CLUSTER INNOVATION CENTRE, DELHI UNIVERSITY

A Research Project on

ACUTE ANGLE TURN BY A LINE FOLLOWER ROBOT

Submitted to: Dr. Dorje Dawa
BY: Aditya Sharma | Vaibhav Jain [II semester]
B.Tech (IT & MI)
Cluster Innovation Centre, Delhi University

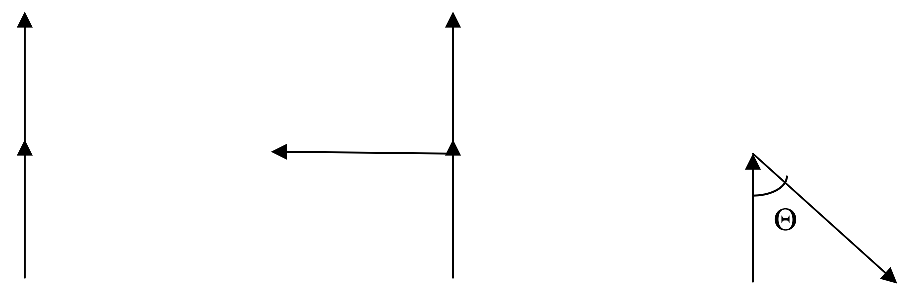
Abstract

For basic line following robots which only use White Line (IR) sensors, following a straight line is a straight forward task; but it becomes more difficult when it have turns specifically acute angled turns. In this project, we propose a strategy to address this problem. We also conduct several experiments to support the success of our algorithm.

Chapter 1: Literature Review

1.1 Introduction

The field of Line following robots has been explored for some time. For a Line follower, most important aspect is its navigation. A variety of navigation tools are used for this purpose which may vary from expensive high accuracy tools to cheap low accuracy tools [1, 4]. One of the most popular tool used for this purpose is an InfraRed (IR)-sensor. An IR-sensor detects the colour of the surface under it hence it is used to detect line. But it also has its limitations. Its main short-coming is that the range of its working space is very narrow. While following mere a straight line is easy, the task becomes more difficult when it have turns specifically acute turns (as shown in fig-1). Other methods do exist such as visual detection [5], pre-mapping of arena etc. But there has not been much work done on solving this challenge by using only IR-sensors. This research aims to investigate this challenge and to develop such an algorithm technique which will be able to allow our robot to turn acute angles with much precision.



(a) (b) (c)

Figure-1: (a) A straight path (b) A T-junction or 90 degree turn (c) An acute turn

Chapter 2: Robot Specifications

The robot we will work on throughout this paper for every experiment is a two-wheeled custom made robot suited for the purpose of this research. Along with two motors for each wheel, it has other sensors mounted on top of it. The Line Follower Robot (LFR) would consist of the following components:

2.1 Motors

The motors used in LFR are POLULU 12W DC Motor with motor driver. It is operated on 10-12V. RPM speed of the motor is controlled with Pulse Width Modulated (PWM) input pin.

2.2 Controller

A controller oversees and processes all the data from sensors and controls output power for motors. Controller used for our LFR is ARDUINO Nano ATMEGA328 micro-controller. Due to its small size and light weight, Nano provides a very good platform for making LFR since these characteristics help LFR having a small turning radius and maintaining high speed.

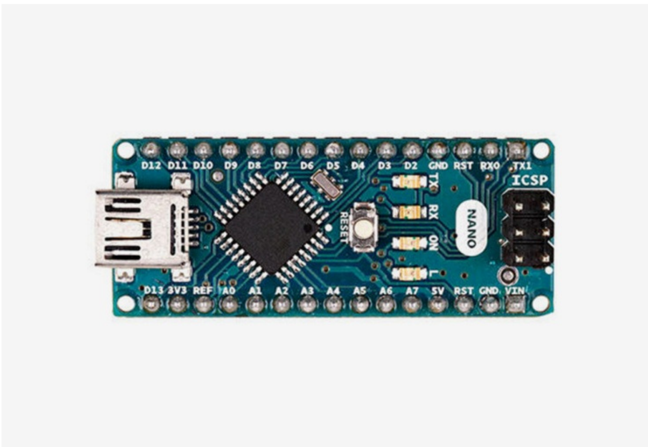


Figure-2: An Arduino NANO board (Source: official Arduino website [7])

2.3 IR-Sensor

IR-sensor used in our robot is the POLOLU QTR-8RC reflectance sensor array. These sensors are generally intended as line sensors but they can also be used as general purpose proximity or reflectance sensors.

