# ECSE 211
# Lab 2: Odometer
# Lab Report

Group 20

Linwei Yuan (260857954)

Le-Li Mao (260800098)

## Section 1: Design Evaluation

Brief Description

For the hardware design, we kept the main structure the same as Lab1 since the main function that the robot should be able to implement in this lab was simply going forward and making turns. So, except for the removal of the ultrasonic sensor, the robot still had two front wheels, a back iron ball, and a base to support the EV3 brick. The major attention in hardware was paid to the firmness of robot and speed balance between two wheels as the two factors had impacts on the moving path of the robot. First, we stabilized the structure by adding more conjunctions to the base of our robot to avoid unexpected mechanical flaws while moving. Second, to balance the speed of two wheels, we changed the wheel design. That was also because we do not want the tile to rub with the motor which can result in different rotational speeds of the two front wheels. The solution was using a longer bar to create separation between motors and tiles. Also, in order to have the exact same separation, we added nuts with exact same size in between. In this way, we not only solved the rubbing problem but also made it possible to adjust the distance of two wheels while keeping the symmetry of them.

The software used the tacho count feature within the lejos motor library to analyze the rotation amount of the motor. After we acquire the tacho count, we computed the delta and use the delta on both sides of the motor to calculate the distance travel by taking the average and also calculate the delta angle by taking the difference and dividing by the wheel width shown in Equation 1 below.

$$\Delta\, Distance \;\approx\; Radian(Delta\ tacho\ Count) \times Wheel\ Radius\ \ \text{(cm)}$$

$$Average\ \Delta\ distance\ \approx \frac{Left\ \ \Delta\, Distance + Right\ \Delta\, Distance}{2}\ \text{(cm)}$$

$$\Delta\,\theta\ \approx \frac{Left\ \ \Delta\, Distance - Right\ \Delta\, Distance}{Base\ Width}\ \ \text{(rad)}$$

Equation 1: Calculation of the delta theta and distance

After computing the delta angle, we approximated the average angle throughout the turn of the robot by adding half of the delta angle and the current angle. This was based on a method similar to the central differences in numerical analysis. After we compute the average angle of the turn. We then computed the $\Delta X$ and $\Delta Y$ by taking the sine and cosine respectively and multiplying by the average delta distance. With $\Delta X$, $\Delta Y$, $\Delta\theta$ computed, we then called the update function to calculate the new position and angle.

Workflow

1. Read the description and the lab requirement.

2. Designed the hardware and make sure the robot is balanced when performing the movement.

3. Developed the code of the odometry and make sure all calculation is valid and reflect the actual movements.

4. Tested the robot and modify the hardware and software structure to make the integration of the system more accurate and behave correctly.

