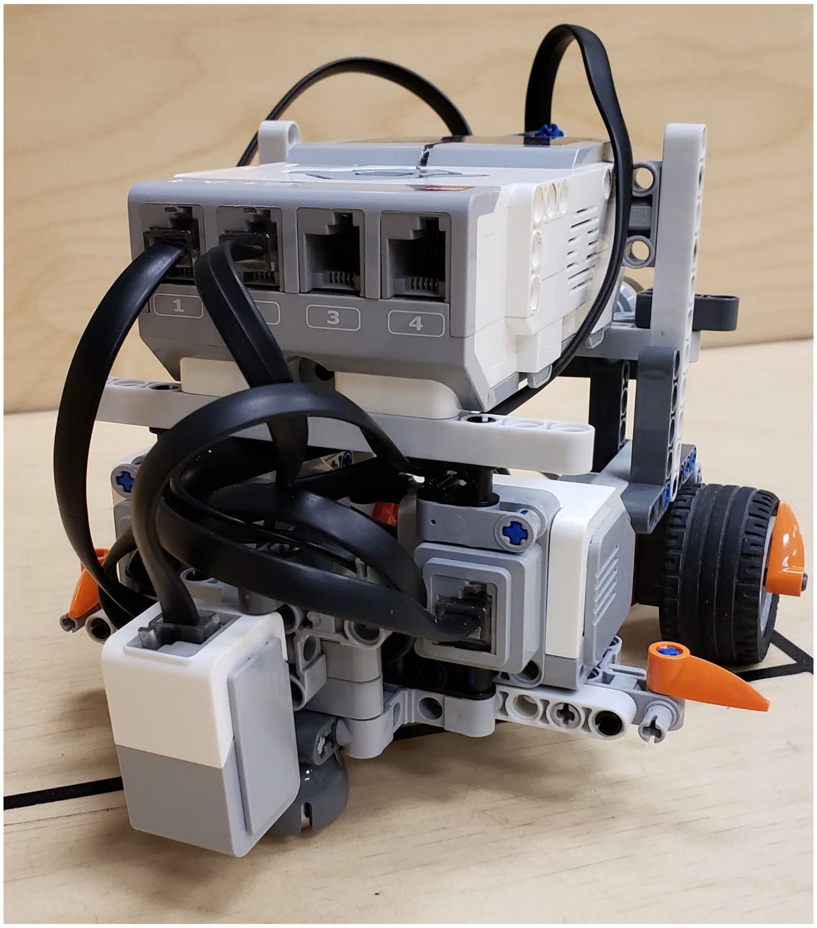
Lab 4: Report

Sameen Mahtab (260737048)

Chen Liu (260776963)

For this lab, we required two sensors; the ultrasonic sensor attached at the middle close to the body in the front of the robot and a light sensor attached close to the ground at the back of the robot. Similar to the previous labs, the robot consists of two wheels powered by motors and one wheel at the back used to hold the rest of the body of the robot up and to balance the robot. The main intention of the Ultrasonic sensor placement is to decrease the error of the distance reported by the ultrasonic sensor and the actual distance that the robot is from the wall. The light sensor is placed at the back so that when the robot is rotating, the sensor comes across all of the lines.

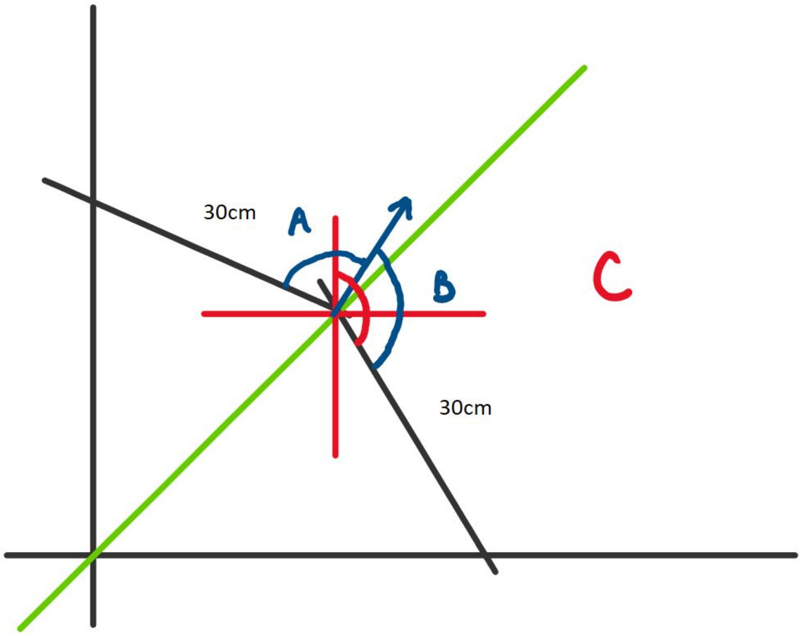


For the software, we applied the ideas used in the previous labs. We used the navigation and odometer classes that we used in the previous labs. The two new classes that we added for this lab are Ultrasonic and light localization. The ultrasonic localization was implemented to handle two cases, falling edge and rising edge.