3.1 LFR Navigation Technique

Before trying to work out the problems associated with acute turn we must acknowledge the technique used for the navigation of LFR. Overview of the algorithm we used is illustrated below in figure - 4.
At this point we would like to state that in this algorithm or any of the algorithms described further in this paper, choice 'YES' is to be considered true whenever the controller returns HIGH value.

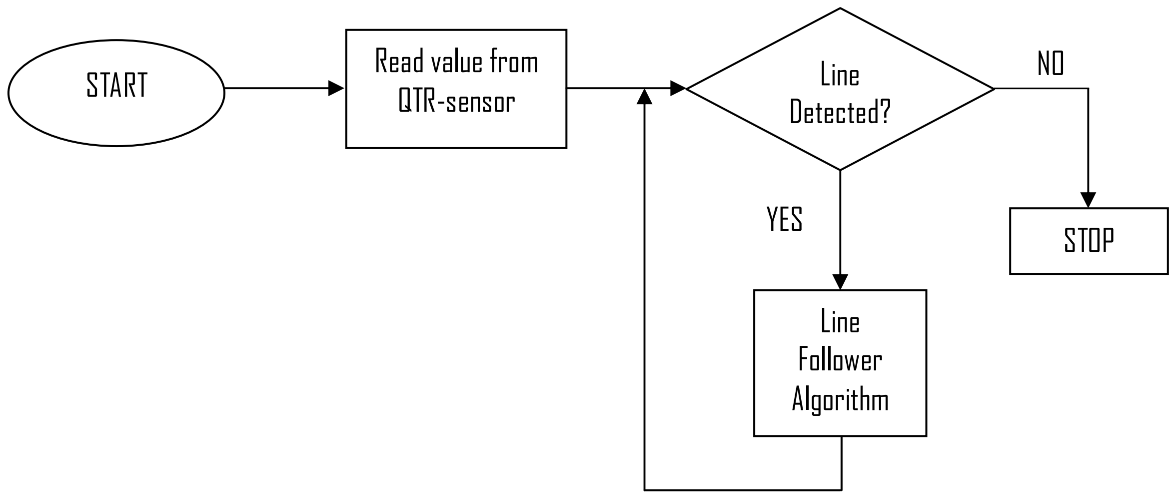


Figure - 4: Basic control flow chart of navigational strategy of LFR

Flow chart of the navigational strategy shown in figure-4 is divided into several differentsteps, for example: Line Detection, Line follower algorithm etc. Further in this section we will discuss each of these blocks, their logic and algorithm, in more details.

3.1.1 Reading Values from QTR

QTR sensor gives us two types of values which are used at different steps. First reading is given by the IR-array which helps in identifying the colour of the surface. Values are received in form of an array of 8 numbers from each IR representing the nature of the colour of the surface. Its value ranges between 0 to 1000 with 0 being highly reflecting surface (i.e. white) and 1000 being highly absorbing surface (i.e. black).
Second value gives us the average position of the centre of the line. The value is 0 if line is at extreme left to the centre and it is 7000 when line is at extreme right. 3500 is the value when line is aligned with the sensor.

3.1.2 Line Detection

A line is detected using the values returned by QTR. Following figure shows position of line for different readings.