5. Fine-tuned the constant until reaching the ideal behavior

6. Made measurement of robot behavior and perform analysis on the data and reflect on it in the report.

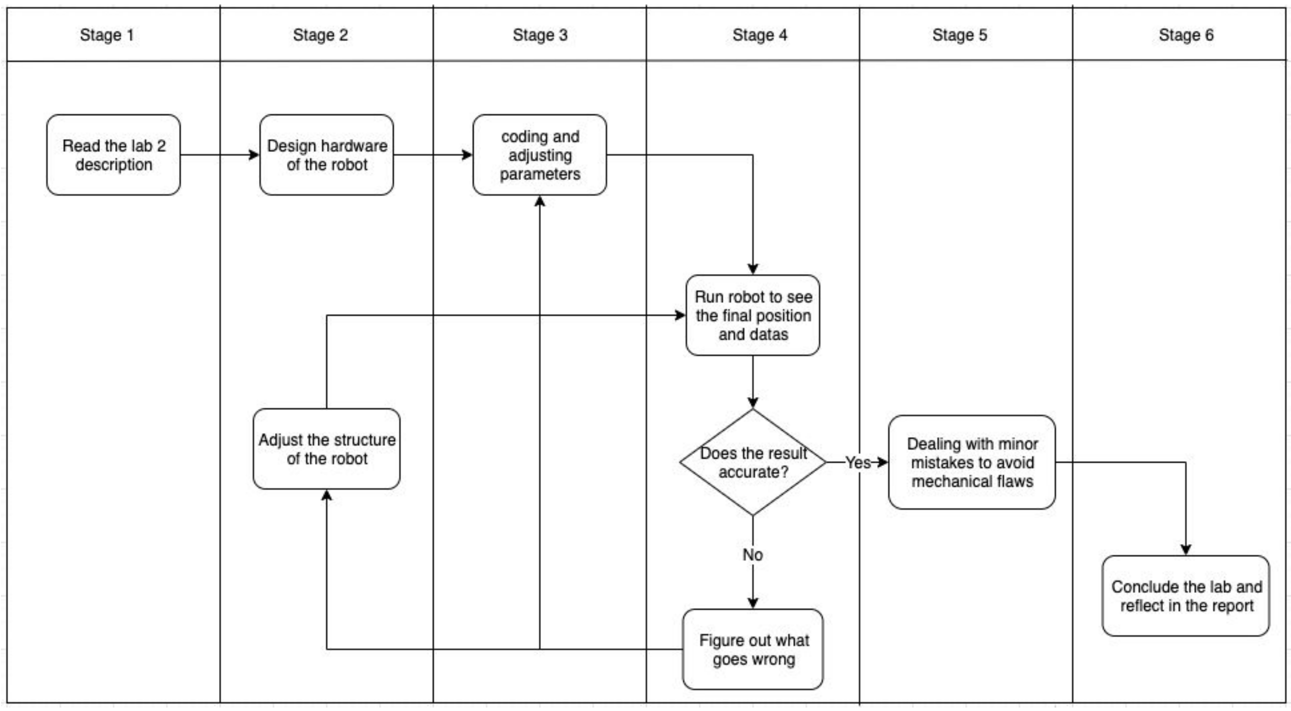


Figure 1: Workflow diagram

<u>Photos of Hardware Design</u>

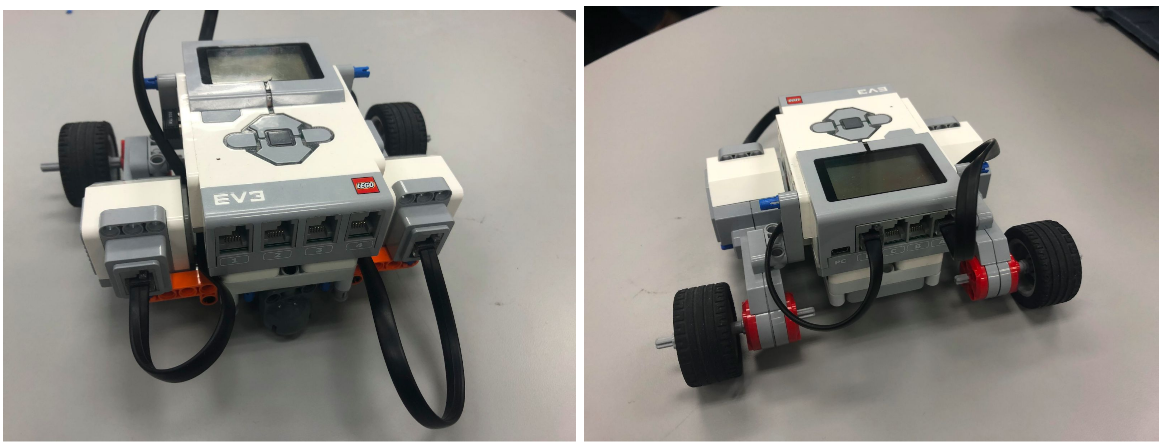

Figure 2: Back view (left) and front view (right) of the robot

Visuals for the Software Design

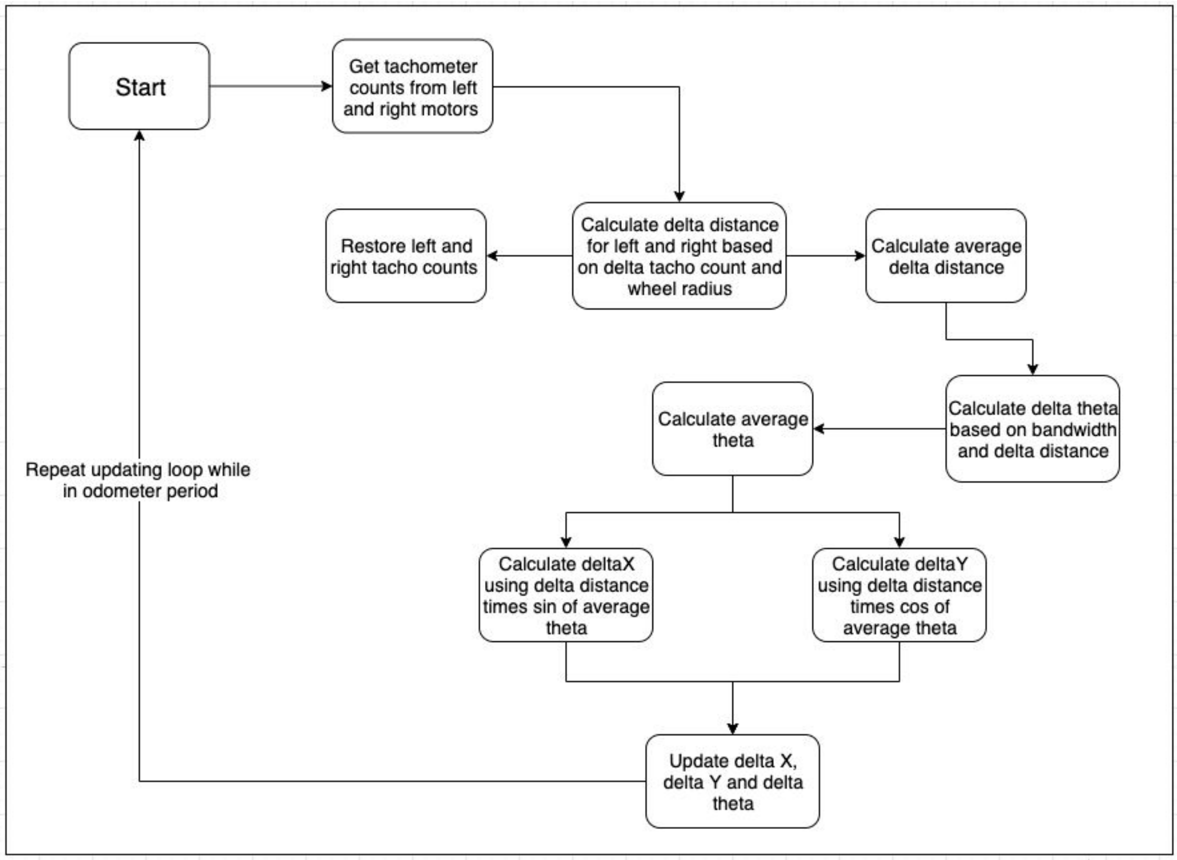


Figure 3: Odometry Calculation Process

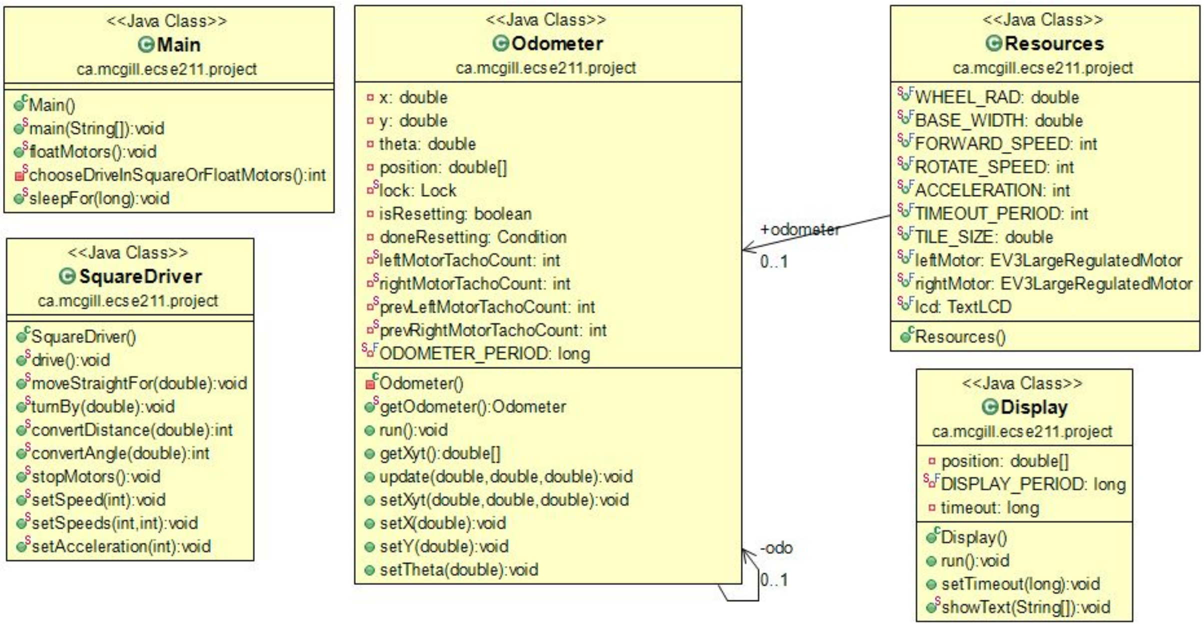


Figure 4: Class diagram of the project

## Section 2: Test data

Odometer test

