ECSE 211 Winter 2020

Lab 3: Localization

Group 54

Joanna Koo (260865846)

Muhammad Mustafa Javed (260808710)

Due February 6th, 2020

Table of Contents

Section 1: Design Evaluation

Workflow:

The objective of the lab was to localize the robot with respect to the grid system. In order to correct its orientation to 0°, the robot must use the ultrasonic sensor, and move to the (1, 1) position on the grid using the ultrasonic sensor or the light sensor. We approached the design problem by adding the necessary hardware (ultrasonic sensor and light sensor), and then working on the software. The orientation and localization using the ultrasonic sensor was set as the priority, with the light sensor to be implemented if the resources and time allowed.

Hardware Design:

The ultrasonic sensor was positioned on the center axis of the robot so that the rotation of the robot on the perpendicular tile line resulted in the rotation (and not the displacement) of the ultrasonic sensor. The light sensor was added to the front-right side so that when the robot is moving with a black line in its center, it could detect another line crossing its path as shown in *Figure 1*. The light sensor was placed as close to the ground as possible to minimize the noise from ambient sources. The final robot design can be found in *Figures 2, 3* and *4*.

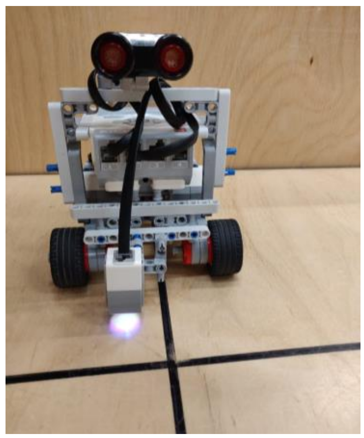

Figure 1. Light sensor placed at the right to detect a perpendicular line.

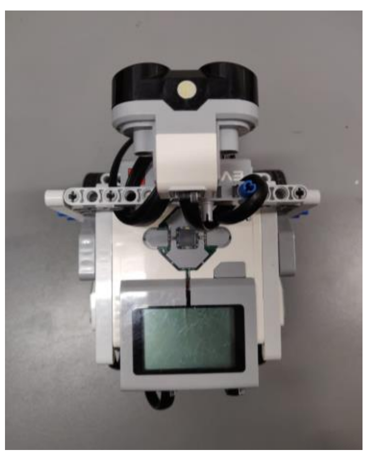

Figure 2. Bird's eye view of the robot.

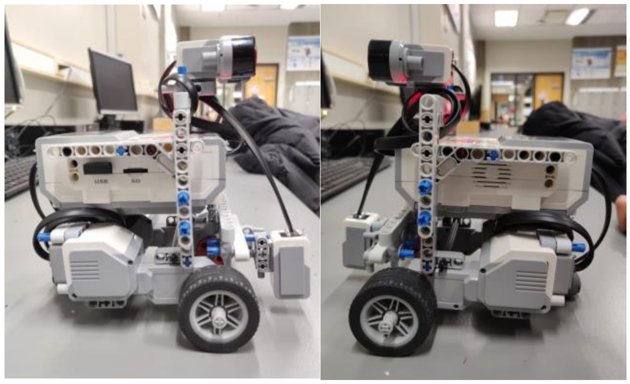

Figure 3. Left and right side profiles of the robot.

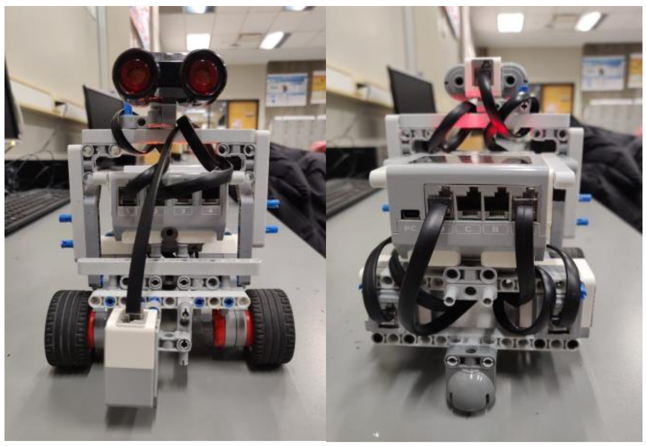

Figure 4. Front and back profiles of the robot.

