ECSE 211
Design Principles and Methods
Fall 2019

Lab 4 Report:
Localization

Group 21
Culha, Dafne 260785524
Udupa, Suhas 260867826

Section 1: Design Evaluation

In this laboratory, the goal was to design and implement an autonomous robot that could orient itself using landmarks in its environment. Using the point intersection between two walls and ground grid lines, the robot was required to determine its position in its environment.

We began this laboratory with the hardware design and to modify our previous robot model to best fit the criterions of the new laboratory. As always, we used two wheels, attached to the large EV3 motors on either side of the brick, a caster ball to provide stability and support to the front wheels, and the ultrasonic sensor from the previous laboratory, which was placed at the front of the robot at the level of the brick to take accurate measurements of the robot's distance with respect to the walls. We wanted to keep the robot's design as compact as possible and to also place more weight on the wheels for better traction to the ground. Placing the EV3 processor brick directly above the wheels increased wheel traction as the added pressure helped with the wheels' grip of the ground surface. This allowed for better transfer of the robot's rotational movement into lateral movement. In order for the robot to detect the black grid lines on the floor, we added a color sensor to the back of the robot, which was needed for the light localization portion of the laboratory. This placement allowed us to maintain our robot's compact nature and to not have to bring any significant changes to our design from the previous laboratory. We also adjusted the sensor to be as close to the ground as possible (as permitted by our design), to enhance the accuracy of the color intensity reflection readings and to reduce any noise. Figure 1 below shows the hardware design of our robot.

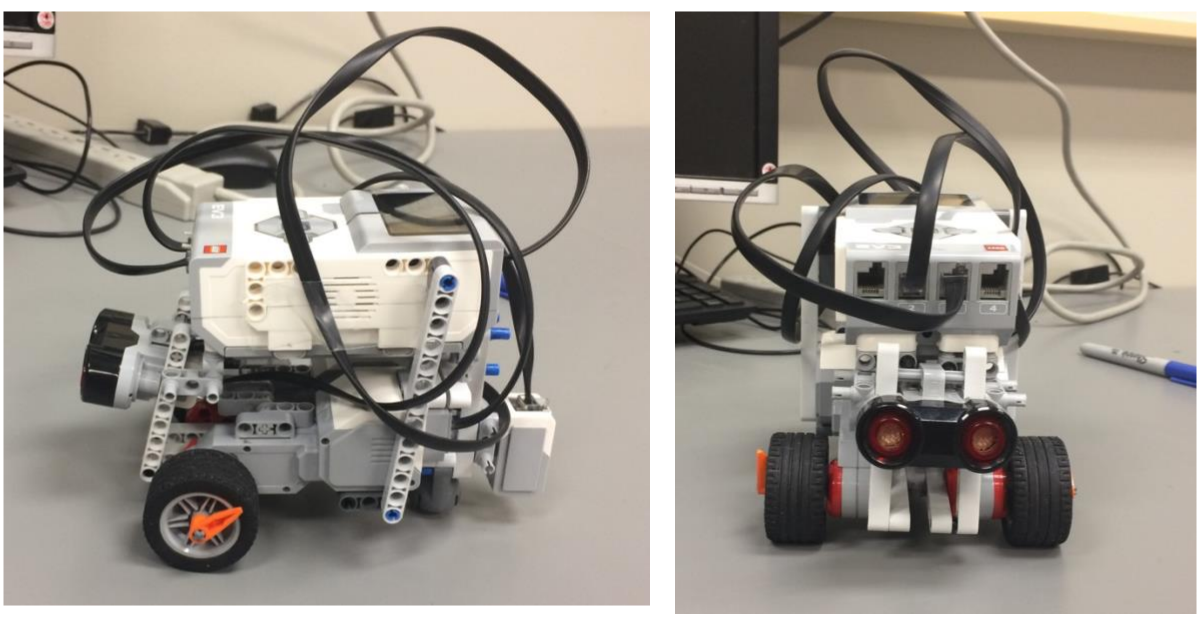

Figure 1: Hardware design of the robot

In terms of the software component of this laboratory, the robot orients itself using the ultrasonic sensor through two different methods which estimate an error correction for the angle of the robot: the falling edge method and the rising edge method. The falling edge method records the angles at which the robot turns to face two consecutive walls, while the rising edge method captures the angles at which the robot turns away from two consecutive walls. These are then manipulated to determine the actual angle of the robot and to make the latter turn to face due north, as depicted in Figure 2.

