LINE FOLLOWING BEHAVIOR FOR AN AUTONOMOUS MOBILE ROBOT USING ARTIFICIAL NEURAL NETWORKS

Manash Kumar Mandal

1203043

Department of Electrical & Electronic Engineering

Khulna University of Engineering & Technology

Contents

Abstract

In order to achieve tasks by the mobile robots, these robotic systems must have been intelligent and should decide their own action. To guarantee the autonomy and the intelligence for line following behavior, it is necessary to use the techniques of artificial intelligence like the artificial neural networks. This project report presents an approach for line following task by an autonomous mobile robot using a single layer neural network. The proposed controller is used for following any line on a plain surface with different width. This controller can also be upgraded to determine the value of k_p and k_d and make the autonomous line following a PD controller based. The results acquired from Neural Network simulation and implementation on the robot are shown and discussed.

Keywords- Feedforward Neural Network, Artificial Intelligence, Robotics, Stochastic Gradient Descent, Proportional Derivative Controller

1   Introduction

The line follower is a self operating robot that detects and follows a line that is drawn on the floor. The path consists of a black line on a white surface (or it may be reverse of that). The control system used must sense a line and maneuver the robot to stay on course, while constantly correcting the wrong moves using feedback mechanism, thus forming a simple yet effective closed loop system. The robot is designed to follow very tight curves.

In this project, the conventional control system is being replaced by artificial neural network.

1.1   Line detection mechanism

Line can be detected by either using Infra-red (IR) sensors, Light Dependent Resistor (LDR) or by a camera with line detection algorithms. Line tracking is a very important notion in the world of robotics as it give to the robot a precise, error-less and easy to implement navigation scheme.

Detecting line using IR   To detect line using IR, a threshold value must be measured. IR sensor gives different reading on different colored surface. Both values from IR on white line and IR on black line must be recorded before sensing the line.
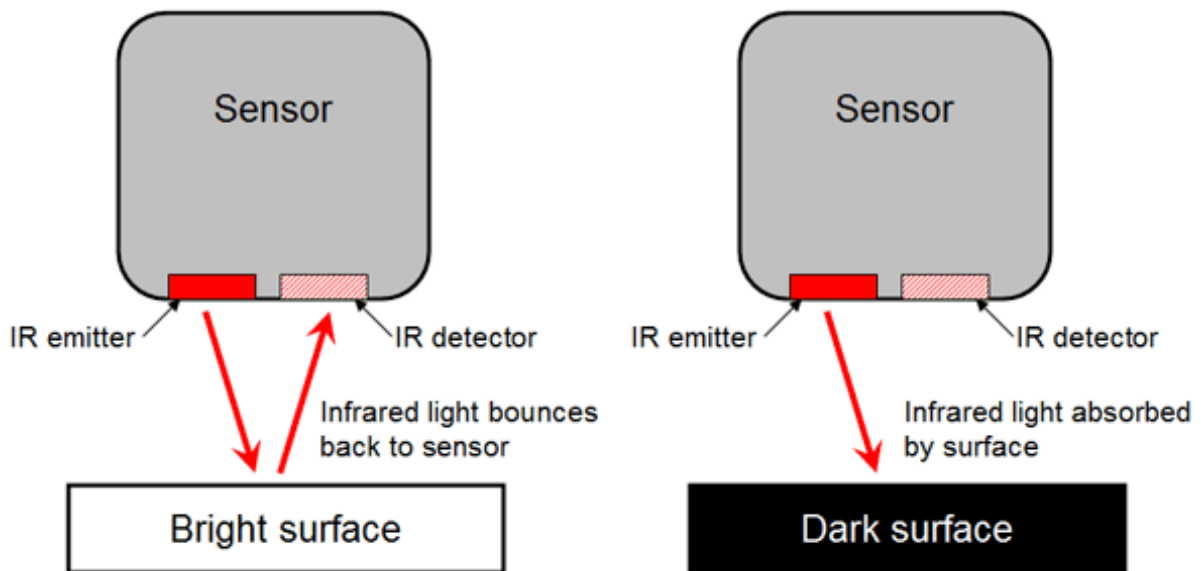
Figure 1: Detecting line using IR sensor