A constant distance is set as 30 cm for both falling and rising edges. The ultrasonic measures the distance to the wall as the robot is rotating. The value detected by the US sensor is compared to the constant value(30cm). Based on logics, when the robot turns to the wall, the distance detected will decrease. Falling edge method first let the robot turn anticlockwise to the wall, the distance detected continue decreasing, when it detects a distance that is equal to 30 cm, the odometer record its turning angle at the moment. Meanwhile, it starts to turn clockwise (away from the wall). when it sees the distance equal to 30 cm again(it sees the other wall), the odometer records the turning angle again. Based on the trigonometry, the ideal zero angle direction can be easily calculated.

Rising edge works in an opposite way, the robot does not stop turning anticlockwise the distance smaller than 30 cm is detected, but record the turning angle when it detects the distance greater than 30 cm again, and then turn clockwise. The robot will do the work again by looking for another turning angle that have the detected distance greater than 30 cm in the first place. Using the two angles measured and trigonometry, the zero angle direction can also be calculated.

The calculation can be explained by the picture below



C=(A+B)/2+45

Test Data:

Rising Edge:

Trial	Sensor Angle Error	X final (cm)	Y final (cm)	Angle final (Degrees)	Euclidean error (cm)
1	-4	-3.4	-0.1	12	3.4
2	-1	-5.2	0.5	5	5.3
3	-7	-0.8	0.7	3	1.06
4	-11	3.1	-0.1	-2	3.1
5	-5	5.2	1.4	4	5.7
6	-3	-1.2	-0.4	-6	1.3
7	-7	-3.9	0.1	2	3.9

8	3	3.1	-3.1	5	4.38
9	-6	-4.0	2.8	1	4.8
10	-8	-1.0	-0.4	2	1.2

Mean of data:

Rising edge

Sensor angle error (degree)	Final angle error	Euclidean Error(cm)
-4.9	2.6	3.29

Sensor angle mean: (-4-1-7-11-5-3-7+3-6-8)/10= -4.9

Final angle error: (12+5+3-2+4-6+2+5+1+2)/10= 2.6

Euclidean error is calculated by taking the values of x final and y final to the power of two, adding them and finding the square root of the result.

Falling edge:

Trial	Sensor Angle Error	X final (cm)	Y final (cm)	Angle final (Degrees)	Euclidean error (cm)
1	-3	-2.1	2.1	1	2.9
2	-12	4.2	-3.1	2	4.6
3	-5	0.3	1.4	7	1.7

4	10	1.5	2.1	3	2.6
5	-8	-3.1	-0.4	5	3.3
6	1	2.1	-0.5	-1	2.4
7	-6	-2.1	3.2	-5	3.8
8	-12	-3.5	-1.2	4	3.7
9	-7	1.2	-1.2	3	1.69
10	3	-3.0	0.5	-1	3.2