Software Design:

The software solution implemented to orient the robot to 0° was based on the *falling edge* principle. The falling edge is characterized by a decreasing pattern of distance measured by the ultrasonic sensor. The robot turns clockwise until it detects a falling edge, determined when the distance becomes less than a particular value (band center) kept at 35 cm with a noise margin of 2 cm. The deflection angle, α**,** for the first edge is stored. The robot then turns counter-clockwise until it detects the second falling edge. The angle, β, is recorded. The correction required to orient the robot to 0° is calculated using *Equations 1* and *2*. To move to the (1, 1) position, the robot first turned 90 degrees counterclockwise to face the x = 0 wall and moved backwards until the distance read by the ultrasonic sensor is equal to the tile size. Afterwards, it turned an additional 90 degrees counterclockwise to face the y = 0 wall and carried out the same procedure to finalize its location at (1,1). The flowchart illustrating the software functionality is shown in *Figure 5*.

Equation 1.

$$DeltaTheta = 45 - \frac{\alpha - \beta}{2} \ if \ \alpha < \beta$$

Equation 2.

$$DeltaTheta = 225 - \frac{\alpha - \beta}{2} \ if \ \alpha > \beta$$

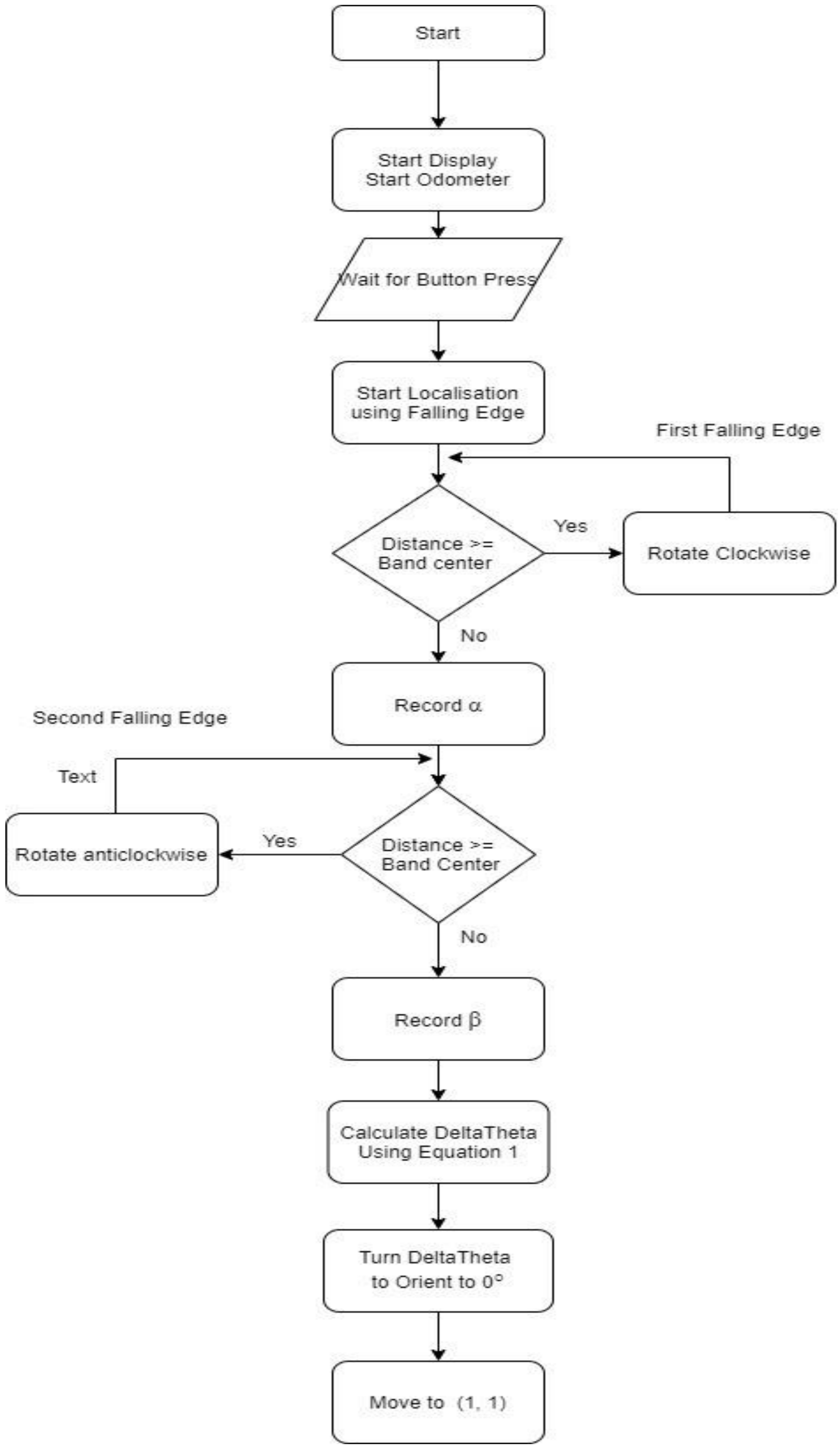

Figure 5. Flowchart for software functionality.

Section 2: Test Data

10 independent trials were performed for the ultrasonic localization test using the *falling-edge* method, and is recorded in *Table 1*.

Trial	Ultrasonic angle error (°)	Euclidean distance error (cm)	Final angle error (°)
1	0.0	2.0	0.0
2	1.0	1.1	1.0
3	2.0	1.0	1.0
4	3.0	1.4	1.0
5	1.0	1.8	2.0
6	0.5	2.0	2.0
7	0.0	1.1	1.0
8	0.5	1.4	1.0
9	0.0	2.0	0.0
10	1.0	1.5	0.0

Table 1. 10 independent trials for ultrasonic localization.

Section 3: Test Analysis

The mean and standard deviation for ultrasonic angle error, Euclidean distance error and final angle error are computed in *Table 2*. *Equations 3* and *4* were used to calculate the mean and the standard deviation, respectively.

