Comp3402 Line Following

C.B.Price November 2021

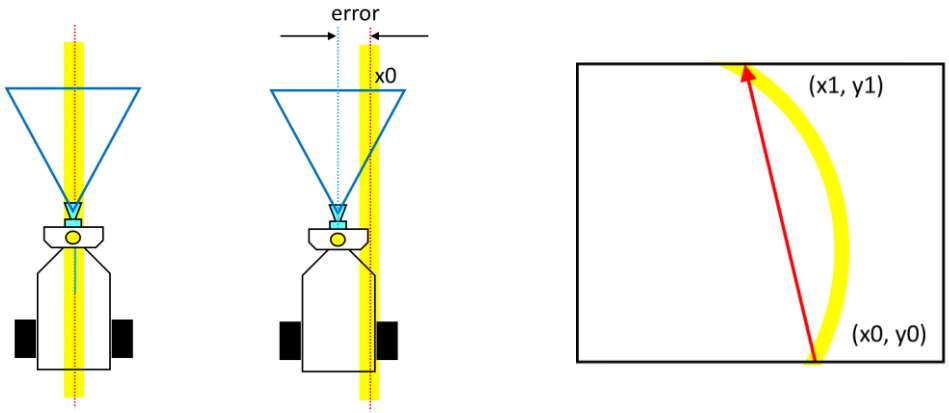
Purpose	(i) To learn about PID control (ii) To get the robot to follow a line
Files Required	Arduino Sketchbook and Octave scripts on the web-pages.
ILO Contribution	LOs 1
Send to Me	nix
Homework	Read chapter 3

Activities

---

1 The Scenario

The Pixycam will be configured in line tracking mode with a frame-width of 79 pixels. Pixymon will show you something like the diagram on the right. The API returns a vector between two points on the line. We shall use the x0 component of the tail of the vector. When this is in the centre of the frame, then the robot is centred on the line. Anything else gives us an error value shown bottom centre.



Here's the relevant parts of the API we shall need

pixy.changeProg("line");	switch mode
frameWidth = pixy.frameWidth;	
pixy.line.getAllFeatures();	get all line features
x0 = pixy.line.vectors[0].m_x0;	x-coord of vector tail
numBarCodes = pixy.line.numBarcodes;	num of codes seen
barCode = pixy.line.barcodes[0].m_code;	current code

---

2 Coding the Line Follower

Open the sketch **CBP_3402_R_Line_Follower** and add the following code.

(a) Calculate the error, using the above diagram. Remember to cast all ints to floats before doing any maths. Here's an example **myFloat = (float)myInt;**

(b) 'Normalize' the error, i.e., divide by some quantity so that the error will now lie in the range -1.0 to +1.0.

(c) Calculate the drives to the left and right servos. Remember to use the constant **fwd** drive and also the error. Use **Kp** to multiply the error.

(d) Now draw a track using the felt-pen provided. You may like to think about how to determine the sharpest bend (arc with smallest radius) the robot can follow. We shall return to this later.

---

3 Tuning the Controller

(a) First observe the behaviour of the robot for the value of **Kp** provided. Now reduce **Kp** and see how the behaviour changes. Then increase **Kp** and see how the behaviour changes.

(b) Systematically increase **Kp** until the robot starts to hunt (oscillating left and right to the line). At this point

(c) Progressively increase **Kd** until the oscillations are damped as much as possible.

---

4 Sharpest Bend

This depends on the robot geometry, the value of **fwd** and the value of **Kp**. Have a think how these factors influence the sharpest bend the robot can follow; start by considering the geometry. Test out your ideas.

---

5 Optional – Exploring PID in Octave

The script **MK_PID** lets you investigate the controller coefficients for the example given in the book chapter.
