Group 53 – Localization
Alessandro Commodari 260636932
Asher Wright 260559393

Lab 4: “Localization” Report

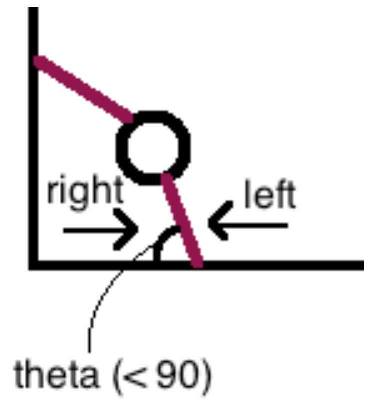
Data

Falling Edge Error (deg)	Rising Edge Error (deg)
4.3	-12.0
3.2	-16.0
-0.1	-9.0
2.5	-8.0
1.8	-14.0
-2.5	-11.0
1.3	-9.0
2.6	-12.0
1.8	-17.0
1.4	-9.0

	Falling Edge	Rising Edge
Mean (deg)	1.63	-11.7
Standard Deviation (deg)	1.87	3.13

Observations and Conclusions

1. The falling edge method performed significantly better. There were a few reasons for this. For one, with the rising edge, there is the anomaly of the sensor reading the gap in the wall (corner) as a maximum distance, and counting it as an angle. Even after solving this issue, though, there is a reason that the falling edge is superior. Since the sensor counted an angle at a distance of roughly 30cm, and the size of the block was 30cm, the sensor was always past the 90-degree mark (theta < 90 in diagram below) when recognizing an angle. This means that, upon overshooting with falling edge, the reading difference would be less than if the robot overshot with rising edge (see diagram).



Since theta < 90, if the sensor is moving right, and it overshoots by time, t, the error will be much larger than if it is moving left, and overshoots by the same time amount. This is because the rate at which the distance is changing is increasing when it is moving right, but decreasing when it is moving left.

2. The light sensor is able to correct the x, y, and theta value of the robot given that it starts in a general position (top right corner, with known orientation). From this position, it is able to correct its position and angle very accurately. On the other hand, the ultrasonic sensor only works perfectly if the robot starts exactly on the diagonal. Every amount it is off the diagonal will correspond to error. This is why the light sensor is more accurate.

3. To find the position of the robot using the ultrasonic sensor, one could take angles at the minima of the ultrasonic sensor distance readings. The angles of the two minima readings would correspond to theta = 180 (along x axis), and theta = 270 (along y axis). The main issue with detecting the minima is that the robot has to consider readings before and after a point X to know if that point is a minimum. Thus, it is necessary for the robot to store values around each point. This can also mean that the angles are not as accurate (depending on how many values you store), or this could be difficult for the robot to process.

Error Calculations

$A_{FE} = \frac{1}{10}(4.3 + 3.2 - 0.1 + 2.5 + 1.8 - 2.5 + 1.3 + 2.6 + 1.8 + 1.4)$
$A_{FE} = \frac{1}{10}(16.3) = \mathbf{1.63\ deg}$

$A_{RE} = \frac{1}{10}(-12 - 16 - 9 - 8 - 14 - 11 - 9 - 12 - 17 - 9)$

$A_{RE} = \frac{1}{10}(-117) = \mathbf{-11.7\ deg}$

The formula for standard deviation is (taken from Wikipedia):

$$s = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{x})^2}.$$

$$S_{RE} = \sqrt{\frac{1}{10-1}\sum_1^{10}(x_i - A_x)^2} = \sqrt{\frac{1}{9}[(4.3-1.63)^2 + (3.2-1.63)^2 + (-0.1-1.63)^2 + (2.5-1.63)^2 + (1.8-1.63)^2 + (-2.5-1.63)^2 + (1.3-1.63)^2 + (2.6-1.63)^2 + (1.8-1.63)^2 + (1.4-1.63)^2]}$$

$$S_{RE} = \sqrt{\frac{1}{9}\ (31.561)} = \sqrt{3.507}$$

$\mathbf{S_{RE} = 1.87\ deg}$

$$S_{FE} = \sqrt{\frac{1}{10-1}\sum_1^{10}(x_i - A_x)^2} = \sqrt{\frac{1}{9}[(-12+11.7)^2 + (-16+11.7)^2 + (-9+11.7)^2 + (-8+11.7)^2 + (-14+11.7)^2 + (-11+11.7)^2 + (-9+11.7)^2 + (-12+11.7)^2 + (-17+11.7)^2 + (-9+11.7)^2]}$$

$$S_{FE} = \sqrt{\frac{1}{9}\ (88.1)} = \sqrt{9.7889}$$

$\mathbf{S_{FE} = 3.13\ deg}$

Further Improvements

1. There are a few ways of avoiding small errors instead of clipping the values. One simple way is to implement a filter, similar to the one implemented in the wall-following code. This filter would ignore values past a certain threshold until they were read a certain amount of times. Another way of preventing small errors that does not ignore any values is by setting up a moving average. This would record the last X values and take an average of them. It could then use this average to determine what it should do at the point. This considers many points instead of a single point, so outliers “do less damage”.

2. Rather than having an ultrasonic sensor, the robot could be outfitted with an infrared sensor. There are a few benefits to this sensor, the most obvious being its speed. The

waves of the electromagnetic radiation sensors (radar, IR, etc.) would all move at the speed of light (much faster than that of sound, obviously). Thus, there would be almost no delay in-between sending out a signal and receiving a signal. This would not only result in more accurate results, but would also allow for higher polling rates. Both of these combined would result in a more accurate localization altogether.

3. Another form of localization is the one hinted at throughout this report. The robot could use a more accurate sensor (or rotate slower) to accurately find the minima around it. Once it found these minima, and it found on which side of the minima was the corner and which was open space, it could determine its position as follows. It would know its x-position based on the sensor distance at the x-minimum. It would know its y-position based on the sensor distance at the y-minimum. Finally, it would know to rotate 180 degrees from the x-minimum to face theta = 0. It could then drive to any point accurately.