Brandon Freiberger 260739743
Liang Zhao 260781081
Group 13

ECSE 211: Design Principles and Methods

Lab 4: Localization

1.0 – Design

1.1 – Hardware Design

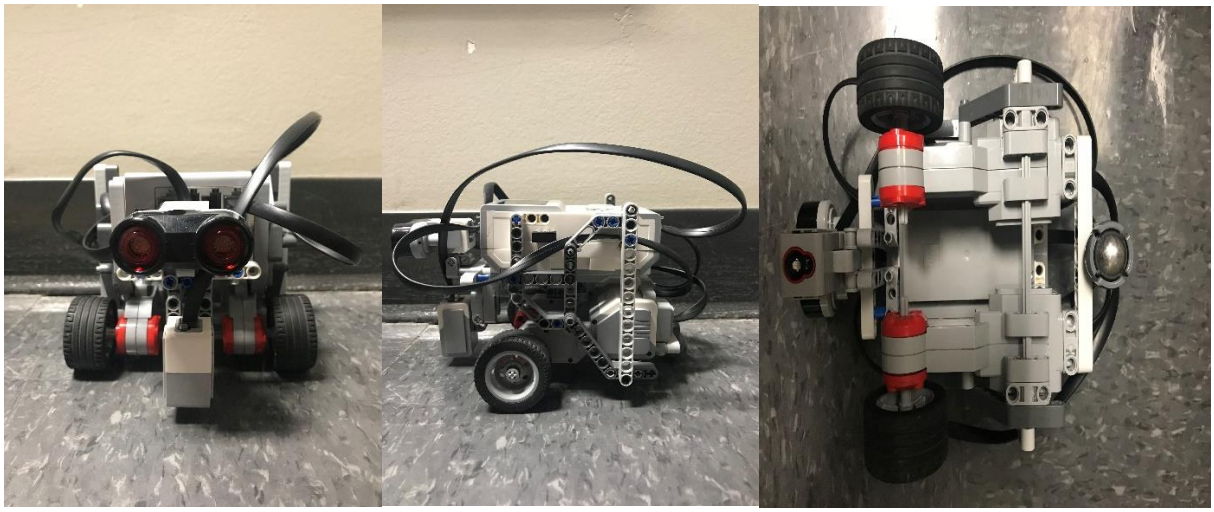

Figure 1.1 – Front, side, and bottom view of our robot design

Our robot used the same chassis as previous labs: we have 2 EV3 motors powering the wheels, with a ball bearing in the back for added rear support as can be seen above in fig. 1.1. Support beams are added on the side of the robot in order to stop it from compressing vertically when it goes over any uneven surfaces when it is traveling, as well as to give us the ability to keep the wires from protruding out from the sides of the robot, thus helping prevent wall collisions.

At the front, we added the US and light sensors. The US sensor was front mounted, as the orientation of the robot was changed instead of the angle of the sensor as was the case in lab 3. The robot did not need to do any sort of obstacle avoidance, so we left it front facing instead of having it at a 45 degree angle as we did in lab 1 and 3. Beneath the US sensor was the light sensor which used the RGB setting, which we had found by experimentation gave more consistent readings of the back lines between the tiles on the grid between both the main white grid as well as the blue grid in the smaller demo room.

