Team Project 2 - Final Report

Obstacle Avoiding Robot

Computational Methods and Modeling for Engineering Applications

GENG-8030

Summer 2020

STUDENT NAME	STUDENT ID
Venkata Sai Pranav Karri	110024279
Mohith Karnam	110022312
Nishant Choudhary	110013628

Submitted to:

Professor Roozbeh Razavi Far

Submitted on 10 August 2020

Abstract

Automation Industry has been in existence for several decades and has been growing ever since. Automatic Robotic systems which can perform repetitive tasks in industry or household setting are very important. These robots can perform a task without much human interference and can save labor cost and time.

In most of the tasks, robots have to move in a certain direction and while working at a crowded site it may have obstacles in its way, which can hamper its efficiency, damage it and in a worst-case scenario it can shut down the process completely. To steer clear of this, an obstacle avoiding robot plays a very crucial role. Using this mechanism, we can program a robot to avoid any obstacle in its path.

The project aims to design a robotic system where a robot will follow a certain forward path and on detecting any kind of obstacle it will stop moving forward and will randomly start to move either in the left or right direction from its original path till the obstacle is in its path and will again continue to move in forward after getting rid of the obstacle. The project is made possible by the combination of Simulink, MATLAB, and Arduino. The implementation is initially carried out in Simulink with the help of MATLAB functions and later Arduino Kit is interfaced to fully implement the hardware.

This Report provides complete working of the project, details about the components used in its implementation, code snippets, flowchart, and references.

Table of Contents

List of Figures

1. Introduction

The main objective of this project is to create an obstacle-avoiding robotic system that is controlled by the Simulink and MATLAB functions with the help of an Arduino kit. The objective is achieved if the robot works without any human interference.

Obstacle detection is done by using an ultrasonic sensor, after the detection of the obstacle, the DC Motor, which is moving the robot in forwarding direction will stop and the servo motor will start to move the robot either in the left or right direction. When the obstacle is no longer in its path the D.C Motor will operate, and the robot will move forward until it detects another obstacle.

There are several components used in the project such as Arduino UNO kit, Ultrasonic Sensor, motor drive, etc. The Arduino is programmed to integrate all the hardware components to achieve the project objectives.

2. Objectives

- The robot will use 2 motors, D.C motor will only move forward and the servo motor will only move in the left and right direction.
- Only one motor moves at a time.
- If no obstacle is detected in front of the ultrasonic sensor, D.C motor will spin constantly and the servo motor will remain stationary. The LED will be off and LCD will display RPM of the DC Motor.
- If the ultrasonic sensor detects any obstacle, D.C motor will stop moving and the servo motor will rotate either left or right until the obstacle is present, the LED will be ON, while the LCD will display “ALERT” and d direction of movement of the servo arm.
- When the obstacle is no longer detected by the ultrasonic sensor, the D.C motor will again start moving while the servo arm will remain in a neutral position.

3. Main Components

To develop a system with these features and functionalities enabled, many components are required.

3.1 Hardware

3.1.1 Arduino UNO

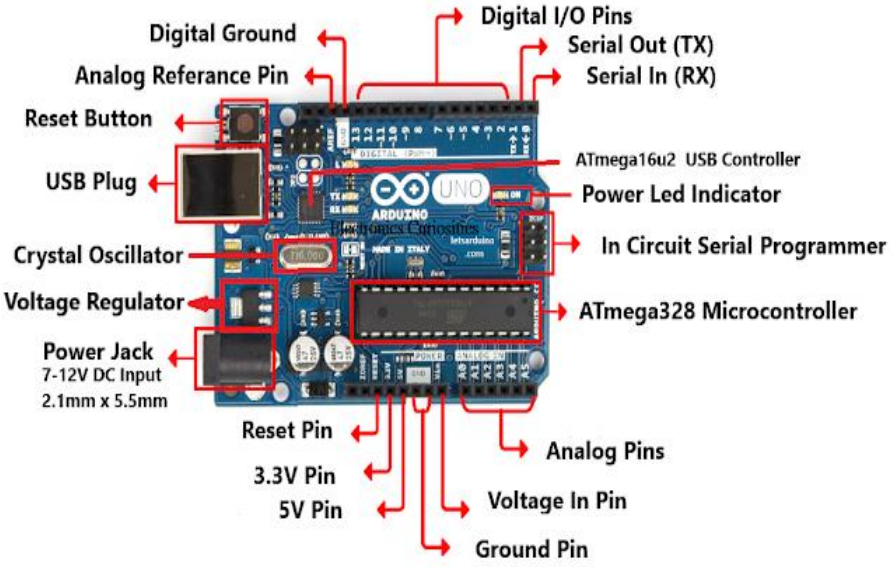


Figure 1: Arduino UNO [1]

Arduino UNO is an environment that can handle functionalities of the ATmega328P microcontroller. As shown in figure 1. It consists of various digital and analog input/output pins, a USB connection, and a power jack. This can hence be used to develop a logic that enables functionality to other components like switches, LCD, and servo motor to meet the requirements of this project.

