**CSCI 3302 Robotics Lab 2 Report**

1. Darrien Lee, Daniel Kim
2. In the robot.step(TIME\_STEP) statement, the code runs a loop to set the velocity of the left and right motors (move forward), checks the ground sensors to adjust the left and right motor speeds (to turn clockwise or counterclockwise) and generally to move along the given line of the state space. We also implemented loop closure in this loop to make sure that the error of our pose updates (x, y, theta) do not grow out of hand every time the robot reaches the start line.
3. If you don’t properly measure the elapsed time since the last odometry update, the calculation for the theta will be incorrect which results in a different pose\_x and pose\_y position estimations.
4. The e-puck’s average speed was between 0.1202 m/s to 0.125 m/s when covering the 10 cm distance, with an average time of 0.832 seconds to cover this 10 cm distance.
5. The e-puck’s pose should ideally show (0, 0, 0) as the pose\_x, pose\_y, and pose\_theta respectively when crossing over the starting line.
6. We implemented loop closure in our controller by setting a loop\_closure\_counter variable to 0, and whenever all three ground sensors picked up readings lower than the threshold over a certain period of time (0.1 seconds), it would reset the pose to (0, 0, 0), and reset the counter variable back to 0 as well.
7. We spent about 2 hours outside of class programming this lab which roughly estimates to 5 to 6 hours including labs times allotted to us.
8. Our implementation does work as expected, as the error of the pose\_x, pose\_y, and pose\_theeta is to be expected when the wheels of the e-puck is constantly using the max speed even when turning. We initially had some problems with calculating the odometry where we were overthinking the equations, but with the help of the professor, we managed to clear any misunderstandings up and finish the lab.