| Trial Number | Physical Position $X_f$ (cm) | Physical Position $Y_f$ (cm) | Reading X (cm) | Reading Y (cm) |
|---|---|---|---|---|
| 1 | 15.30 | 13.85 | 15.359 | 14.001 |
| 2 | 15.80 | 13.95 | 15.508 | 14.390 |
| 3 | 15.40 | 13.10 | 15.453 | 13.929 |
| 4 | 16.05 | 12.95 | 16.609 | 13.850 |
| 5 | 15.90 | 14.20 | 15.552 | 14.398 |
| 6 | 16.70 | 14.75 | 16.567 | 14.461 |
| 7 | 16.05 | 13.70 | 15.225 | 14.075 |
| 8 | 15.50 | 13.95 | 15.365 | 14.659 |
| 9 | 17.20 | 14.05 | 17.171 | 14.246 |
| 10 | 15.90 | 13.90 | 15.743 | 14.239 |

Table 1: Odometer test data

## Section 3: Test Analysis

Compute the Euclidean error distance ϵ of the position for each test.

Sample Calculation:

$$\sqrt{(x-x_f)^2+(y-y_f)^2} \;=\; \sqrt{(0.059)^2+(0.15)^2} = 0.162\ cm$$

| Trials | error distance ϵ (cm) |
|---|---|
| 1 | 0.162 |
| 2 | 0.528 |
| 3 | 0.830 |
| 4 | 1.059 |
| 5 | 0.400 |
| 6 | 0.318 |
| 7 | 0.906 |
| 8 | 0.721 |
| 9 | 0.198 |
| 10 | 0.373 |

Table 2: Error result of each trial

Compute the mean and standard deviation for X, Y, and ϵ. That means you need to perform 3 mean and 3 standard deviation calculations in total. Use the sample standard deviation formula. Show one sample calculation for both mean and standard deviation formulas.