Equation 3.

$$\bar{a} = \frac{1}{n}\sum_{i=1}^{n} a_i$$

Equation 4.

$$\sigma_a = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(a_i - \bar{a})^2}$$

	Ultrasonic angle error (°)	**Euclidean distance error (cm)**	**Final angle error (°)**
Mean	0.9	1.5	0.9
Standard deviation	0.9	0.4	0.7

Table 2. The mean and standard deviation for the trials.

As seen in *Table 2*, the mean for both ultrasonic angle error and final angle error is 0.9°, and the mean for the Euclidean distance error is 1.5 cm, which are fairly accurate. The small standard deviation values, 0.9°, 0.7° and 0.4 cm, indicate the small dispersion of reported data.

Section 4: Observations and Conclusions

1. *Describe other possible localization techniques and explain why you chose yours.*

   Another possible technique that can be implemented is the rising edge principle. Using this principle, the robot stores the angle α when it detects the first rising distance pattern, then turns in the other direction and records the angle β when it detects the second increasing distance pattern. The *Equations 1* and *2* are used to calculate the angle to orient the robot to 0° bearing. However, we chose the falling edge method in this lab, because given the initial conditions of the robot facing away from the wall, we found that the robot took less time and fewer rotations to orient to 0° when using the falling edge method compared to the rising edge method.

2. *Was the final angle impacted by the initial ultrasonic angle?*

   When using the ultrasonic sensor for localization, the final angle depended on the precise motion of the robot to the (1, 1) position on the grid, which depended on the ultrasonic angle of the robot at the beginning of the motion. The algorithm for the movement to (1, 1) position assumes that the robot is perfectly oriented to 0°. The final angle is a 90° rotation in clockwise and counterclockwise directions at particular points during the motion. Therefore, any error in the ultrasonic angle also induces an error in the final angle.

