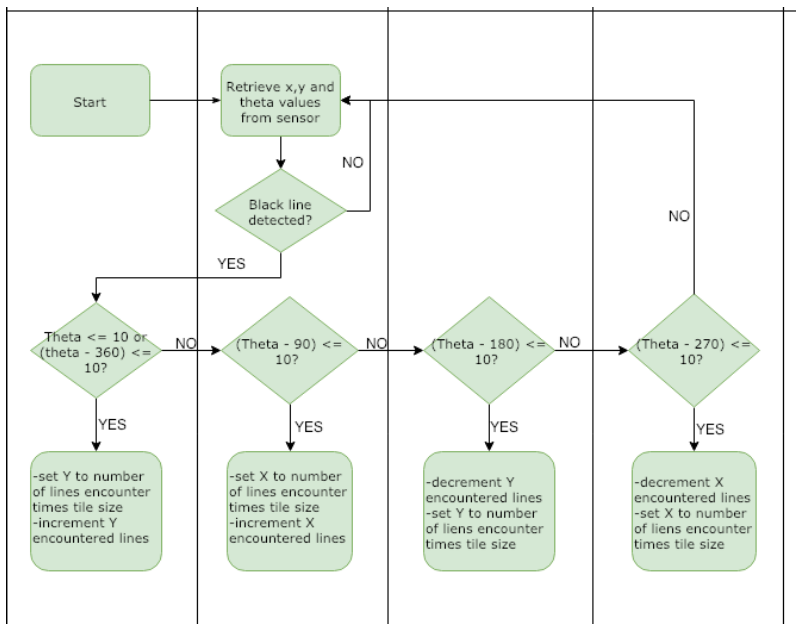

Figure 2: Odometer Correction Workflow Overview

As shown in figure 2, through this implementation, we can reduce the error by setting the value of x and y to the length of the tiles when we move that much. Because the robot may deviate a bit from its original path, we need to ignore that distance and consider the robot only as it moving in a straight line.

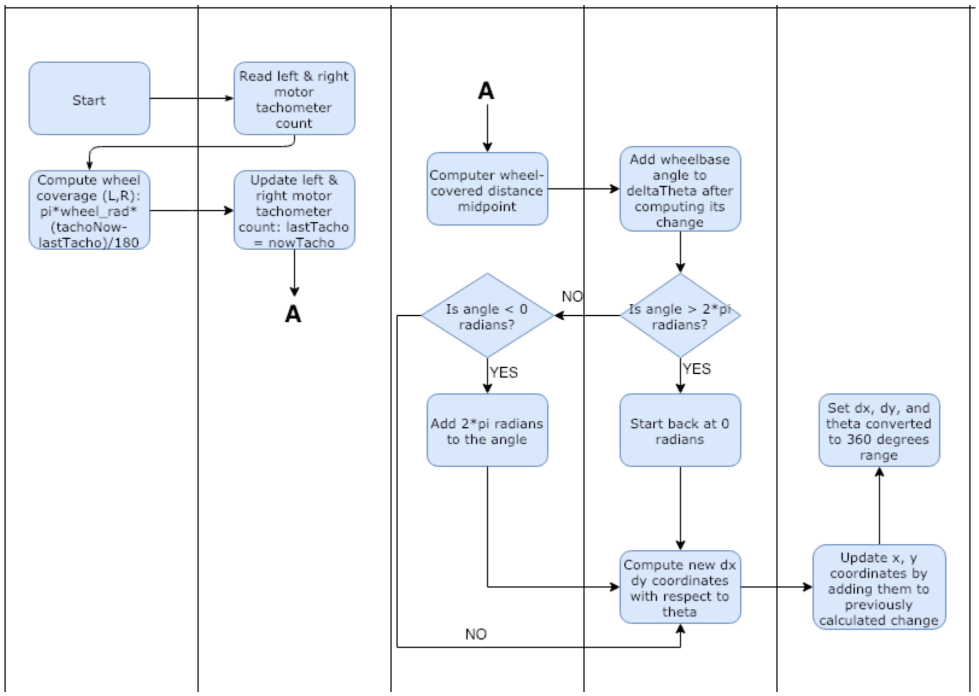

Figure 3: Odometer Workflow Overview

Test Data

Trial	X (cm)	Y (cm)	X_F (cm)	Y_F (cm)	X (cm) (shown on EV3)	Y (cm) (shown on EV3)	Error (cm)
1	-12.20	-6.90	-7.90	-7.50	0.75	-1.08	4.14
2	-14.10	-9.00	-9.50	-13.50	0.58	-0.99	6.44
3	-14.20	-8.80	-10.2	-8.60	0.62	-0.66	4.00
4	-13.60	-8.10	-9.80	-11.70	0.46	-0.85	5.23
5	-13.30	-9.20	-8.30	-11.60	0.66	-0.65	5.55
6	-13.80	-10.50	-8.00	-12.80	0.83	-0.62	2.51
7	-13.20	-10.10	-8.10	-12.10	0.75	-0.63	5.48
8	-14.00	-9.50	-11.20	-11.20	0.40	-0.78	3.28
9	-11.70	-10.40	-10.10	-11.50	0.55	-0.66	1.94
10	-12.00	-9.80	-11.00	-10.90	0.59	-1.10	1.49

