

Steven Cangul (260744412) and Noah Levine (260684940)

ECSE 211 - Lab 4: Localization

Data

*Table 1: Localization with Falling Edge Routine*

<b>Odometer X (cm +/- 0.05cm)</b>	<b>Odometer Y (cm +/- 0.05cm)</b>	<b>Actual X (cm +/- 0.05cm)</b>	<b>Actual Y (cm +/- 0.05cm)</b>	<b>Error X (cm +/- 0.05cm)</b>	<b>Error Y (cm +/- 0.05cm)</b>
-0.8	-0.9	0.2	0.7	1.0	1.6
0.3	-0.5	0.7	0.8	0.4	1.3
0.4	-0.8	1.0	-0.5	0.6	0.3
-0.3	0.6	0.6	1.0	0.9	0.4
-1.0	1.4	0.9	-0.2	1.9	1.6
-0.2	-0.2	1.1	-0.6	1.3	0.4
0.7	0.0	0.8	-0.2	0.1	0.2
1.4	-0.1	0.2	-0.4	1.2	0.3
0.4	0.0	0.8	-0.5	0.4	0.5
0.2	-0.2	0.7	0.3	0.5	0.5
			<b>MEAN (cm)</b>	<b>0.83</b>	<b>0.71</b>
			<b>STANDARD DEVIATION (cm)</b>	<b>0.51</b>	<b>0.52</b>

*Table 2: Localization with Rising Edge Routine*

<b>Odometer X (cm +/- 0.05cm)</b>	<b>Odometer Y (cm +/- 0.05cm)</b>	<b>Actual X (cm +/- 0.05cm)</b>	<b>Actual Y (cm +/- 0.05cm)</b>	<b>Error X (cm +/- 0.05cm)</b>	<b>Error Y (cm +/- 0.05cm)</b>
0.7	0.4	0.0	-0.6	0.7	1.0
-0.9	0.5	0.7	0.2	1.7	0.3
0.1	0.9	0.0	-0.5	0.1	1.5
-0.4	0.5	0.9	-0.2	1.4	0.7
-1.0	0.1	1.2	-1.0	2.2	1.0
0.3	-1.0	0.8	0.0	0.5	1.0
0.9	0.3	-0.5	1.0	1.4	0.7
0.1	1.5	-0.3	0.0	0.4	1.5
-0.5	0.5	1.0	0.5	1.5	0.0
0.0	-0.9	-0.6	-0.1	0.6	0.8
			<b>MEAN (cm)</b>	<b>1.05</b>	<b>0.86</b>
			<b>STANDARD DEVIATION (cm)</b>	<b>0.64</b>	<b>0.43</b>

### Observations and Conclusions

The falling edge routine performed better in our situation compared to the rising edge. While the falling edge routine makes the robot sweep through the outside of the wall, facing open space, the rising edge method makes the robot sweep through the inside wall. This in fact means that the rising edge routine would make the robot sense a wall and then rotate through an angle theta until it reaches the other wall, all this while facing a wall. This renders the sensor susceptible to returning false readings as the robot would often tend to get too close to the wall. The falling edge method however is sure to always keep a safe distance from the wall which minimizes the chances of error and delivers better and more accurate results.

The reason for which the light sensor provides more accurate results is due to the certainty associated with the results of the light sensor. In fact, the light sensor is affected much less by noise. Additionally, the light sensor is responsible to detecting black lines which create a high contrast with the tan surface color of the test surface. This high contrast makes it easier for the sensor to distinguish between the color of the test surface and the black band.

In this lab, ultrasonic localization was used in order to determine the initial heading of the robot. For the purpose of this lab, the  $0^\circ$  axis was defined as facing away from both walls, with the wall on the x-axis directly behind the robot and the wall on the y-axis to the right of the robot. If we were to change this convention to have the  $0^\circ$  axis to be facing the wall on the x axis while keeping the wall on the y-axis to the right of the robot. We can easily determine our position in x by taking the reading of the ultrasonic sensor when facing the wall on the x-axis. Our y position can be determined by rotating the robot  $90^\circ$  to face the other wall and reading the position reported by the sensor. There is an alternative method that involves placing the robot away from the x wall until a minimum distance is seen. This process can be repeated for the y axis and our (x,y) position can be obtained. However, this method is less accurate as the robot is never at rest and the minimum distance reported by the sensor will be less accurate.

