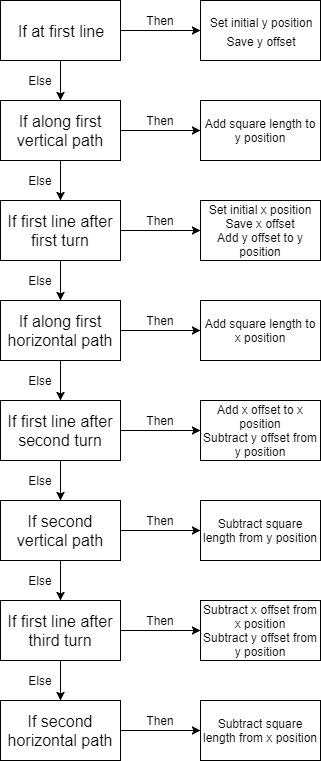
Group 5 - Evan Laflamme (260684099) and Zhiyu Chen (260668145)

ECSE 211 – Lab 2 Report

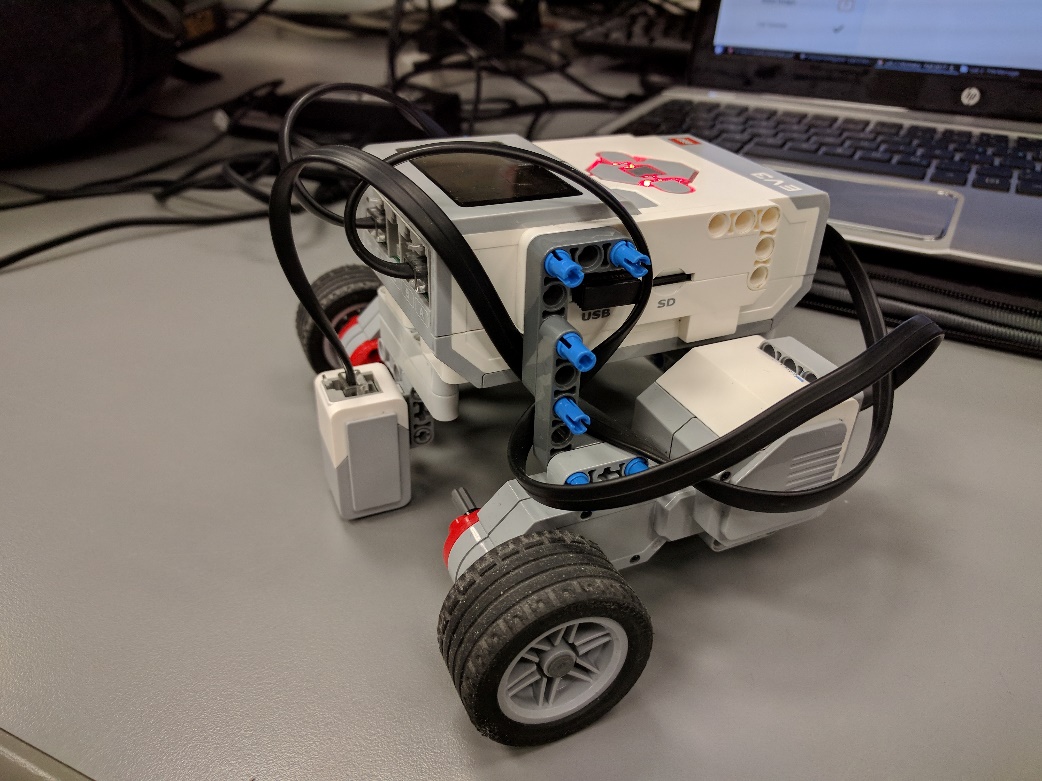
* Design Evaluation:

Software

C:\Users\Evan\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Odometry.png The odometry controller, which was on a different thread than the driver, used the following procedure to update the odometry every 25ms:

 The odometry correction controller, which was on a different thread than the driver and the odometer, made corrections to the odometer depending on the direction of the robot as it crossed a line. The correction used the following procedure:

Hardware

The robot is powered by two motors positioned directly below the brick. The robot also utilizes an EV3 color sensor in order to detect the dark lines on the board. The sensor is placed approximately at the point of rotation of the robot which is directly in the middle of the whole wheels. The sensor was placed as close to the ground as possible to try and eliminate any noise created by ambient light.