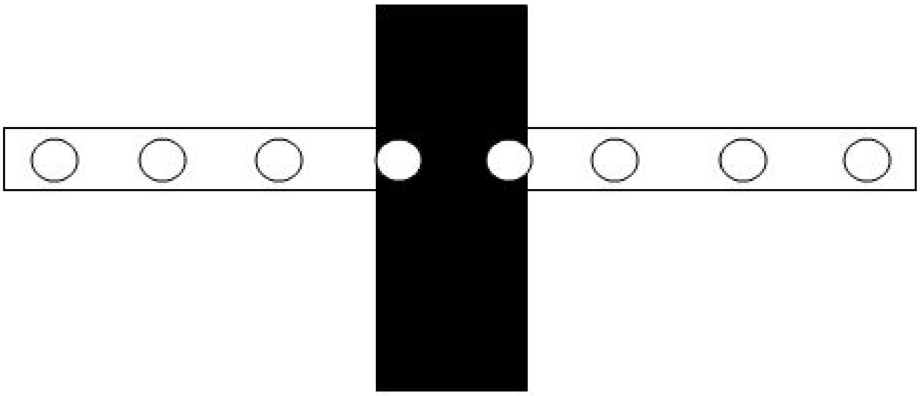


Figure - 5: Ideal position for a line on sensor array. Readings from QTR between 3000 and 4000

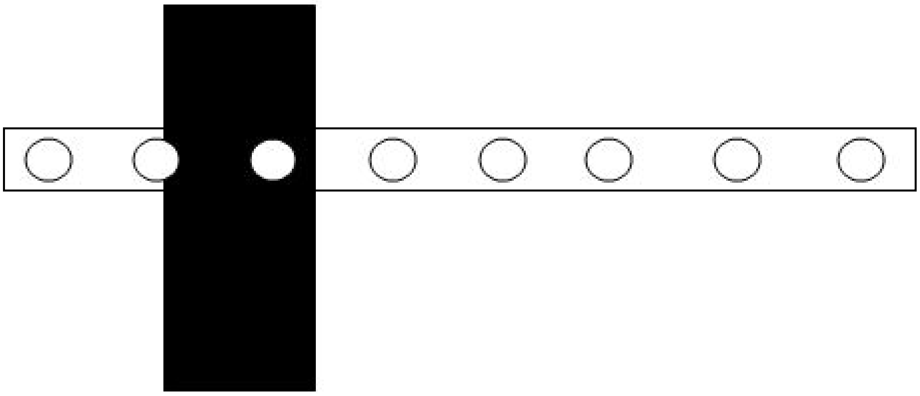


Figure - 6: Position of line for LFR to turn left. Readings from QTR between 0 and 3000

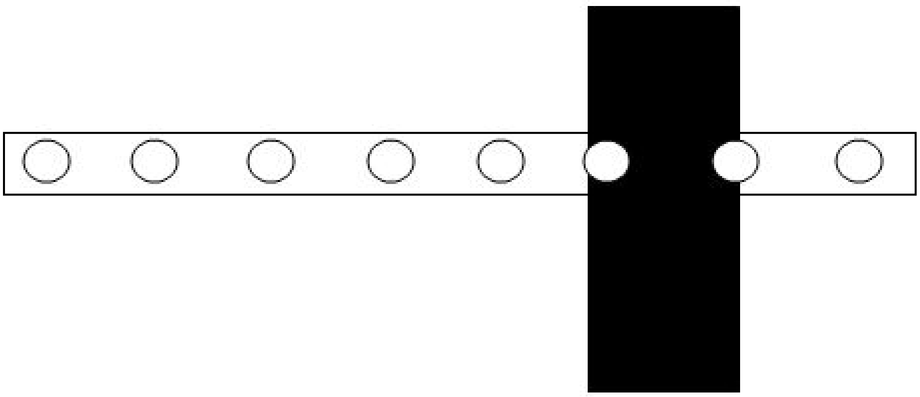


Figure - 7: Position of line for LFR to turn right. Readings from QTR between 4000 and 7000

3.1.3 Line Follower Algorithm

For a robust and responsive control system, a feedback loop control is standard and used widely for many applications. Despite of the nonlinearity of the system's equations, linear control laws are often useful for practical purpose [1].In accordance with the research presented in paper "*A class of proportional-integral sliding mode control with application to active suspension system*" for a quad-car model, we adopt the same PID control for our LFR [2].
Any feedback controller (including PID) can be divided into two parts, Controller and Observer. A controller determines the rate of change of the output based on the current error in input state and an observer finds that error between the desired state and current state. Following figure explains this feedback relation.