Brandon Freiberger 260739743
Liang Zhao 260781081
Group 13

1.1 – Software Design

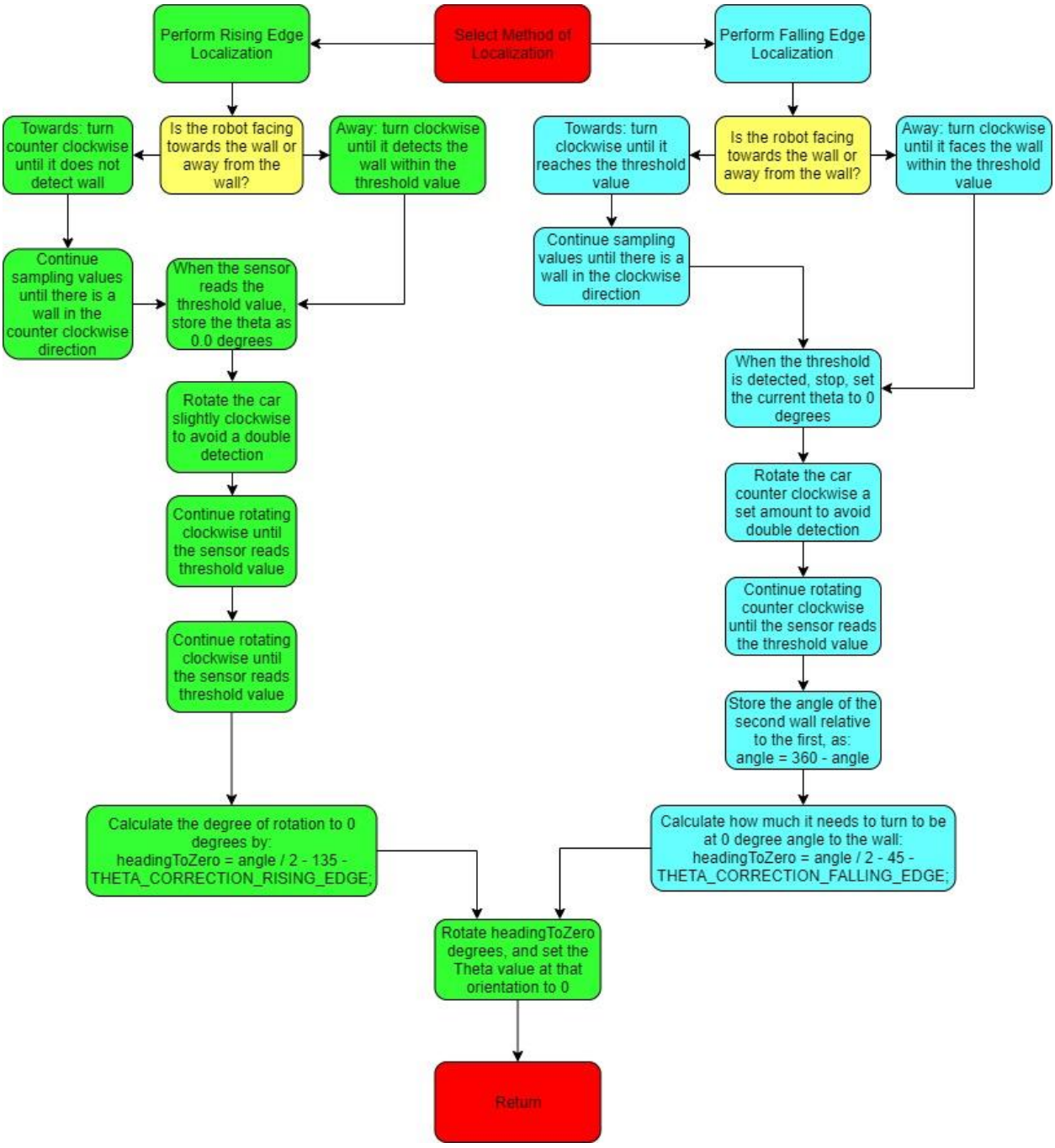

Figure 1.2.1 – The software design of the ultrasonic localization for both the rising edge and falling edge methods. It can be noted that both are quite similar in design, with the main exception being the direction in which they initially rotate.

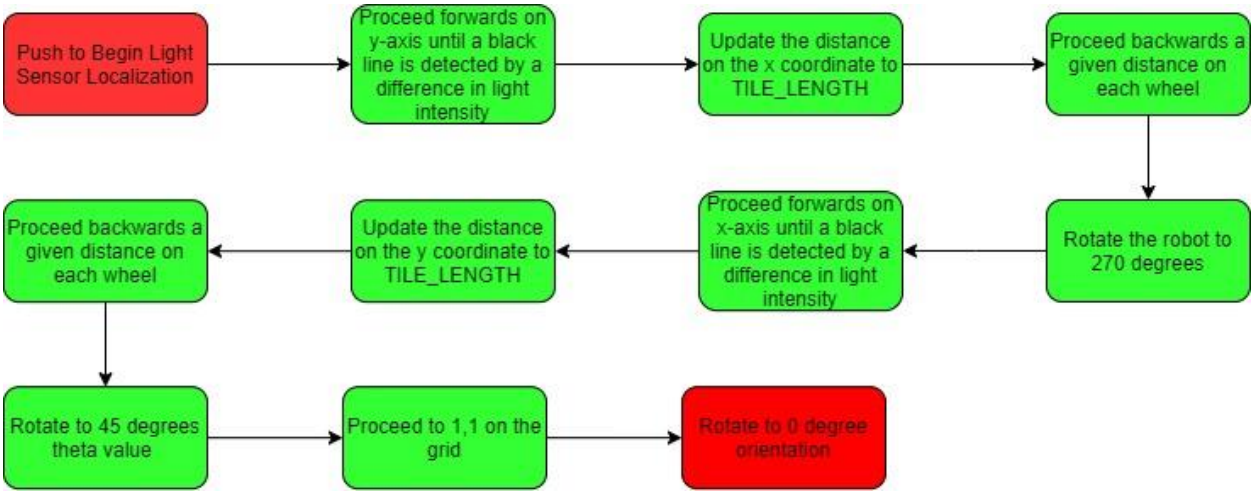

Figure 1.2.2 – The software design for the second part of the localization of the robot, using the light sensor.