3.1.2 LCD Display

It is used to display the current state of the robot. D.C Motor's RPM or the direction of turning of servo arm will be displayed depending on the state of the robot.

3.1.3 Servo Motor

Servo motors are used to precisely control the linear or angular position. Here it will be operated randomly either on the left of the right direction from its neutral position when the robot will encounter any obstacle, when there is no obstacle it will remain stationary.

3.1.4 Ultrasonic Sensor

Figure 2: Ultrasonic Sensor HC-SR04[2]

Ultrasonic Sensor measures the distance of an object by sending ultrasonic sound waves, the reflected waves are then converted to an electrical signal by a piezoelectric crystal. Here the sensor is used to detect the obstacles in its path.

3.1.5 D.C Motor

The DC(direct current) Motor converts Electrical Energy into Mechanical Energy. It works on the principle of Electromagnetic Induction and its direction can be changed by switching the polarity of its terminals. Here it is used to move the robot in the forward direction.

3.1.6 L293D Motor Drive and Chip

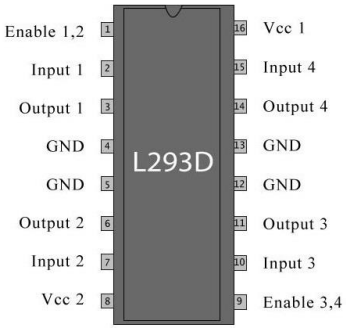


Figure 3: L293D Motor drive and chip [3]

It is a 16-pin motor drive, which can control the DC motor in both directions and acts as a current amplifier.

3.2 Software

For the successful completion of this project, Simulink and MATLAB are used.

3.2.1 Simulink

Simulink is a graphical programming environment for simulation, analyzation, and modeling of dynamical systems. Below is the simulink system that is been used in our project.

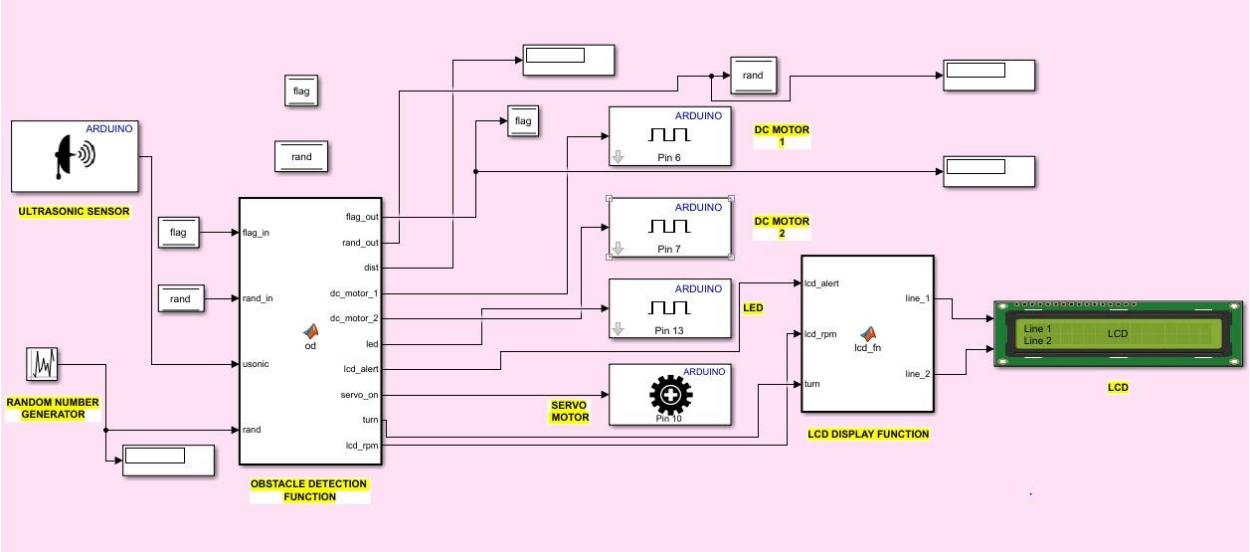


Figure 4: Simulink Model

3.2.1.1 Arduino Ultrasonic Sensor Module

Figure 5: Arduino Ultrasonic Sensor Module

This block gives a precise value of the distance of the obstacle. If the object is beyond the range of detection of the ultrasonic sensor then the output will be 0. The obtained value is then converted to distance in centimeters by dividing it by twice the speed of sound (343 m/s) The inputs for this block are

- Number of signal pins:1
- Signal pins: Pin no.9 of Arduino
- Sample time: 0.3 seconds

3.2.1.2 Uniform Random Number Generator

Figure 6: Uniform Random Number Generator

This block gives a uniform distributed random signal as its output. This generated output is repeatable for a given seed.

