Lab 1

CMPUT 412

CHRISTOPH SYDORA, BRANDON YUE

Part 2

Drawing shapes on paper was generally consistent when performed in a short time span. Certain maneuvers had poor accuracy (notably turning 90 degrees when constructing a rectangle). Since the implementation is based on a guess at how long the motors should be run for, it's natural to expect an imprecise result. The rectangle in particular had varying results at different battery levels. This imprecision also affected the circle and figure eight (which was implemented as two tangent circles) which were both "overdriven"; that is to say the robot traversed farther than 360 degrees when constructing the shapes.

Part 3

The dead reckoning robot implementation is fairly simple. It consists of the class `DeadReckoningRobot` with one primary method `public void drive(int[][] commands)`, which simply loops over all "rows" in the command matrix and sets the motors and delay appropriately. Dead reckoning calculations are performed every 10ms, and are based exactly on the tutorial given in lab:

1. Distance per tick is calculated from the robot's wheel diameter.
2. Radians per tick is calculated from the robot's wheel base diameter (distance from wheel to wheel).
3. The number of ticks the robot has traversed is obtained from the tachometer.
4. The change in distance is calculated by multiplying distPerTick and the average of the tachometer measurements.
5. Velocity is calculated by dividing the change in distance by the change in time (10ms).
6. The change in heading is calculated from the difference in the number of ticks for each wheel and radians per tick.
7. Rotational velocity is calculated by dividing rotational distance traveled by the change in time (10ms).
8. The parameters x, y and theta are updated appropriately.

In addition to dead reckoning calculations, the robot obtains measurements from a gyroscope (used to explore part 4) for comparison.

Part 4

Measuring Distance

1. Compare dead reckoning calculations with physical measurements.
2. Compare dead reckoning calculations with input from a distance sensor (ultrasonic).

The first method is obviously easier than the second, though tedious. The second provides a simple solution for determining distance, though it requires a more controlled environment. Particularly with the ultrasonic sensor (included in the EV3 kit), there must be some surface that the sensor can reflect sound off of, and the measurement is subject to extra error when reflecting of certain surfaces like corners. On top of that, the robot cannot rotate at all (if measuring off of a flat surface) else its measurements will be incorrect.

Measuring Rotation

1. Compare dead reckoning calculations with physical measurements.
2. Compare dead reckoning calculations with measurements from an electronic gyroscope.

The first method is done simply using a protractor. The second uses the gyroscope included in the EV3 kit. Unlike when measuring distance, the gyroscope can operate in “any” external environment (since it measures angle from its internal frame of reference). The measurements themselves are also reasonably accurate.