The ultrasonic sensor was used for the initial localization, where the robot corrected its orientation and theta value on the grid. The major workaround needed for this stage of the localization was to have the robot ignore the first reading of wall if it was not in the desired starting orientation: if it was the rising edge localization, it would ignore the first value if it started facing towards the wall, and if it was the falling edge, it would ignore the first value if it started facing away from the wall. This was done in addition to having the robot have a mandatory rotation before it began detecting after reading its threshold value. This ensured that the sensor did not double count points and start localizing before it had made the anticipated angle changes to become oriented. It also made sure that the robot was rotating in the anticipated clockwise or counter clockwise way so that it oriented itself at 0 degrees with respect to the y axis when it came to a stop.

The light sensor was used to correct the x and y values based on the black lines of the tiles on the grid. A detailed overview of the steps can be seen in fig. 1.2.2. A summary of the steps is that the robot first moves to the first x line and resets its value to the tile length, and rolls back to where it started. It then turns 90 degrees, and repeats the same functionality for the y value. The robot then moves back to its original location, before proceeding to (1,1) and correcting itself to a 0 degree angle.

As we had learned from previous labs, learning to control the speed and acceleration of the robot is key in making sure that slip is prevented. We reduced our acceleration to less than a third of the value from the previous lab, as well as reduce the forward speed of the robot by three quarters. This gives the ultrasonic sensor and light sensor more time to react, and more time to produce representative values of their current orientation.

The most difficult parts of this lab was finding the correction factors needed to orient the robot in the desired way. For example, after the robot had found the second wall when localizing with the ultrasonic sensor, we needed to add a corrective factor of 7 degrees to make sure that the robot was at 0 degrees relative to the y axis when it stopped rotating and set its theta value to 0 degrees, as can be seen in fig 1.2.1.

2.0 – Test Data

Table 2.1: results of testing the rising edge localization method

Trial number	Orientation (degrees)	Ultrasonic Angle (degrees)	Final angle (degrees)	Final X (cm)	Final y (cm)
1	45	2	3	.5	-.1
2	90	5	5	1.0	-.3
3	135	3	4	.7	-.2
4	180	358	359	-.2	0.0
5	235	5	7	1.1	-.8
6	270	3	4	0.6	-0.2
7	315	4	4	1.0	-0.3
8	0	2	1	.6	-.3
9	45	2	2	0.9	-0.4
10	90	359	0	0.0	-0.1

Table 2.2: Results of testing the rising edge localization method

Trial number	Orientation (degrees)	Ultrasonic Angle (degrees)	Final angle (degrees)	Final X (cm)	Final y (cm)
1	45	355	357	-0.7	0.0
2	90	354	353	-0.9	0.3
3	135	0	0	-0.1	-0.2
4	180	359	0	0	-0.1
5	235	358	0	-0.1	0
6	270	351	353	-0.8	0.5
7	315	350	351	-0.7	0.3
8	0	354	357	-0.4	0.2
9	45	353	355	-0.5	0.3
10	90	357	359	-.2	0.0

3.0 – Test Analysis

Where Euclidean error ε is calculated by:

$$\varepsilon = \sqrt{(X_F)^2 + (Y_F)^2}$$

For Initial Sensor Angle Error:

If *Ultrasonic Sensor Angle* ≥ 0 :

α = *Ultrasonic Angle*

Else: α = 360 – *Ultrasonic Sensor Angle*

For Final Sensor Angle Error:

If *Ultrasonic Sensor Angle* ≥ 0 :

β = *Ultrasonic Angle*

Else: β = 360 – *Ultrasonic Sensor Angle*

Table 3.1.1 – Error analysis of rising edge localization

Trial Number	Euclidian Error Distance ε (cm)	Ultrasonic Angle Error α (degrees)	Final Angle Error β (degrees)
1	.509	2	3
2	1.044	5	5
3	.728	3	4
4	.200	2	1
5	1.360	5	7
6	.632	3	4
7	1.044	4	4
8	.671	2	1
9	.984	2	2
10	.100	1	0

Table 3.1.2 – Error analysis of falling edge localization

Trial Number	Euclidian Error Distance ε (cm)	Ultrasonic Angle Error α (degrees)	Final Angle Error β (degrees)
1	.700	5	3
2	.949	6	7
3	.224	0	0
4	.100	1	0
5	.100	2	0
6	.943	9	7
7	.761	10	9
8	.447	6	3
9	.583	7	5
10	.200	3	1