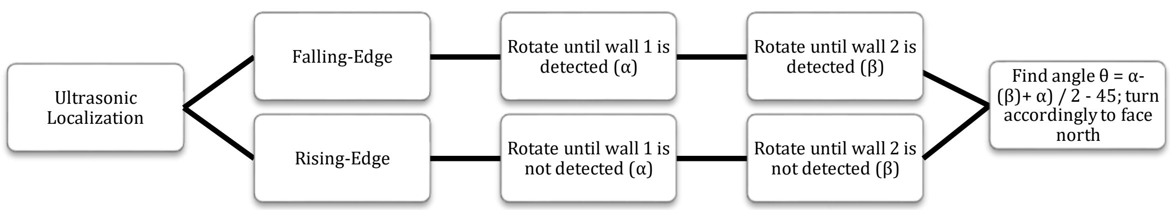

Figure 2: Software design of the Ultrasonic Localization class

Then, after having calibrated the robot's angle, the light localization uses the color sensor in order to align itself with the x and y-axis and to properly position the robot at (1,1) facing the same 0-degree angle. It does so by first travelling along the y-axis until it encounters its first black grid line. When the black line is detected, the robot rotates 90 degrees in the clockwise direction. It then travels until it reaches its second black line, after which it backs-up until its wheels are in line with the y-axis. The robot then rotates back 90 degrees in the counter-clockwise direction to orientate itself in the y-direction. Following this, the robot backs-up once again a certain distance, this time to correct its position in the x-axis and to be in line with the coordinate position (1,1). Figure 3 below explains the Light Localization class in visual detail.