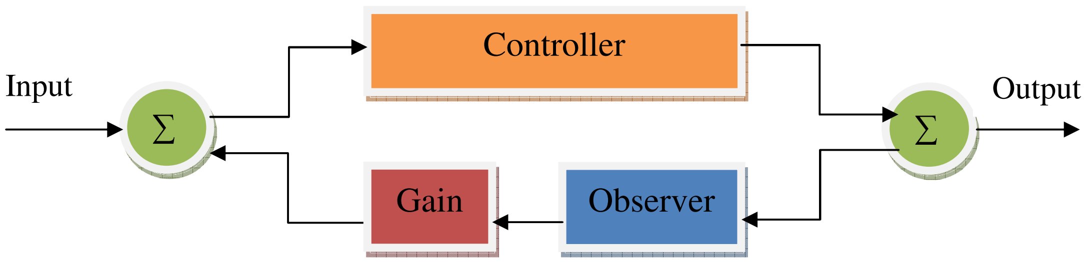


Figure - 8: Graphical representation of feedback control

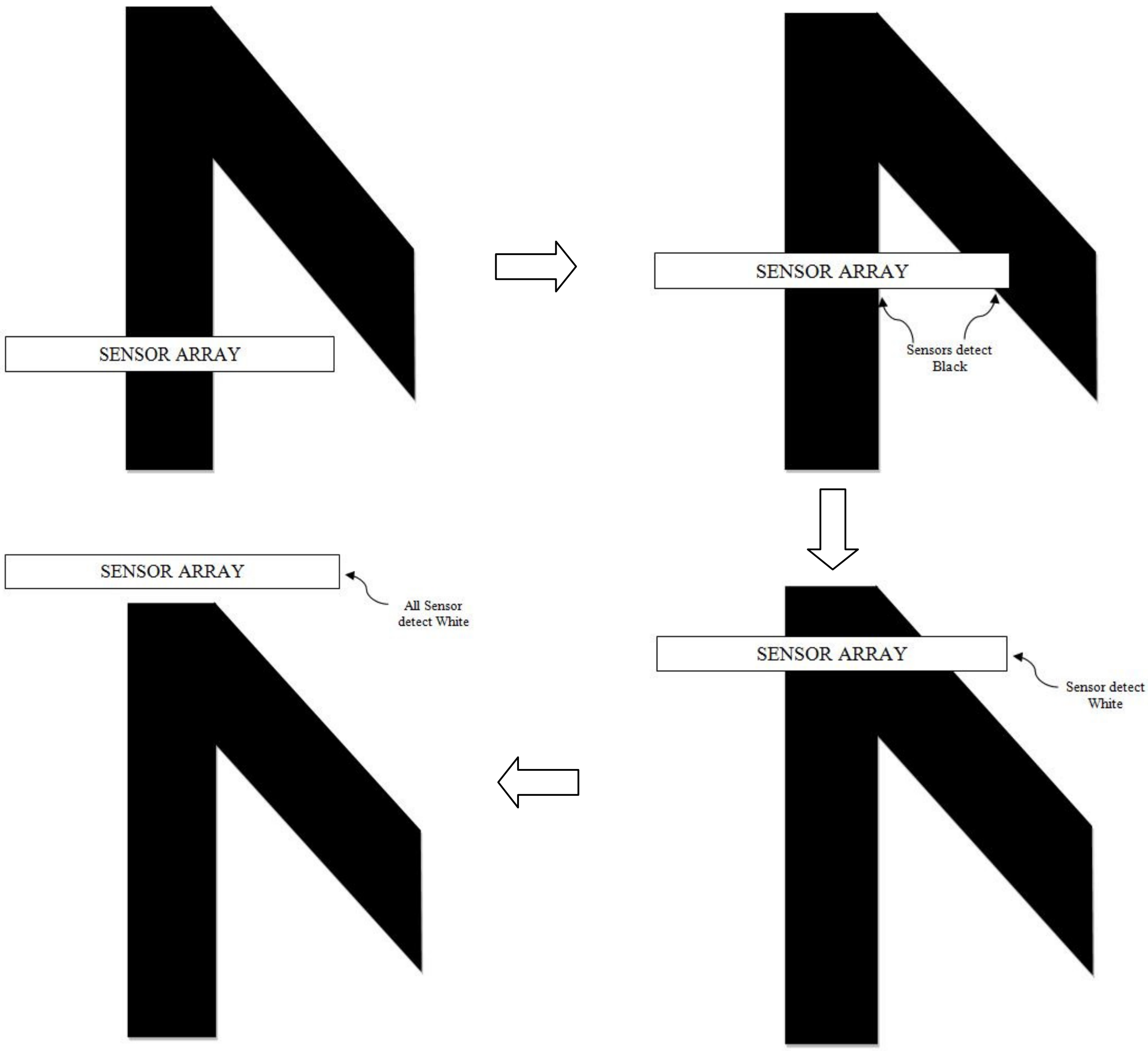
In our case, the controller controls the RPM of motors of our LFR and the observer calculates the error which is the deviation of current position of line on the sensor array from the centre.
We limit the discussion of the PID control to this point as it is a large topic in itself and this paper is not aimed for it. For a further understanding of PID control, read [8].