Table 1: Odometer Without Correction

Trial	X (cm)	Y (cm)	X_F (cm)	Y_F (cm)	X (cm) (shown on EV3)	Y (cm) (shown on EV3)	Error (cm)
1	-13.60	-9.00	-9.20	-10.30	-10.76	-12.84	2.98
2	-14.70	-10.40	-8.10	-12.10	-9.70	-11.79	1.63
3	-15.30	-9.80	-12.00	-9.50	-13.79	-9.79	1.81
4	-13.90	-10.00	-10.60	-15.50	-11.76	-9.42	6.19
5	-14.00	-9.90	-12.30	-14.50	-13.88	-14.77	1.60
6	-14.20	-10.20	-11.80	-12.50	-13.05	-12.34	1.26
7	-13.20	-9.60	-10.10	-11.70	-11.28	-12.07	1.24
8	-14.10	-11.80	-12.90	-13.20	-14.18	-13.89	1.45
9	-15.20	-12.30	-12.80	-14.50	-13.33	-13.98	0.74
10	-12.70	-9.70	-9.20	-11.10	-10.11	-11.71	1.10

Table 2: Odometer With Correction

Test Analysis

	Without Correction	With Correction
X_{mean}	-13.21	-12.18
$X_{standard\ deviation}$	0.87	1.58
Y_{mean}	-9.23	-12.26
$Y_{standard\ deviation}$	1.05	1.64
$Error_{mean}$	4.00	2.06
$Error_{standard\ deviation}$	1.59	1.51

Table 3: Mean and SD computations

As show in table 3 above, the mean and the standard deviation of the error both decreased after correction was implemented. Without correction, the error on the x and y-values is dependent on the accuracy of wheel radius and track values of the square the robot is moving in. If the robot moved a perfect square path, the odometer and the robot's actual position would be (0,0). However, with correction implemented, the error of X and Y was no longer dependent solely on the accuracy of the track. Moreover, outlier values due to external factors such as slipping of the wheels are ignored. The robot's position was corrected each time it passes a line, giving more accurate readings on the odometer.

The error in the x direction is expected to be smaller with because the last correction of the robot's path is entirely on the x-value. When the robot crosses the third line after turning its last corner, only the x-value is corrected. Therefore, it is expected of the robot to have a move accurate value in the x direction compared to the y direction.

In order to prove that our assumption is correct, we calculated the error in the x and y-values in the table below.

Trial	**$\lvert X-X_F \rvert$ (cm)**	**$\lvert Y-Y_F \rvert$ (cm)**
1	2.84	3.84
2	5.00	1.39
3	1.51	0.01
4	2.14	0.58
5	0.12	4.87
6	1.15	2.14
7	1.92	2.47
8	0.08	2.09
9	1.87	1.68
10	2.59	2.01
	$\lvert X-X_F \rvert$ (cm)	**$\lvert Y-Y_F \rvert$ (cm)**
Mean	1.922	2.108

Table 4: Error of mean calculation

As shown in table 4 above, the x-value error is smaller than that of the y-value, which testifies to our initial argument.

Observations and Conclusions

The error observed on the odometer without correction is not tolerable for large distances. This lab focuses specifically on a 3-by-3 tile trajectory, and the track and wheel radius values were tweaked accordingly. Each time the robot makes a turn around a corner, it makes a small deviation from its original path. If the robot were to travel five times the 3-by-3 grid's distance, the error will accumulate on every 90-degree turn; therefore increasing the overall error by a significant amount.

We expect the error to grow linearly with respect to the distance travelled. Whenever the robot turns around a corner, a small error in the angle of rotation occurs, which re-orients the robot to a slightly different path. As the distance travelled increases, the robot is expected to deviate further and further from its original path. Therefore, the odometer's error would increase somewhat linearly with the distance travelled.

Further Improvements

To reduce the slip of the robot's wheels, we can decrease the speed and acceleration of the robot so it moves at a steadier pace. At lower speeds, the robot's wheels can attain a better grip to the ground. However, decreasing the speed would also mean an increase in static friction between the wooden floor and rubber wheels. An algorithm can be implemented such that the robot is able to recognize when it is moving at a steady pace versus when the wheels are slipping. To do this, we can create cases where some angle theta is within a small range of values. Depending on the value of theta, the algorithm is able to calculate a very small range of values x and y corresponding to each theta. If x or y falls outside of this range, the robot's wheels are slipping and the algorithm begins re-evaluating the robot's position to rectify its trajectory.

To correct the angle reported by the odometer using two light sensors, we would position the first sensor the same way we did for our experiment, so it can continue to detect passing lines. The second sensor would be placed on the right of the robot at 90-degrees. The second light sensor can detect lines running parallel to the robot's right hand side while keeping it aligned.

In the case where the robot has only one light sensor, we can calculate the distance between the lines the robot passed by using the x and y values shown on the odometer. Since the predefined distance is set to 'TILE_SIZE' in our code, we can compare the distance we calculated from the reported x and y values to 'TILE_SIZE'. This allows us to estimate how far the robot deviated from its original path. From the values of x and y we can then calculate the deviation of the robot from its original path and therefore obtain a more accurate angle than the one reported by the odometer.