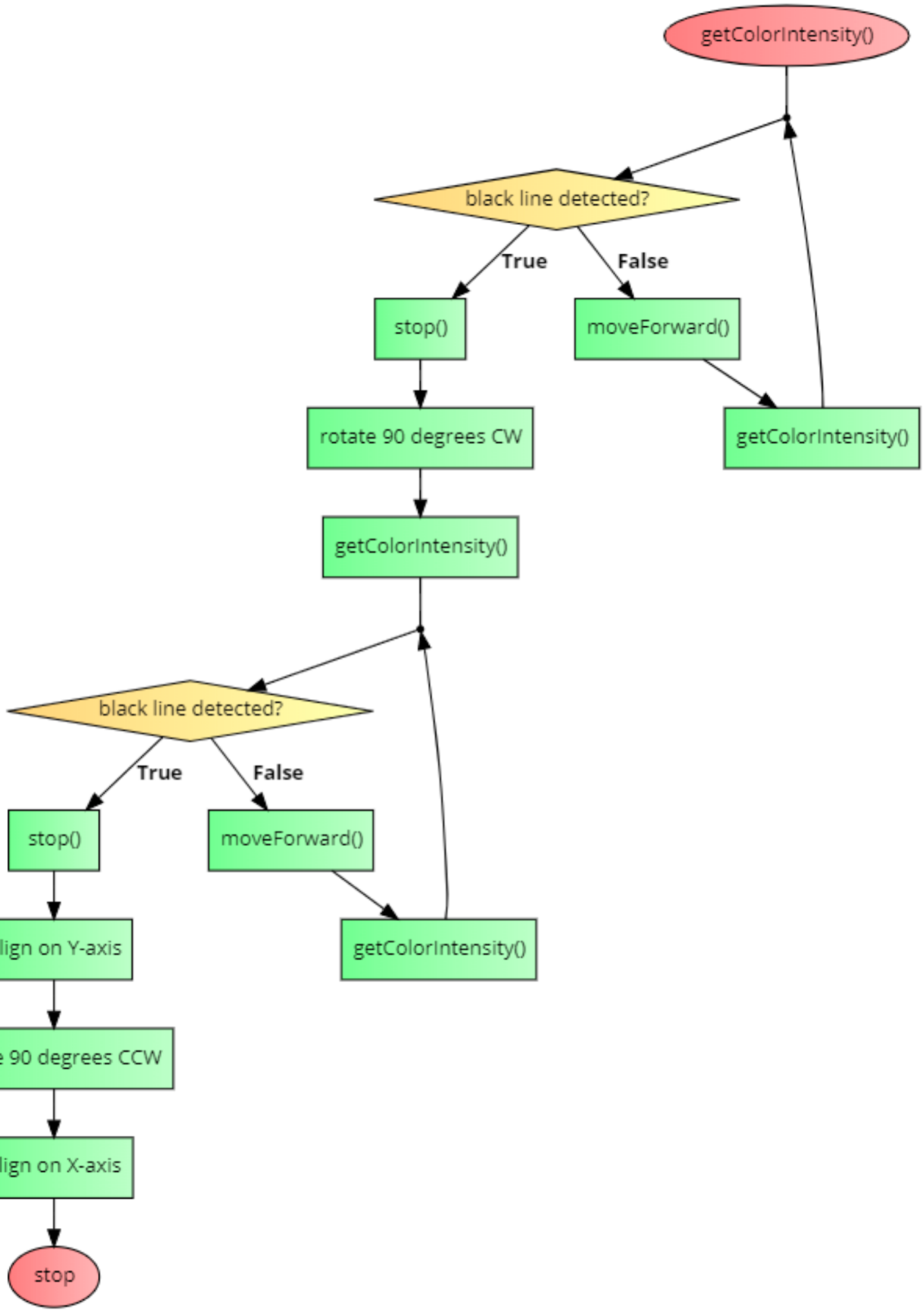

Figure 3: Software design of the Light Localization class

Section 2: Test Data

The ultrasonic angle error has been calculated by getting the absolute value of the difference between the expected angle and the measured angle, as seen in equation 1 below.

$$\theta_E = |\theta(measured) - \theta(expected)°| \tag{1}$$

The final angle error has been calculated using the same equation as the ultrasonic angle error and by following the same logic.

The Euclidean distance error has been calculated using equation 2.

$$\varepsilon = \sqrt{(X_m - X_o)^2 + (Y_m - Y_o)^2} \tag{2}$$

Table 1 presents the ultrasonic angle error, the Euclidean distance error and the final angle error for each of our trials in our localization tests using the rising edge localization routine.

Table 1: Data from Localization Tests using Rising Edge

Trial Number	**US Angle Error (°)**	**Euclidean Distance Error (cm)**	**Final Angle Error (°)**
1	3	1.89	8
2	4	1.53	5
3	2	2.21	6
4	5	1.52	9
5	3	1.32	4
6	4	1.63	6
7	2	1.12	9
8	3	1.42	5
9	6	2.17	8
10	2	1.13	4

Table 2 presents the ultrasonic angle error, the Euclidean distance error and the final angle error for each of our trials in our localization tests using the falling edge method.

Table 2: Data from Localization Tests using Falling Edge

Trial Number	**US Angle Error (°)**	**Euclidean Distance Error (cm)**	**Final Angle Error (°)**
1	1	0.64	0
2	2	0.39	3
3	3	0.18	1
4	1	0.21	4
5	1	0.43	2
6	2	0.33	1
7	2	0.26	1
8	1	0.73	3
9	3	0.67	2
10	1	0.52	1

Section 3: Test Analysis

The mean (μ) for the US Angle Error is the sum of all the US Angle Errors, (Z_i) divided by the number of trials (N) and it can be calculated by using equation 3. The calculation of the mean of our ultrasonic angle error in our falling edge localization routine can be found below. The same equation has been used to calculate the mean of our ultrasonic angle error in our rising edge localization method as well as the mean value for Euclidean distance error and final angle error for both localization routines.

$$\mu = \frac{1}{N}\sum_{i=1}^{N} Z_i \tag{3}$$

$$\mu = \frac{1+2+3+1+\cdots+3+1}{10} = 1.7$$

The standard deviation σ of the ultrasonic angle errors was calculated by using equation 4. The standard deviation gives the average deviation of each error in comparison with the mean. This makes us understand how large the variation between the errors truly is. In the following equation, Z_i is the error for each trial, N is the number of trials and μ is the mean. Our calculation of the standard deviation of our ultrasonic angle errors in our falling edge localization routine can be found below. The same method has been followed to calculate the standard deviations of our ultrasonic angle errors in our rising edge localization method as well as the standard deviation values for Euclidean distance errors and final angle errors for both localization routines.