3.2 Acute Angle Solving Algorithm

The main problem while moving at acute turn is that the PID controller over-rides the turn and goes straight when it is not supposed to. This is a classical problem of behaviour based modelling of robotics [9]. By this approach, our LFR should change its behaviour (i.e. follow the line to something like turn left) when encountered with some trigger condition called switch. So we have to design such a switch which makes the LFR turn acute angles.

3.2.1 The Switch

When the LFR approaches an acute turn, its outermost sensors detect black line along with the middle ones but the sensors between them do not. This situation is created only before an acute turn. We used this condition to create our switch. When this condition is encountered, the LFR notes it. Then if within 20 iterations of code, no line is found, the LFR changes its behaviour from line following to turn whichever side the turn is until 'n' iterations. After that it returns its behaviour to PID control. The series of marked events is shown below.



Here computing the value of ‘n’ is critical as it will decide the responsiveness of the switch. We tested LFR for different values of ‘n’ against a 45 degree angle turn.

Degree Turn	Value of n	Status of turn
30	5	NO
	10	YES
	15	YES
	20	NO
	25	NO
45	5	NO
	10	YES
	15	YES
	20	YES
	25	NO
60	5	NO
	10	YES
	15	YES
	20	NO
	25	NO

Figure - 9: Results showing success status for different values of n

After repeated tests, we concluded that the value of n = 10 is most suitable with most robustness to noises around surface.

Chapter 4: Results and Discussions

The algorithm worked successfully barring hardware errors and sensor calibration errors. Creating the switch as mentioned above, helped in having the line follower turn acute angles via line detection by the outermost sensors.

Normally in order to make a turn at such angles we would have needed image processing i.e. addition of cameras and many more algorithms to support it but use of this algorithm reduces the complexity and uses the input from QTR sensors more efficiently. Hence, this algorithm helps in making the current line followers more precise and functional.

5. References

[1] Samson, C.Path following and time-varying feedback stabilization of awheeled mobile robot*Proceedings of the international conference on advanced robotics and computer vision,* 1992, 13, 1-1

[2] Sam, Y. M.; Osman, J. H. &Ghani, M. A.A class of proportional-integral sliding mode control with application to active suspension system
Systems & Control Letters , 2004, 51, 217 – 223

[3] Baharuddin, M. Z.; Abidin, I. Z.; Mohideen, S. S. K.; Siah, Y. K. &Chuan, J. T. T.Analysis of line sensor configuration for the advanced line follower robot
University TenagaNasional, 2005

[4] Aziz, A. R. A.Designing and Constructing an Automatic Steering Vehicle (AGV).*Dept of Electrical Engineering (2004)*

[5] Ismail, A. H.Vision-based system for line following mobile robot
Industrial Electronics & Applications, 2009

[6] Mellodge, Patricia, and Ingrid Russell.
Using the arduino platform to enhance student learning experiences
Proceedings of the 18th ACM conference on Innovation and technology in computer science education. ACM, 2013

[7] Image of Arduino NANO
As shown on official Arduino website
(*https://www.arduino.cc/en/Main/arduinoBoardNano*)

[8] Willis, M. J
Proportional-integral-derivative control
Dept. of Chemical and Process Engineering University of Newcastle (1999)

[9] Mataric, Maja J.
Behaviour-based robotics
MIT encyclopaedia of cognitive sciences (1999): 74-77.