* Test Data:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | Odometer Test | | | | |  |
|  | X | Y |  |  | Error |
| 1 | -14.64 | -15.68 | -8.7 | -18.6 | 6.62 |
| 2 | -15.15 | -15.32 | -12.7 | -19.9 | 5.20 |
| 3 | -14.71 | -15.85 | -16.4 | -10.9 | 5.23 |
| 4 | -14.71 | -14.6 | -15.01 | -13.1 | 1.53 |
| 5 | -14.57 | -15.22 | -13.2 | -16.1 | 1.63 |
| 6 | -14.4 | -15.53 | -11.2 | -18.8 | 4.58 |
| 7 | -14.56 | -15.15 | -12.9 | -15.9 | 1.82 |
| 8 | -14.67 | -15.62 | -9.6 | -17.8 | 5.52 |
| 9 | -14.42 | -15.51 | -12.9 | -18.8 | 3.62 |
| 10 | -14.42 | -15.96 | -12.4 | -17.8 | 2.73 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | Odometer Correction Test | | | | |  |  |
|  | X | Y |  |  | Error |
| 1 | -20.08 | -11.9 | -16.3 | -12.1 | 3.78 |
| 2 | -21.28 | -11.68 | -17.2 | -11.8 | 4.08 |
| 3 | -18.72 | -11.97 | -16.3 | -11.3 | 2.51 |
| 4 | -19.07 | -11.43 | -15.9 | -11.3 | 3.17 |
| 5 | -17.06 | -10.95 | -13.8 | -12.5 | 3.60 |
| 6 | -18.2 | -11.86 | -15 | -12.8 | 3.33 |
| 7 | -19.01 | -11.95 | -16.4 | -12.5 | 2.67 |
| 8 | -18.57 | -11.94 | -16.7 | -12.7 | 2.02 |
| 9 | -18.06 | -10.25 | -15 | -10.5 | 3.07 |
| 10 | -16.86 | -10.46 | -14 | -11.2 | 2.95 |

* Test Analysis:

|  |  |  |
| --- | --- | --- |
|  | Odometer Test | Odometer Correction Test |
|  | -14.63 | -18.70 |
|  | 0.22 | 1.31 |
|  | -15.44 | -11.44 |
|  | 0.40 | 0.65 |
|  | 3.85 | 3.12 |
|  | 1.84 | 0.62 |

From the test data we get above, we can see that the standard deviations of X and Y for the one with correction are larger than that for the one without correction. While the standard deviation of error for the one with correction is smaller. Because with the correction the odometry readings will be closer to the actual traveled distance which varies at each trial. However, for the one without correction, the odometer reading will always tend to be around (-15,-15) due to our design. Thus, the one with correction will have a larger scattering and thus a bigger standard deviation. While for errors, as the X and Y after correction will be closer to and the errors calculated from these values will be more precise and will have a smaller standard deviation. So, the correction program can provide us a more accurate travelling distance.

Due to the logic of our design, we added one square length to the axis of corresponding travelling direction and adjust the odometer’s reading accordingly. Because the final edge is along X-axis, so our reading for X will be closer to the actual traveling distance. Thus, for the calculating the errors, we shall get a smaller value for the X-axis.

* Observations and Conclusions:

According to the data we observed from odometer with no correction the error for now is quite large, though sometimes we could get a small error but the standard deviation for it is around 3 times of that of the one with correction which means we tends to get a random result. And for it to run a longer distance the error will only get bigger and will have a larger scattering. So, for it to run 5 times of the 3-by-3 grid’s distance with no correction will be intolerable.

The odometer’s error shall grow linearly with respect to the distance. Because as our measurement of the wheel radius and track length can not be the same as the real value, the more distance it travels and the more turning it does will all accumulate the error. So the error will increase with the increasing of the total distance.

* Further Improvements:

Slipping Reduction:

One method to reduce wheel slipping using software is to reduce the speed of the motors, this would reduce the acceleration of the wheels to avoid slipping. Another method would be to use smooth acceleration in order slowly accelerate to the desired velocity, thus reducing slippage.

Correcting odometer angle:

Using two light sensors, the angle could be corrected if the two sensors are placed in parallel. This would allow us to calculate any difference in line detection between the two sensors. Then, using the current speed of the robot and the time difference between the left and right light sensors, we could calculate the robots angle and make the correction.

Using only one light sensor, we could store the corrected x and y position after passing over a line. Then, at the next line, we could calculate the change in x and y on the odometer, and use trigonometry to find the angle at which the robot was moving between the lines. The robot would still need to use the odometry to know which direction it is turning (ie. Is it at 90o or 180o?) in order to make the correction.