## Error Calculations

Means were computed as follows:

$$MEAN(i) = \frac{\sum Error(i)}{10} \quad [for\ i = x, y]$$

Standard deviations were computed as follows:

$$STDEV(i) = \left( \frac{\sum (Error(i) - MEAN(i))^2}{10} \right)^{\frac{1}{2}} \quad [for\ i = x, y]$$

$$FALLING\_EDGE\_MEAN(x) = \frac{1.0+0.4+0.6+0.9+1.9+1.3+0.1+1.2+0.4+0.5}{10} = \mathbf{0.83cm}$$

$$FALLING\_EDGE\_MEAN(y) = \frac{1.6+1.3+0.3+0.4+1.6+0.4+0.2+0.3+0.5+0.5}{10} = \mathbf{0.71cm}$$

$$RISING\_EDGE\_MEAN(x) = \frac{0.7+1.7+0.1+1.4+2.2+0.5+1.4+0.4+1.5+0.6}{10} = \mathbf{1.05cm}$$

$$RISING\_EDGE\_MEAN(y) = \frac{1.0+0.3+1.5+0.7+1.0+1.0+0.7+1.5+0.0+0.8}{10} = \mathbf{0.86cm}$$

$$FALLING\_EDGE\_STDEV(x)$$

$$\left( \frac{(1.0-0.83)^2+(0.4-0.83)^2+(0.6-0.83)^2+(0.9-0.83)^2+(1.9-0.83)^2+(1.3-0.83)^2+(0.1-0.83)^2+(1.2-0.83)^2+(0.4-0.83)^2+(0.5-0.83)^2}{10} \right)^{\frac{1}{2}}$$

$$= \mathbf{0.51cm}$$

$$FALLING\_EDGE\_STDEV(y)$$

$$\left( \frac{(1.6-0.71)^2+(1.3-0.71)^2+(0.3-0.71)^2+(0.4-0.71)^2+(1.6-0.71)^2+(0.4-0.71)^2+(0.2-0.71)^2+(0.3-0.71)^2+(0.5-0.71)^2+(0.5-0.71)^2}{10} \right)^{\frac{1}{2}}$$

$$= \mathbf{0.52cm}$$

$$RISING\_EDGE\_STDEV(x)$$

$$\left( \frac{(0.7-1.05)^2+(1.7-1.05)^2+(0.1-1.05)^2+(1.4-1.05)^2+(2.2-1.05)^2+(0.5-1.05)^2+(1.4-1.05)^2+(0.4-1.05)^2+(1.5-1.05)^2+(0.6-1.05)^2}{10} \right)^{\frac{1}{2}}$$

$$= \mathbf{0.64cm}$$

$$RISING\_EDGE\_STDEV(y)$$

$$\left( \frac{(1.0-0.86)^2+(0.3-0.86)^2+(1.5-0.86)^2+(0.7-0.86)^2+(1.0-0.86)^2+(1.0-0.86)^2+(0.7-0.86)^2+(1.5-0.86)^2+(0.0-0.86)^2+(0.8-0.86)^2}{10} \right)^{\frac{1}{2}}$$

$$= \mathbf{0.43cm}$$

## Further Improvements

One way to avoid small errors more accurately is by implementing a filter that would take multiple values into consideration. This would in return allow us to give more weight to the values that recur more often in the sensor's reading array. In fact, we could code the robot to consecutively take in a range of values and then calculate the average of those values. This would allow us to have a better representation of the robot's position with respect to a wall as it considers its previous position and will be less susceptible to sudden changes. Of course, the clipping method would still need to be implemented as large values would tend to diverge the sensor's average readings significantly. Another technique we could have used is implementing a Kalman filter that takes in a series of results and outputs a value that are usually more precise than those based on a single measurement using joint probabilities.

One sensor design that would result in us having more accurate and reliable results in implementing an infrared sensor rather than an ultrasonic sensor. In fact, according to the EV3 sensor library, the infrared sensor is capable of measuring distances ranging up to 50 cm to 70 cm. While this is in fact lower than the capabilities of the ultrasonic sensor, the advantage of the infrared sensor is that it allows for greater accuracy when trying to sense objects that are in close proximity to the robot. This is where the ultrasonic sensor fails to accurately read values and returns false readings.

Another form of localization that we could have used is called the Markov localization. This method is based on a probability distribution of the robot's location. Using a selection of sensors, the robot examines its surroundings and adjusts the probability of being in a specific location accordingly. For example, if the robot senses a wall, it will increase the probability of being next to a wall while decreasing the probability of being somewhere else. It is important to note however that it does not set the other probabilities to 0 as noise and errors from the sensors may give return false readings.

(<http://www.cs.cmu.edu/afs/cs/project/jair/pub/volume11/fox99a-html/node3.html#SECTION00021000000000000000>)