Sample Calculation:

$$\mathbf{X}_{\text{mean}} = \frac{15.359+15.508+15.453+16.609+15.552+16.567+15.225+15.365+17.171+15.743}{10}$$

$= 15.855$ cm

**X standard deviation :**

$$\sum_{i=1}^{10}(X_{mean} - X_i)^2$$

$$= (15.855 - 15.508)^2 + (15.855 - 15.453)^2 + (15.855 - 16.609)^2 + (15.855 - 15.552)^2 + (15.855 - 16.567)^2 + (15.855 - 15.225)^2 + (15.855 - 15.365)^2 + (15.855 - 17.171)^2 + (15.855 - 15.743)^2$$

$$= 0.408 \ cm^2$$

$$\sqrt{\sum_{i=1}^{10}(X_{mean} - X_i)^2} = \sqrt{0.408}$$

$= 0.639$ cm

| Measurement Type | Mean (cm) | Standard Deviation (cm) |
|---|---|---|
| $X$ | 15.855 | 0.639 |
| $X_f$ | 15.98 | 0.56 |
| $Y$ | 14.225 | 0.246 |
| $Y_f$ | 13.84 | 0.49 |
| $\epsilon$ | 0.550 | 0.296 |

Table 3: Mean and Standard Deviation of each measurement data

What does the standard deviation of X, Y, and ϵ tell you about the accuracy of the odometer? What causes changes in standard deviation?

If the standard deviation was very large, X, Y, and ϵ were not very accurate since their data were located far away from the average value. The standard deviation tells us how consistent were the measurements made by the motor and also how consistent was the environment around the robot. Based on the standard deviation we received the sensor, all physical conditions like the environment, firmness of robot structure, a slight adjustment of components might cause changes in standard deviation. Also, other factors like a more or less optimized algorithm, change of parameters, and doing more trials would affect the data distributions, which could result in changes in standard deviation as well.

What is the sampling frequency of your odometer (i.e. the frequency at which the tacho count is measured)? What is the tradeoff of having a high sampling frequency versus a low sampling frequency?

The frequency of the polling was 40hz which meant that the robot would calculate and update its position every 25ms. If you had high sampling frequency, the system would be under more stress which might result in a motor error but would result in better accuracy. On the other hand, the low frequency would result in less accuracy of position data but would impose less stress on the system overall.

## Section 4: Observations and Conclusions

Is the error you observed in the odometer tolerable for a larger distance? What happens if the robot travels 5 times the 3-by-3 grid's distance?

The error that we observed for our odometer was not tolerable for large distances. When we ran the robot 5 times in the 3-by-3 grid's distance, each time the robot made a 90-degree turn at the corner of the grid, it had a small deviation from its original angle. Therefore, it resulted in a deviation from the original path. This error would accumulate on every turn, and the robot was not going in a perfectly straight line due to the fact that it is not identically symmetric like in an ideal situation. So, the resulting error would be increased by an intolerable amount. The robot was very hard, nearly impossible, to go back to the point where it started.

Do you expect the odometer's error to grow linearly with respect to travel distance? Why?

Based on our observations, we expected the error to grow linearly with respect to travel distance. For each time the robot rotates in 90 degrees, the robot would have a deviation. The deviation in angle would lead to a different pathway for the robot to travel. As the traveling distance increases, the error was added up each time it turns. Therefore, the odometer's error would increase linearly with travel distance.

## Section 5: Further Improvements

Proposing a means of reducing the slip of the robot's wheel using the software.

If we want to reduce the slip of wheels, the software measure that we can take is to decrease the speed or acceleration of the robot. Lower speed and acceleration decrease the chances for the robot to exceed the static friction, so the tiles will have a better grip on the board. Also, reducing acceleration not only promises a better grip for rotating wheels but also for stopping wheels. Due to large inertia, it is possible for the robot to keep moving forward while it is intending to stop, which can be simply resolved by decreasing acceleration or speed.

Proposing a means of improving the accuracy of the odometer using one or more light sensors.

Our way to improve the accuracy of the odometer involves two light sensors, we would position the two light sensors at the front of the robot with one in the top left corner and the other on the top right. For each time the robot passes a line, the two light sensors can detect the black line. If the robot navigates away from the intended path, one of the light sensors will detect the line faster than the other. Then, the algorithm will calculate how much angle the robot has deviated from the original path based on the light sensor input (can be an angle of deviation). The algorithm starts to calculate the angle of reorientation and the robot will react to go back to its trajectory. Therefore, the robot will correct itself as soon as it passes the black line, and it will do this several times before driving in a complete square. So, the accuracy will be improved if the function is implemented effectively.