Introduction to Vision and Robotics
Robotics Practical: Line Follower

Dylan Angus, Matthew Martin

November 21, 2016

1 Introduction

The purpose of this practical is to learn about controlling a four-wheeled robot within a known environment. We used Lego's EV3 Python toolkit, assembling our own robot and developing all the robot code in Python. The robot is meant to accomplish three tasks:

- Follow a curved line from beginning to end
- Follow a set of broken and staggered lines, going from one line to the next
- Complete a lap of a closed circuit while circumventing an object placed in the path of the robot

2 Methods

We approached these tasks in a series of steps. First, we tried to gain familiarity with the operation of the robot by performing several tests on it to see its movement based on commands that were sent to it. Then, using this information, we developed a system of odometry and dead-reckoning. Finally, we solved the tasks sequentially, as each subsequent task built on some of the methods developed in the prior task.

2.1 Testing

We conducted several tests in order to get consistency in how the commands that were sent to the robot translate to actual distance moved in the world.

First, we ran the motors for a series of durations using `run_to_rel_pos()` keeping the `duty_cycle_sp` parameter constant at 25%. These durations were in the unit of tacho counts, which is how the rotary encoder inside the motor measures turns. We performed tests at 25% power for tacho counts of 100 to 700, incremented by 50. See Figure **??** for this data. From these tests and the slope of the trend line observed, we concluded that for forward commands we can convert from centimeters to tacho counts by performing the following calculation:

$$tachoCounts = \frac{centimeters}{4.807090465}$$

2.2 Odometry and Dead Reckoning

2.3 Tasks

2.3.1 Line following

2.3.2 Staggered line navigation

2.3.3 Obstacle avoidance

3 Results

4 Discussion

Appendix

`code`