Pi Camera & Computer Vision + Other Goodies

ESE 421 – 2018C – Lab Week 2

Objectives

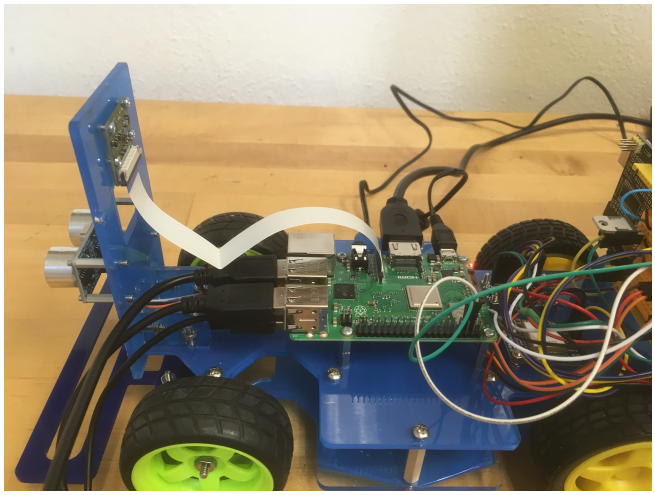
- Continue development of our estimators. In addition to heading, we need to know velocity, position. It would be great to have some data from Penn Park tests to evaluate the accuracy of our estimators over longer periods of time.
- Setup the Pi Camera and interface to openCV library to start processing images for the purpose of estimating heading and lateral offset. Ideally, we would get a system working to the point that we could use the camera heading as a replacement for GPS heading in the low-frequency part of our complimentary filter, and then feedback for heading control. Ambitious goal would be to also include lateral offset feedback to stay a fixed distance from an edge. Both of these are probably best tested on the tennis courts, rather than the roads in Penn Park, at least for a first try.
- Develop various tools that might help us test and debug the robot.

Deliverables

The results of the lab will be integrated into the next homework assignment.

Complementary Filters

This part is up to you…



Connecting the Pi Camera

With the Pi shut down, install the camera, as shown. Now boot up the Pi. Go to the Main Menu → Preferences → Raspberry Pi Configuration → Interfaces, and Enable the camera. You may need to re-start the Pi for this setting to take effect.

Then run the following python code:

```
import picamera
camera = picamera.PiCamera()
camera.capture('testPiCamera.jpg')
```

You can double-click on the file in the Pi Explorer window to view the photo.

Installing & openCV

Open a Linux shell and enter the following commands:

```
sudo apt-get update
sudo apt-get install python-opencv
sudo apt-get install python-scipy
```

We also need `python-numpy`, but the latest version of this is installed by default on the version of NOOBS that we put on the Pi, so we don't need to install it.

Now find a "test roadway" and place it in front of the car. With the 13cm mounting height of the camera, it only starts to "see" the ground about 20cm in front of the camera. Run `TestCV.py`, which has been provided on Canvas. You should see something like this:

This code is a just a start. We will need to modify it:

a. We need to change parameters in the various function calls to more robustly identify the edges of the "road"

b. We need to figure out how to ignore the lines that we don't really care about.

c. We need to use the math that we did in lecture to figure out the relative heading and offset from the road, using the information we have about the lines that we think represent the "road".

Accessing the Pi From Your Phone with SSH

When we are testing in the park, we don't have a monitor, keyboard, and mouse. But we might want to interact with the Pi. Here is how to do that with your cell phone. You have to do this setup before you go outside.

- Turn on Wifi Hotspot on your phone. Connect to the corresponding wireless network from the Raspberry Pi (click on the wireless logo in the menu bar). Don't use any spaces or other strange characters in the network name on your phone, because this might make the Pi angry.
- On Raspberry Pi, find the IP address using this command in a terminal window:

  ```
  hostname -I
  ```

  The IP address is 4 numbers separated by periods.
- Go to the Pi Main Menu → Preferences → Raspberry Pi Configuration → Interfaces, and Enable SSH. (You should also have the camera and I2C enabled already.)

Pi Camera & Computer Vision + Other Goodies

ESE 421 – 2018C – Lab Week 2

- Download the Terminus App. Follow the appropriate link:

```
https://www.raspberrypi.org/documentation/remote-access/ssh/ios.md
https://www.raspberrypi.org/documentation/remote-access/ssh/android.md
```

- Launch the Terminus App. You now have a shell window on your phone! You will have to think about how to set up your python code so that useful information can be displayed here, or useful commands entered here.

Using a USB Joystick with the Pi

Check out `TestJoystick.py`. Maybe you can use this to manually drive around Penn Park by passing commands from the Pi to the Arduino…