3. *What factors do you think contributed (positively and negatively) to the performance of your method?*

   The biggest factor that contributed to the performance of our methods were the sensors. The ultrasonic sensor was often unreliable as it frequently returned erroneously high values. These false readings were fatal for the localization process, since the sensor would believe that it was turning away from the wall.

   Another factor was that the initial position of the robot had to be precisely on the diagonal of a tile for the localization to work perfectly. If the sensor was even slightly off the diagonal, the ultrasonic angle would end up having an error.

   The last factor that contributed to the performance was the slip of the tires. Dust and scratches on the board might have resulted in a change of testing environment, and possibly contributed in returning an erroneous result.

4. *Do you think that the ultrasonic sensor or the light sensor method performs better? (If you did not do the bonus, answer based on the reliability/quality of the readings of the light sensor vs. ultrasonic sensor). Which method do you plan to use for navigating in future labs?*

   Based on our tests, the light sensor method outperformed the ultrasonic sensor. The advantage was evident as the final angle in the light sensor method was not affected by the initial ultrasonic angle error. At the (1, 1) position, the robot stops rotating when the light sensor detects the horizontal (y =1) line and orients to 0°. Ceteris Paribus, we plan on using the light sensor for navigating in future labs.

Section 5: Further Improvements

1. *<u>Propose a software or hardware way to minimize errors in the ultrasonic sensor.</u>*

   One way to make a software improvement to minimize errors in the ultrasonic sensor is to set up the constants BAND_CENTER and NOISE_MARGIN well-adjusted for different starting positions of the robot. As the starting position can lie anywhere along the diagonal line that links (0,0) and (1,1), the difference in starting positions can be a maximum of approximately 40 centimeters, i.e. when the robot starts right beside the walls and when the robot starts near (1,1). Consequently, having one fixed value for BAND_CENTER constant and another fixed value for NOISE_MARGIN constant for trials starting at different points along the diagonal line is not efficient. For example, from multiple tests, we have observed that when the robot started at a point far from the origin, it often failed to detect any falling edges and to localize. As a solution, a new method in the UltrasonicLocalizer class can be implemented to return the distance from the walls at starting position, and use this value as an offset to an optimized band center constant. i.e. BAND_CENTER = startingDistance + BAND_CENTER_CONST. By making these adjustments, ultrasonic sensor performance will be more accurate regardless of the different starting positions along the diagonal line.

2. *<u>Propose another form of localization.</u>*

   Another form of localization can be implemented by having two ultrasonic sensors, each installed on the right and the left side of the robot. The robot would turn 360 degrees, in a similar manner to the ultrasonic localization in this lab, and determine when the wall distance reading from both sensors return the same value within an error margin. This method can localize the robot effectively, because both sensors being approximately the same distance away from the wall means the robot is square to that wall length. Accordingly, using the angles collected from detecting the walls to calculate the deltaTheta to correct the odometer readings would orient and localize the robot in the same manner as using falling-edge method.

3. *<u>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.</u>*

   Light localization can be used repeatedly when the robot approaches the black lines on the floor. For example, for each iteration of a given time interval, the robot would stop and approach a grid line using the light sensor, turn 90 degrees, then drive until the sensor detects another line. Once the robot reaches a grid line intersection, the odometer values for X, Y and theta would be stored, then using light localization at the grid point, the corrections to the odometry errors would be performed.

4. *<u>How could the robot quickly localize given two light sensors?</u>*

   The two light sensors, each mounted on the front-right and the front-left corner of the robot, enable the localization process to be more efficient. If a time difference in the detection of the black line from each light sensor is observed, the motor speed is adjusted accordingly to correctly orient the robot. For example, if the right light sensor detects the line first, the robot stops the right motor and keeps rotating the left motor until the left sensor detects the black line as well. Through an accurate localization procedure using two light sensors, the robot quickly and accurately locates itself at a (1,1) coordinate with a 0 degree angle.