3.2 – Error Analysis and Standard Deviation

Each of the mean deviations was calculated using the following formula:

$$\text{Mean Average } \bar{x} = \sum x \div n$$

The standard sample deviations were computed using the following formula:

$$\text{Sample Standard Deviation } s_x = \sqrt{(x_i - \bar{x})^2 \div (n-1)}$$

Table 3.2.1 – Average value and standard deviation for the error of the rising edge localization

	Euclidian Error Distance ε (cm)	Ultrasonic Initial Angle Error (degrees)	Ultrasonic Final Angle Error (degrees)
Mean Value	.7272	2.900	3.100
Standard Deviation	.374	1.300	2.022

Table 3.2.2 – Average value and standard deviation for the error of the falling edge localization

	Euclidian Error Distance ε (cm)	Ultrasonic Initial Angle Error (degrees)	Ultrasonic Final Angle Error (degrees)
Mean Value	.501	4.900	3.5
Standard Deviation	.317	3.176	3.170

4.0 – Observations and Conclusions

4.1 – Performance comparison

The best performing localization was the rising edge detection. If the sensor was much more accurate (for instance, the sensor was accurate to with the mm instead of cm), the difference between the 2 different methods would be much smaller. However, as the sensor is prone to large errors, it is unexpected that the rising edge localization routine works better as it involves the robot working in a sub-optimal range for the US sensor for the majority of its measurements. The robot will scan between 10-25cm from the wall in the rising edge detection, and each degree of rotation will cause small changes in readings for the sensor.

The falling edge may have underperformed due to inconsistencies in the “cone” shape where the US sensor gets its measurements from. It is possible that the sensor received a false positive when it should have kept rotating a few more degrees, and began to rotate in the other direction early. On the contrary, as the rising edge localization read smaller values, the cone radius was not as large and allowed for more accurate values to be read by the sensor.

Given these differences, both methods of localization performed well, and had both distance and angle error that was most often in the acceptable range for the scope of this lab.

4.2 – Impact of Initial US Angle

The final angle was not largely impacted by the initial ultrasonic angle. This makes sense, as the first part of the localization lab relies entirely on where the US sensor reads the threshold distance value. The only way that the initial angle of the US sensor would effect the final angle would be introducing more error by having the robot rotate more, as was the case in the falling edge trials when the robot was oriented between 270 and 0 degrees and had to do more turning due to avoidance of double detection.

4.3 – Change of Lighting Conditions

Changing the light conditions did not effect either localization. This made sense, based on the fundamental properties of the 2 sensors:

- The Ultrasonic sensor receives its readings based on how long sound takes to travel from the sensor to an obstacle and back to the sensor. The only way that light would contribute to the readings is if the light created a major rise in temperature, casing the ultrasonic waves to travel faster, giving the sensor a lower distance reading than expected.
- The light sensor has its own radiance, and thus does not rely on background lighting. This sensor could then be effected if there was a very high quantity of ambient light, but it would have to be well above the intensity of the light sensor to have a major effect on the readings of the light sensor. If we were not having the sensor recognize differences in light intensities and instead just measuring to find the intensity of the black light, the ambient lighting would have had a greater effect on the localization as the light sensor would most likely ignore the intensity readings when it goes over a black line.

5.0 – Further Improvements

5.1 – Minimization of Error

It would be better to use a filtering method when using the rising edge method to filter out any extraneous high values, as seen in lab 1 when the robot gets too close to the wall. This filter would be especially useful for the rising edge localizing method, as the majority of the values it reads are less than 20cm, which is out of the optimum range of 20-50cm of the US sensor. This would work best when coupled with an averaging filter, taking the values from the last 3 readings.

5.2 – Another Form of Localization

It would be easiest to find where the corner of the 2 walls are to pinpoint the current orientation of the robot and then have adjust it rotate to align itself with the y axis as it did before. This could work by setting a pre determined minimum distance from the US sensor to the wall, and having the robot continue rotating towards the corner after it finds the minimum distance. After finding the minimum distance, it will slow down until it then finds the corner. Then, the robot can then turn clockwise 135 degrees so that it knows it has lined up with the y axis, and have the odometer set its theta value to 0.

5.3 – Using Localization To Fix Errors in Larger Environments

Localization using the light sensor will only work in larger environments if we can assume that the robot will have access to a grid system like the labs, where there is a marking for every given dimension of the grid. If we make the assumption that the robot will be operating on a marked grid, odometry correction will work only if we know what tile of the grid that the robot is on. If we know that it is in a given tile, it could do this by using its current angle, finding the next line in the positive y direction, then reversing to find the next line in the negative y direction, and centering itself on the y axis to the middle of the tile. It can then do the same with the x axis, and move back to the center of the tile.