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(|Z_i| - \mu)^2} \tag{4}$$

$$\sigma = \sqrt{\frac{(1-1.7)^2 + (2-1.7)^2 + (3-1.7)^2 \ldots + (1-1.7)^2}{10}} = 0.82$$

The mean and the standard deviation of ultrasonic angle error, Euclidean distance error and final angle error when using rising edge localization are presented in Table 3.

Table 3: Mean and Standard Deviations of Calculated Errors when Using Rising Edge

	Mean (μ)	**Standard Deviation (σ)**
US Angle Error (°)	3.40	1.35
Euclidean Distance Error (cm)	1.59	0.39
Final Angle Error (°)	6.40	1.96

The mean and the standard deviation of ultrasonic angle error, Euclidean distance error and final angle error when using the falling edge localization routine are presented in Table 4.

Table 4: Mean and Standard Deviations of Calculated Errors when Using Falling Edge

	Mean (μ)	**Standard Deviation (σ)**
US Angle Error (°)	1.70	0.82
Euclidean Distance Error (cm)	0.44	0.20
Final Angle Error (°)	1.80	1.23

Section 4: Observations and Conclusions

Which of the two localization routines performed the best?

When comparing the mean and standard deviation values of the US Angle Error and Final Angle Error, it can be seen that the Ultrasonic localization is more accurate with its lower mean error and more consistent with its readings due to its lower standard deviation. In terms of the falling edge method versus the rising edge approach, based on the test results, the falling edge localization routine performed better for the ultrasonic localization than the rising edge approach, as displayed by their respective mean US angle error. Indeed, the average error of the ultrasonic angle was of 1.7° for the falling edge routine, which was less than the average error of the rising edge method (3.4°). This higher accuracy later impacted the light localization, with a low Euclidean error of only 0.44 cm compared to 1.59 cm. We assumed that this is because the falling edge routine detects walls and uses the robot's position with respect to these walls to orient itself whereas the rising edge localization routine detect open spaces, so it is more prone to error and false data.

Was the final angle impacted by the initial ultrasonic angle?

The test results have helped conclude that the final angle of the robot was in fact impacted by the initial ultrasonic angle. On average, the error of the ultrasonic angle was smaller using the falling edge method, and this consequentially led to a smaller average final angle error. As a matter of fact, following the falling edge localization, the average error was of only 1.7° compared to a much larger 3.4° error following the rising edge localization routine. This delta is due to the fact that, at the start of the light localization, the robot presumes it is facing exactly north. If the initial ultrasonic angle has an error, then this maintained throughout the robot's trajectory and thus factors proportionally into the final angle.

What factors do you think contributed to the performance of each method?

Both methods are affected by the accuracy of the odometer since they both rely on odometer values in the calculations required to direct the robot. However, for this lab, the odometer values were relatively accurate since the robot did not need to cover a large distance which increases the chance of error accumulation. Another factor that highly contributed to the success of the performance of both methods was the reliability and precision of the values collected by their respective sensors. The ultrasonic sensor often gave false and inaccurate readings of distance, which often error in the ultrasonic localization angle. One factor that helped the performance of the localization was the speed of the robot. By having the robot cruise at a relatively low speed, the sensors had more time to collect data relative to the environment and more accurate readings could hence be done. Indeed, most of the measurements done by the sensors had to be very precise in order for the robot to localize itself as accurately as possible, so having proper readings was crucial. Finally, noise in the ambient lighting of the room contributed to the decrease of the performance of the localization as it interfered with the readings of the color sensor. For instance, if the room is dark, the sensor has more difficulty picking up on the contrast between the lines and the board.

How does changing light conditions impact the light localization?

