Design Principles and Methods - Odometry Lab Report

Harley Wiltzer (260690006)
Juliette Regimbal (260657238)

October 6, 2016

1 Objective

To determine the accuracy of the implemented odometry system, and implement a simple correction using a light sensor.

2 Method

1. In the file `Odometer.java`, implement code that performs the task of an odometer as described in the Odometry tutorial and in class. You should only need to add member variables to and modify the `run()` method of the /texttttOdometer class.

2. Run the robot in a 3-by-3 tile square (where one tile is 30.48 cm in linear dimension) using the provided code and tweak the `leftRadius`, `rightRadius`, and `width` parameters passed to the `SquareDriver.drive()` method in `Lab2.java` until the robot returns (approximately) to its starting position. If your left and right wheel motors are not connected to motor ports A and B respectively, you may need to also modify those parameters. The call to `SquareDriver.drive()` is shown below.

   ```
   SquareDriver.drive(leftMotor, rightMotor, leftRadius, rightRadius, width);
   ```

3. Now, run the robot in a 3-by-3 tile square ten (10) times and measure the signed x and ydistances from the actual location of the robot on the field (recalling that its starting position isthe origin) to that reported by its odometer. Feel free to calibrate your odometer prior to thisstep. Record these results. (Note: Your odometer is reporting the position of the point directlybetween your wheel axles).

4. In the file `OdometryCorrection.java`, implement code that, using the light sensor,detects grid lines and updates/corrects the odometer's position data as needed. Your robotshould start its motion in the center of a tile, and thus grid lines will occur at x = 15 cm, 45 cm,75 cm, ... and y = 15 cm, 45 cm, 75 cm, ... You do not need to account for grid line intersectionsin this case.

5. Repeat step (3) with your odometry correction enabled. Do not recalibrate your odometer.

6. Demonstrate to a TA your code. The TA will first 'float' your motors (i.e. the motors will be shut off in such a way that they do not provide resistance to being backdriven) and push your robot on the field to confirm that your odometer works correctly. The TA will then start your robot off-center in a tile and look for it to report its correct position after running in a 3-by-3 tile square.

3 Data

Offset from Origin - Correction Disabled

X (cm)	Y (cm)	θ (rad)
-0.42	-0.66	0.01
-0.67	-0.86	0.01
-0.84	-0.84	0.01
-0.45	-0.46	0.01
-0.64	-0.82	0.01
-0.82	-0.45	0.01
-0.69	-0.66	0.01
-0.39	-0.44	0.01
-0.84	-0.66	0.01
-0.64	-0.63	0.01