Detecting line using LDR To detect line using LDR same procedure from IR can be used. Since reflected line intensity depends on the reflecting surface. If the color of the surface is white, maximum light is reflected. If it is black then minimum light is reflected. Reflected light has different intensity based on the color of the surface it is being reflected from. So, nearest colors can be differentiated using LDR sensors.
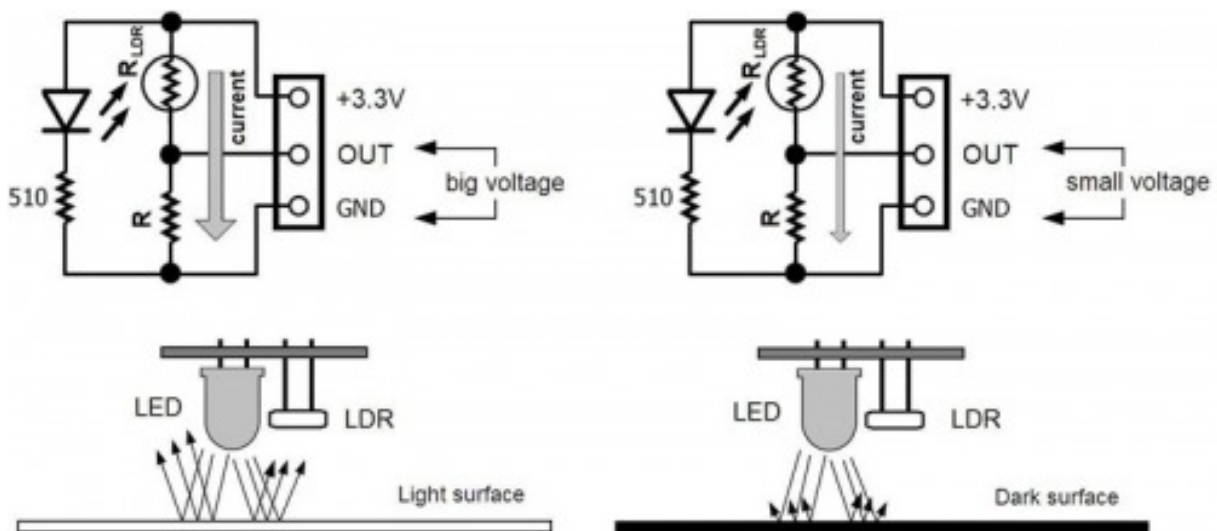

Figure 2: Detecting line using LDR sensor

1.2 Algorithm for detecting line

Algorithm 1 Line Detecting Algorithm

1: **procedure** DETECTLINE(irPin)
2:     $irPin \leftarrow$ ir receiver pin
3:     $irReading \leftarrow$ analog reading from ir receiver
4:     $threshold \leftarrow$ threshold value for differentiate betweeen white and black line
5:     $irDigitalReading \leftarrow$ converts analog into binary format
6:     $irReading \leftarrow$ reading from irPin
7:     **if** $irReading > threshold$ **then**
8:         $irDigitalReading \leftarrow 1$
9:     **else**
10:        $irDigitalReading \leftarrow 0$

1.3 Getting current position of the robot using IR / LDR Array

Current position of a robot on a line can be extracted by using IR array consisting of two or more than two IR/LDR sensors. Suppose, a robot has an IR array of 5 IR sensors. The spacing between two IR sensor is **1cm**, and the width of the line is **1.5cm**. At any time, when the robot is on the line, two values will differ from other three values meaning robot is facing straight, leaning left or leaning right. Depending on the position of the robot on the line, speed of motors can be varied to keep it on the line. This process is completely experimental and varies with the body, circuitry, battery rating, motor rating and mostly other things. The position value can be returned either in binary form such as **01100** or in weighted value such as **2500**.

1.3.1 Algorithm for weighted positional value

Algorithm 2 Position calculating algorithm

1: **procedure** CALCULATEPOSITION()
2:     $numberOfSensors \leftarrow$ number of ir/ldr sensors
3:     Let $numberOfActiveSensors \leftarrow 0$
4:     Let $digitalReading[numberOfSensors]$ be new array
5:     $weightedValues \leftarrow 0$
6:     $weight \leftarrow 1000$ $\triangleright$ Setting weighted value 1000
7:     Let $position \leftarrow -1$ $\triangleright$ Setting current position at -1
8:     **for** $index = 0$ to $numberOfSensors$ **do**
9:         $digitalReading[index] \leftarrow$ DETECTLINE($index$)
10:         **if** $digitalReading[index] \leftarrow 1$ **then**
11:             $numberOfActiveSensors \leftarrow numberOfActiveSensors + 1$
12:             $weightedValues \leftarrow weightedValues + digitalReading[index] * weight$
13:     $position \leftarrow weightedValues/numberOfActiveSensors$
14:     **return** $position$

1.4 Conventional line following mechanism

Conventional line following robots follow lines on a surface based on either predefined conditions or line patterns.

1.5 Importance of the project

Robotics technology is emerging at a rapid pace, offering new possibilities for automating tasks in many challenging applications, especially in autonomous self driving vehicles. A lot of parameters and conditions and other necessary things are needed to be considered to build a complete autonomous self driving vehicles, yet making the vehicle to learn to follow the line of the path is one of the basic building blocks to build a complete autonomous self driving vehicles. The main objective of the project is to train a network and apply it on a autonomous line following prototype which can navigate autonomously at any line consisting of any width.