EE 144 - Lab 4: Working with the physical robots

Week of May 1st

1 Introduction

Up to this point, we have been working with the simulated Turtlebots. In this lab, the goal is to become familiar with working with the real robots.

1.1 Lab assignments

The following are the assignments for the lab. You should ideally be able to complete the assignments in the 3-hour period of the lab session, but you may also continue your work during the following days if needed. For this lab, you will be working in **two-person teams**, and you should report your work in a team lab report, due one week after the start of your lab session. In your submission you should also include your **commented** code, as described in the instructions.

Assignment 1: Waypoint navigation

You are provided with an implementation of waypoint navigation, which will command the robot to move continuously on a square (see Lab 2 for a description). To run this code with the real robot, please follow the instructions for setting up the robot, posted on iLearn.

- Try different values of the control gain k_p and the distance threshold d_{thresh}, and observe the behavior of the robot when it tries to move through the waypoints. Select values that give good performance, and compare these values to the values you used in the simulator in Lab 2. If they are different, what may explain the difference?
- After choosing values for the parameters that will allow the robot to closely follow the desired square trajectory, let the robot complete the square several times. Compare the trajectory plot that you see in Matlab with the actual trajectory of the robot in the world. What do you observe? What explains the difference?

Assignment 2: Color-based tracking

In this assignment, your goal is to implement color-based tracking, as in Assignment 1 of Lab 3. We will here use colored cylinders, instead of the colored balls used in the simulator.

You are provided with the file `camera_example.m` which reads and stores 20 images from the netbook's webcam, and then displays the images. Run the code, after changing the IP address as required, to make sure that you can access and use the camera on the robot. Then, implement the following:

- Find appropriate thresholds on the Hue of each pixel, so that we can detect the green cylinder in the images (similarly to Lab 3). For each image, your code should create a binary image where the pixels that are deemed to belong to the green cylinder are set to one, and everything else is set to zero. Try to find threshold values that will lead to as few errors as possible (cylinder pixels that are missed, or non-cylinder pixels that are classified as belonging to the cylinder), with different cylinder orientations and distances.

- Do the same thing using an orange cylinder. Does the detection work equally well? Try using additional thresholds on the Saturation and Value of each pixel (in addition to the thresholds on the Hue). Does this improve performance?
- Write a P-controller, similarly to Lab 3, to make the robot track the green cylinder. The robot should rotate in place, so that the cylinder appears in the middle of the image. Tune the controller gain so that the robot can successfully track the cylinder as it is moved within its field of view.

Note: for the first two parts, you need to find thresholds that will work well on a variety of images, with different orientations and distances to the cylinder. To make it easier to test different thresholds, you can use the provided code to store images with various conditions, and then test different thresholds using the stored images (instead of testing with "live" images from the robot every time). In your report, include the images you used for testing, as well as the resulting binary images (plotted using subplot as in the provided code).