# LAB 2 REPORT:
# ODOMETRY

**GROUP NUMBER 62**
**MUHAMMAD ANZA KHAN 260618490**

**RYAN CHALMERS 260581055**

**Data Collected:**

We ran the robot through a 3-by-3 grid, testing both the odometer and corrected version of the odometer. Table 1 represents our findings for odometer. Table 2 represents our findings for the corrected version of odometer.

**TABLE 1: ODOMETER DATA**

| **Actual x (in cm)** | **Reported x (in cm)** | **Delta x = Actual x - Reported x (in cm)** | **Actual y (in cm)** | **Reported y (in cm)** | **Delta x = Actual x - Reported x (in cm)** |
|---|---|---|---|---|---|
| **-5.6** | **-0.05** | **-5.55** | **5.1** | **-0.13** | **5.23** |
| **-7.7** | **-0.06** | **-7.64** | **6.9** | **-0.08** | **6.98** |
| **-7.3** | **-0.05** | **-7.25** | **5.6** | **-0.08** | **5.68** |
| **-7.8** | **-0.07** | **-7.73** | **6** | **-0.28** | **6.28** |
| **-8.0** | **-0.05** | **-7.95** | **5.7** | **-0.44** | **6.14** |
| **-7.5** | **-0.08** | **-7.42** | **5.4** | **0.26** | **5.14** |
| **-6.1** | **-0.08** | **-6.02** | **4.9** | **0.21** | **4.69** |
| **-8.2** | **-0.09** | **-8.11** | **5.5** | **-0.27** | **5.77** |
| **-6.8** | **-0.10** | **-6.7** | **5.3** | **0.06** | **5.24** |
| **-7.7** | **-0.09** | **-7.61** | **5.2** | **-0.27** | **5.47** |

**Standard Deviation in x = 0.846493**
**Standard Deviation in y =0.664861**

**TABLE 2: ODOMETER CORRECTION DATA**

| Actual x (in cm) | Reported x (in cm) | Delta x = Actual x – Reported x (in cm) | Actual y (in cm) | Reported y (in cm) | Delta y = Actual y – Reported y (in cm) |
|---|---|---|---|---|---|
| **-1.0** | **-0.05** | **-0.95** | **1.2** | **0.05** | **1.15** |
| **-1.1** | **-0.06** | **-1.04** | **1.4** | **0.04** | **1.36** |
| **-1.2** | **-0.03** | **-1.17** | **1.6** | **0.06** | **1.54** |
| **-2.5** | **-0.02** | **-2.48** | **1.4** | **0.03** | **1.37** |
| **-0.9** | **-0.07** | **-0.83** | **1.9** | **0.02** | **1.88** |
| **-0.5** | **-0.03** | **-0.47** | **2.1** | **0.07** | **2.03** |
| **-0.6** | **-0.04** | **-0.56** | **1.2** | **0.02** | **1.18** |
| **-0.4** | **-0.01** | **-0.39** | **1.4** | **0.03** | **1.37** |
| **-0.3** | **-0.03** | **-0.27** | **1.3** | **0.09** | **1.21** |
| **-1.3** | **-0.09** | **-1.21** | **1** | **0.08** | **0.92** |

**Standard Deviation in x =0.635628**
**Standard Deviation in y =0.337917**

**Data Analysis:**

a) **What was the standard deviation of the results without correction (compute it for x and y separately, and provide the four (4) values in a table)? Did it decrease when correction was introduced? Explain why/why not**

Standard deviation decreased when correction was used because using correction improved the accuracy of the odometer. Since standard deviation is a measure of the deviation of data from the mean, correction increases the consistency of the odometers estimated positions over a number of trials, thus decreasing standard deviation. The completely blind reckoning odometer has a much higher chance of incurring error, which spreads data points over a larger range of values, increasing standard deviation. As expected, the absolute values the robot sensed were more with correction; when subtracted from the actual position it is more accurate to have the correction

b) **With correction, do you expect the error in the x position or the y position to be smaller? Explain.**

The error is corrected in the direction of travel. We used a standard left-handed co-ordinate system in which the angles increase clockwise. Our robot is aligned to the y direction. Since the robot moves in a square starting going north, then east, south, and finally west, it corrects y, x, y, and then x. The value which is corrected more recently will incur less error because it has less time to incur error. In this case it is x. Thus the error in x is expected to be less than y.

**Observations and Conclusions:**

**Is the error you observed in the odometer (without correction) tolerable for larger distances (i.e. circumnavigating the field requires a travel distance five (5) times larger than that used for this lab)? Do you expect the error to grow linearly with respect to travel distance? Explain briefly.**

The error in the dead reckoning odometer (the one without correction) is not tolerable as it grows linearly. This is because individual errors are independent from each other. Because future error is not effected by past error, the error will simply build up linearly over time. This is not suitable for large distances because without correction, the robot eventually goes astray. With the light sensor on the ground, it allows us to get the absolute position every tile. This corrects the odometer and limits the error. It's analogous to the heading gyroscope, which is quick acting but neutrally-stable, being corrected by the compass, which is slow, but accurate.

**Further Improvements:**

a) **Propose a means of, in software, reducing the slip of the robot's wheels (do not provide code).**

We could decrease the acceleration of the motors by decreasing the acceleration value in ***SquareDriver.java*** or we could add a method, again in ***SquareDriver.java*** that sets the speed, a getSpeed () method could suffice.

b) **Propose a means of, in software, correcting the angle reported by the odometer, when (do not provide code):**

i) **The robot has two light sensors**

If the robot had two light sensors placed side by side on the front or at back, it would be able to determine the angle it was facing every time it crossed a line, and thus correct it. If the robot is perpendicular to the line, then the sensors will simultaneously cut and detect the line.

However, should there be no line present at this point there will be a delay in the second sensor detecting the line. This time delay can be converted into distance by getting the tachometer value of the motors when the first light sensor makes a detection and finding its difference from the tachometer reading when the second sensor detects the same line. We can convert this angular difference into radians, and if we multiply it by the radius of the wheel, we get the distance traveled between the light sensors detecting the line. This creates a right triangle with the robots' skew and the line it is crossing. Determining the corrected angle will depend on the direction in which the robot is traveling in but the error in it will be the inverse cosine of this calculated distance divided by the distance between the light sensors.

**ii) The robot has only one light sensor**

If the robot only has one light sensor, we have no method to determine the angle it is facing as it crosses the line. However, if the robot remembers when it crosses the line and then measures the distance to the next line, (here we assume it knows the distance between the lines beforehand) it can calculate the error in the angle based on the error in the expected versus the measured distance traveled between lines which was detected by the light sensor. If the robot travels a distance larger than the actual distance between lines, and at the same time is traveling in the simple square which we used in this lab, then a right triangle could be created with the actual distance the robot expects to travel between the lines and the distance it measured. The error in the angle will be the inverse cosine of the actual distance between the lines (this could be assumed to be small, thus linear approximation could be used) divided by the measured distance traveled between light sensor detections.