Sensor angle error mean: (-3-12-5+10-8+1-6-12-7+3)/10=-3.9

Final angle error mean: (1+2+7+3+5-1-5+4+3-1)/10=1.8

Falling Edge

Sensor angle error (degree)	Final angle error	Euclidean Error(cm)
-3.9	2.6	3.36

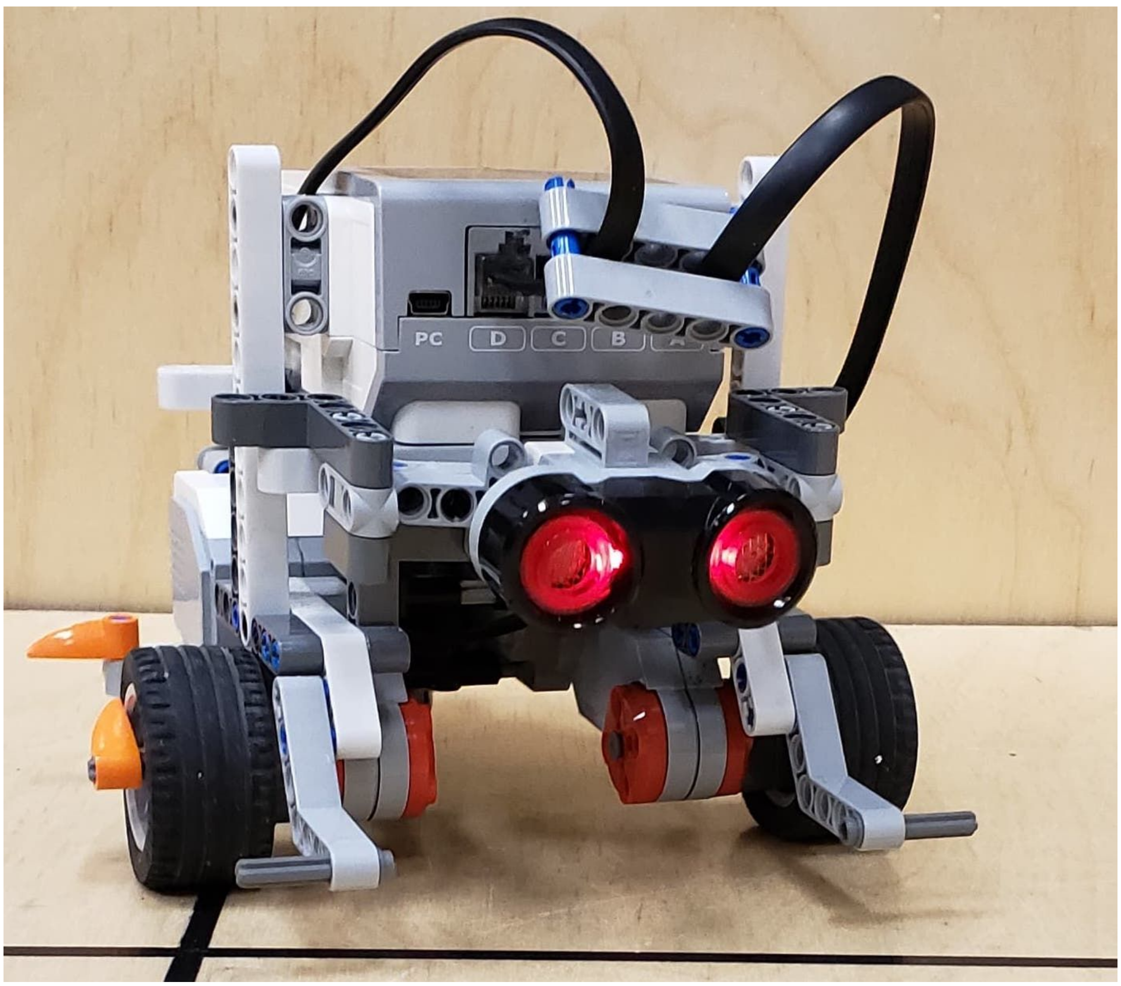
Standard deviation:

The standard deviation is calculated by using the formula:

$$s_x = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}}$$

<u>Rising edge sample SD:</u>

Sensor angle error (degree)	Final angle error	Euclidean Error(cm)
7.1	4.2	



Falling edge sample SD:

Sensor angle error (degree)	Final angle error	Euclidean Error(cm)
7.1	4.2	

Observations and conclusions:

Q: Which of the two localization routines worked best?

A: The falling edge routine worked best. Results were better when the light sensor was used. The main reason is that when using the light sensor, we are taking more factors into account, it utilized one more sensor to improve the accuracy. It senses the location of the grid and also the distance from the walls to fix the angle of the robot.

The chance of error with the ultrasonic sensor is also greater than that of the light sensor because it simply has to return a wider range of readings in real time. On the other hand the light sensor has to sense if there is a line or not.

Q: Was the final angle impacted by the initial Ultrasonic angle?

A: Ultrasonic angle did affect the final angle. If there was a large error in the Ultrasonic angle, the robot might have hit the wall.

Q: What factors do you think contributed to each method?

A: For the Ultrasonic sensor, changes in lighting was a big factor, changes in lighting leads to the sensor fetching incorrect data. . The precision of the odometer with regard to the synchronisation of the two motors was also a big factor.

For the light sensor, changes in lighting was much less of a factor as it was placed very close to the surface. A factor that affected the light sensor was the clearness of the lines. If the variation in colour of the of the lines and the floor is similar, then the light sensor might fail to detect the lines.

Also if the light sensor is not mounted firmly and if the sensor is let to oscillate, then the readings might not be accurate.

Q: How do changing light conditions affect light localization?

A: Changing light conditions affected light localization in that when the robot was in a dark condition, there would be a lesser difference between the intensity from a tile and a line. But as the robot was tested in a well lit room most of the time, this was not much of a problem. And the room would have to be very dimly lit for there to be any error to be shown by the light sensor.

Further improvements:

Q:Propose a software or hardware way to minimize error from the ultrasonic. sensor

A: The ultrasonic sensor can be fixed in a motor. The motor would be used to swivel the sensor from one side to another to sense the distance of the wall on each side. Using this method the robot would not have to be rotated to find those readings, this would decrease the errors greatly.

Another implementation would be to use more light sensors. If we put two light sensors to two sides of the robot, then the robot passes a line. If the robot is not completely straight, the light sensors will detect the lines at different times, accordingly the robot can me made to travel straight.

Q: Propose another type of localization other than rising edge or falling edge.

A: Another form of localization could be circular localization. The robot would first rotate 360 degrees and would store the angles in a 2-D array. The the distance from the walls would be calculated from the angles that are found.

Q: Discuss how and when light localization could be used outside of the corner to correct odometry errors. Eg. having navigated to the centre of a larger floor.

A: The robot can be made to keep on going straight until it crosses a black line. After detecting the line the robot will move backwards until it reached a tile. After doing that the robot will move forward to half the distance it travelled backwards. It will turn 90 degrees and will go forwards and backwards again. After collecting all of that data the robot will know the coordinates of where it was and then correct itself.