
Lab 7: Line Follower

Learning Outcomes

- Use HSV Segmentation
- Calculate Center of Mass with Moments
- Implement PID Control
- Tune PID parameters

Section 1: Requirements and Design

In this lab, you will use PID control to make Jet follow a line.

Section 2: Installing the Lab

To perform this lab, you will need to get the `lab7_line_follower` template into your catkin workspace. First ensure that your Jet has internet access by connecting it using WiFi or ethernet. Next ssh into Jet and enter the following command:

```
wget http://instructor-url/lab7_line_follower/lab7_line_follower-code.zip
```

Where the url should be replaced by the URL provided by your instructor. Now unzip the lab:

```
unzip lab7_line_follower-code.zip -d ~/catkin_ws/src/jetlabs/lab7_line_follower
```

Delete the zip file:

```
rm lab7_line_follower-code.zip
```

To build the code, use the following command when you are in `~/catkin_ws/`:

```
catkin_make --pkg lab7_line_follower && source devel/setup.sh
```

To run the system, execute the following

```
roslaunch lab7_line_follower lab7.launch
```

Section 3: Calculating the Error

Before you start the lab, rotate the camera to point towards the ground. Securely attach the camera to its new position. We will use OpenCV to detect the line; then we will calculate how far the robot has deviated from the line.

Recall from lab 3 how we tracked a ball using HSV segmentation. We will reuse that code and modify the lower and upper HSV thresholds to detect the pixels in the line. Place that code in the `imageCallback` function in `line_detector.cpp`. Publish the resulting mask using `user_image_pub`. Before continuing make sure that you have correctly segmented the line.

Now add the code from lab 3 that calculates the center of mass. For estimating the deviation of the robot from the line, we will only consider the x coordinate of the center of mass. There are two cases that we should consider: when the line is visible and when it is not. The `m00` value corresponds to the area of the segmented region. If `m00` is greater than 6000, set `error_msg.data` to be the x coordinate of the center of mass minus the image width divided by 2. When `m00` is less than 6000, then the line is not significantly present in the image; in this situation, set the `error_msg.data` to the special value 12345 to signal that the line is not present.

Use `line_error_pub` to publish the current `error_msg`.

Section 4: Implementing PID

Now open `line_pid.cpp`. In the `errorCallback`, set the velocity values to 0 when the special value (12345) is received.

When the error value is not the special value, we need to set the velocity of the robot to ensure that the robot maintains its course on the line. Let's keep the robot moving forward by setting the linear velocity to .7.

Then consider the requirements for PID - the current error, integral of error, and derivative of error. Update the `integral_error` value by adding the current error. Now set the angular velocity:

```
vel_msg.angular.z = -(Kp * cur_error + Kd * derivative_error + Ki * integral_error);
```

Finally, we the `prev_error` to the `cur_error`.

Section 5: Tuning PID Parameters

Open the `lab7.launch` file and set the `Kp` value to 0.01. Now compile and launch the lab. Set the robot on a line and assess how well the robot follows the line. Kill `lab7.launch` and adjust the parameters in the launch file slowly to achieve a smooth drive down the length of the line. Remember that the integral value can get very large, so often you can achieve good results with very small values of even 0 for the `Ki` term.

Challenge

Add another color of tape at the end of your current line. Adjust the thresholds in `line_detector.cpp` to detect both line colors. Now rerun the lab to see if the robot can handle the color variations in the line. As an additional exercise, remove the code that stops the robot when the line is not detected. Instead, have the robot maintain its previous course. Now make holes in the line and see if the robot can handle the small gaps in the line.