Changing the light conditions will impact the light localization. In this lab, the light sensor's role is effectively to detect the black grid lines on the ground. This is done measuring the contrast between the board and the line, and when the contrast exceeds a certain threshold, the sensor determines that it traversed a black grid line. Therefore, if there isn't enough light in a room or if there is a shadow over the robot, this would surely impact the light localization as the color sensor would take false readings. As a matter of fact, the contrast wouldn't be high enough for the robot to recognize the presence of a line, consequently impacting the light localization. On the other hand, if there is too much light in the room, this would also impact the light localization as the black lines would then appear lighter than they truly are, thus affecting the contrast measure. Hence, going back to the previously discussed threshold value, if this constant is determined during tests done in a very illuminated room, it will be affected by the high intensity of the light. When the light localization is after used in a less illuminated room, the threshold would need to be changed in order to adapt to the different values returned by the black lines. A lack of adjustment could lead to the sensor either reading too many or too few lines.

Thus, in order to reduce the impact of changing light conditions on the light localization, we could first start by placing the sensor very close to the ground. This way, the point being polled beneath the sensor would not receive much exterior light initially, due to the sensor's shadow, and the red beam that the sensor outputs would be sufficient to detect grid lines with enough precision and accuracy. Also, using a differential filter which detected lines by the difference in light reflection between the current position and the previous sensor measurement would be more beneficial than using a pre-set threshold value as discussed above. That way, if the lab lighting conditions were altered, the difference in reflection would still be the same.

Section 5: Further Improvements

Propose a software or hardware way to minimize errors in the ultrasonic sensor.

The biggest source of error caused by the ultrasonic sensor was the presence of noise. Indeed, since the sensor often presents false values, it would be beneficial to implement a stronger and sturdier filter, which would eradicate false readings. The filter used for our lab removed the most extreme outliers, but unfortunately, many false values were still kept. A plausible solution would be to apply a median filter, which would greatly minimize the error as . This filter takes in a set number of past values given by the sensor and calculates the median, which would allow spikes in distance values to be corrected. Then, the current sensor value is replaced by the median and so on. This effectively removes any false values and would further ensure that unreasonably large changes in distance values are ignored and that sudden random fluctuations in distance readings due to the sensor's low accuracy and precision are eradicated.

Propose another form of localization other than rising-edge or falling-edge.

An alternative form of localization would have been a combination of the rising edge and falling edge trigger. We could measure the angle between the first falling edge and the first rising edge. In this case, the angle change would have been a mean of the differences between both measured angles. This is illustrated in the figure below.

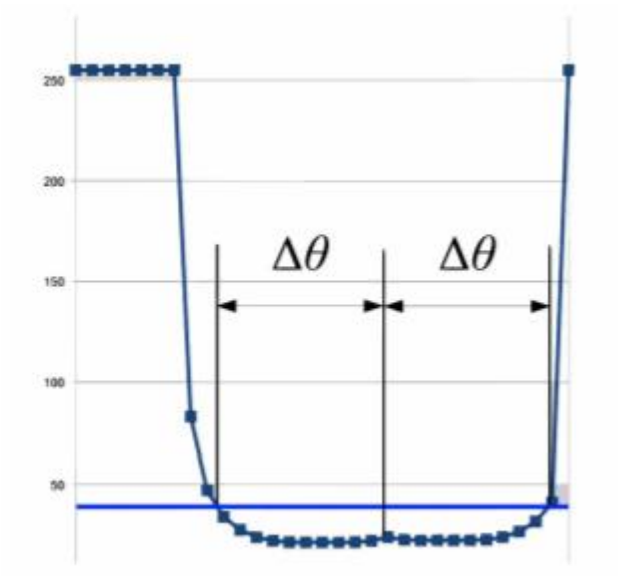

Figure 4: Alternative form of localization

Discuss how and when light localization could be used outside of the corner to correct Odometry errors, e.g. having navigated to the middle of a larger floor.

The light localization could be used to systematically and periodically check the position of the robot along the lines of the grid during navigation. Indeed, when using the ultrasonic sensor to localize the robot, it is possible to keep check of the lines crossed using the color sensor. When the robot passes over an intersection between the x-axis and the y-axis, the robot can use this information to pinpoint the robot's real orientation and to correct its expected position relative to its actual position. Indeed, at each intersection crossing, by performing a light localization maneuver, the robot can correct the odometer error. By doing so, the robot can more accurately navigate to any point in the middle of the floor.