Offset from Origin - Correction Enabled

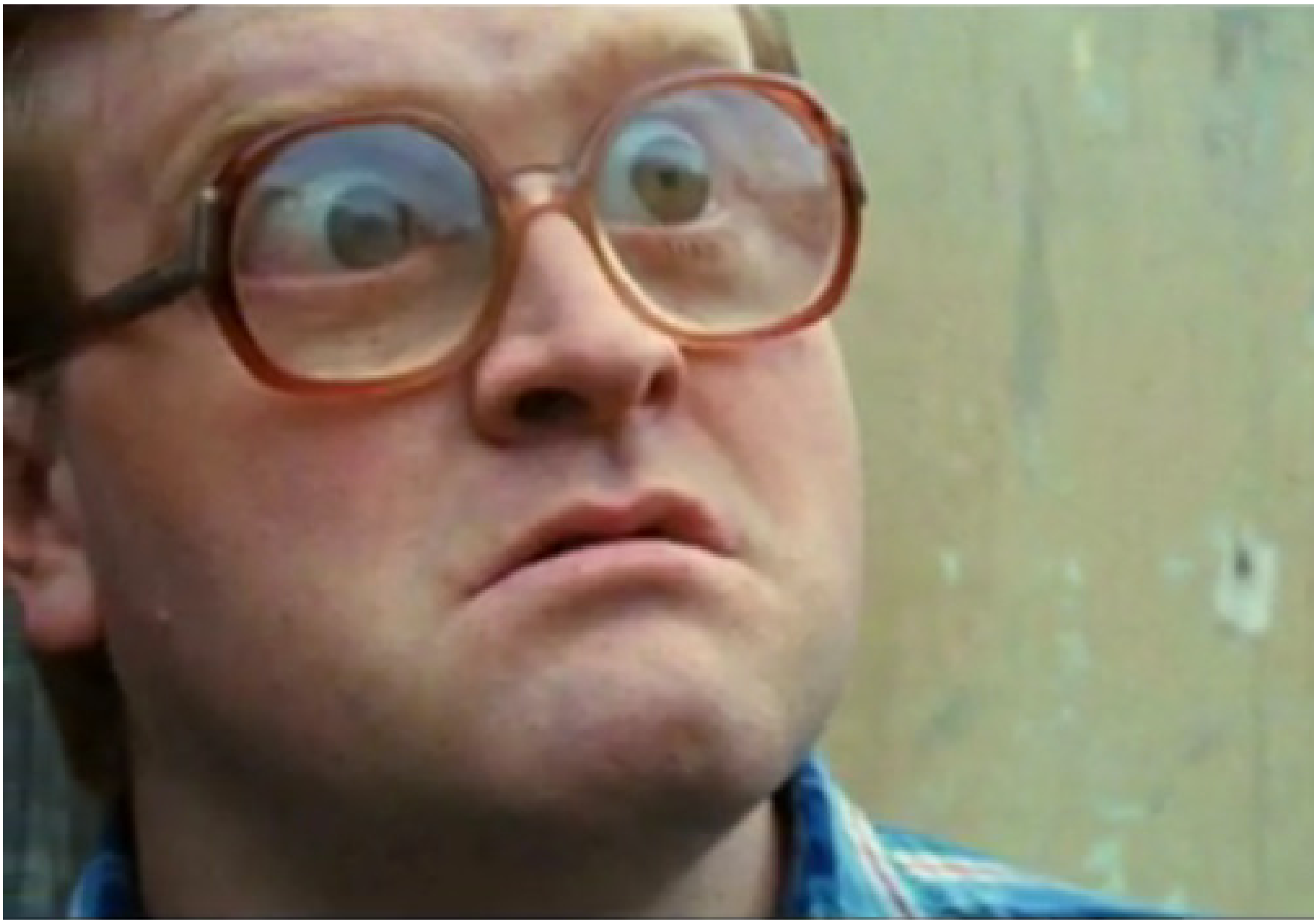
X (cm)	Y (cm)	θ (rad)
-0.04	-0.09	0.01
-0.28	-0.18	0.01
-0.18	-0.19	0.01
-0.28	-0.27	0.01
-0.31	-0.18	0.01
-0.19	-0.20	0.01
-0.27	-0.13	0.01
-0.18	-0.25	0.01
-0.08	-0.12	0.01
-0.17	-0.11	0.01

4 Data Analysis

4.1 What was the standard deviation of the results without correction? Did it decrease when the correction was introduced? Explain why/why not.

The standard deviation of the data corresponding to the robot's distance from the origin without any odometry correction was 0.17cm on the X axis and 0.16cm on the Y axis. The standard deviation did decrease significantly with the odometry correction enabled, as the standard deviation of the X position was measured to be 0.09cm, and only 0.06cm for the Y position. This makes sense, as the odometer reading was modified when the robot passed a grid line. The grid lines are a known, fixed distance away from each other, thus change in position can be accurately measured (in one spatial axis at a time) by detecting the grid lines. Therefore, with the odometry correction, no matter how much the wheels spin the robot could more accurately report its position as it crosses the gridline, reducing the effect of wheel spin on the overall odometer reading. This allows for more precise measurements.

4.2 With correction, do you expect that the error in the x position or the y position will be smaller? Explain.



5 Observations and Conclusion

6 Further Improvements

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

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

6.2.1 The robot has two light sensors.

6.2.2 The robot has only one light sensor.