The input parameters are:

- Minimum: -1
- Maximum: 1 Seed: 0
- Sample time: 0.3 seconds

3.2.1.3 Data Store

i. Data Store Memory

There are two memory blocks namely “flag” and “rand” used to hold data that can be re-used in the “od” function block. All data related blocks with the same nomenclature utilize the data which is available in this memory and this is a buffer data that can be read or written as needed by read/write blocks.

Figure 7. Data store memory

ii. Data Store Read

It is used to read data and the data is being read from the Data store memory block.

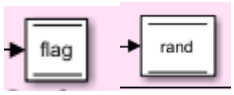


Figure 8. Data store read

iii. Data Store Write

It is used to write data to function block and the data is now being written to the data store memory.

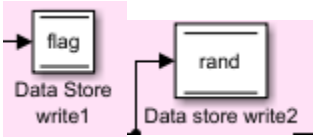


Figure 9. Data store write

3.2.1.4 Arduino Digital Output

Figure 10: Arduino Digital Output

These blocks give the logical value of a digital output pin. If the value is 0 then the pin remains at OFF state, when the value changes to 1, the pin switches to its ON state.

The Arduino pin output configurations are:

- Pin 6: Dc motor input 1
- Pin7: Dc motor input 2
- Pin13: LED

3.2.1.5 Arduino continuous servo write

Figure 11: Arduino continues-servo write

This block is responsible for rotating the servo motor arm from -90 to 90 degrees. The value zero means that the servo arm is in a neutral position. Here, Pin 10 of Arduino is allotted to the continuous servo write in the Simulink model.

3.2.1.6 LCD Display (16 x 2)

Figure 12: LCD

The LCD block shows two one dimensional arrays of type uint8 as its output in the form of line 1(top) and line 2(bottom).

The inputs fed to the Arduino pins from LCD are as below:

RS: 12, EN: 11, DB7: 2, DB6: 3, DB5: 4, DB4: 5 (where RS,EN,DB7,etc are LCD pins)

3.2.2 MATLAB Functions

3.2.2.1 Obstacle Detection Function(od)

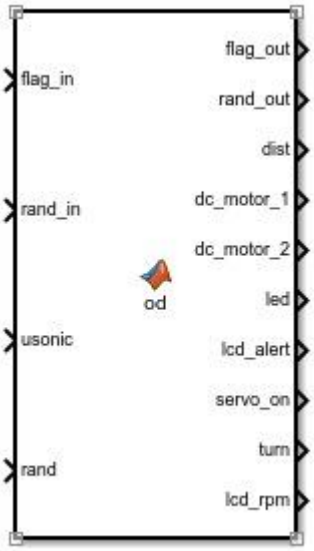


Figure 13: Obstacle Detection Function

The Obstacle detection MatLab function is one of the most important part in the Simulink model. It has 10 outputs and 4 inputs, each responsible for different operations.

Input description:

- flag_in: It reads the value from the ‘flag’ datastore read block and has a binary value which depends on the obstacle.

- rand_in: It reads the value from the ‘rand’ datastore read block and has a random value between -1 to 1.
- usonic: It reads the value from the Arduino ultrasonic sensor which depends on the distance of the obstacle from the sensor.
- rand: It comes from the uniform random number generator with a sample time of 0.3 seconds.

Output description:

- flag_out: give outputs to the ‘flag’ datastore write block
- rand_out: give outputs to the ‘rand’ datastore write block.
- dist: shows the distance of the obstacle from the ultrasonic sensor.
- dc_motor_1: binary output 1 to dc motor
- dc_motor_2: binary output 2 to dc motor.
- led: sets the LED to high state if an obstacle is encountered.
- lcd_alert: enables the string ‘ALERT’ to be displayed in the first line of LCD in case of an obstacle.
- servo_on: sets the angular value either -90 or 90 according to the generated random number.
- turn: enables the string ‘TURNING RIGHT’ or ‘TURNING LEFT’ or the RPM value to be displayed in the second line of the LCD.
- lcd_rpm: enables the string ‘RPM’ to be displayed in the first line LCD.

3.2.2.2 LCD Display Function(lcd_fn)

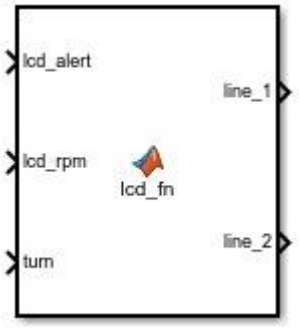


Figure 14: LCD function

The LCD function is used to display the strings/outputs in the LCD according to the given situation.

Input description:

- disp_alert: enables the string ‘ALERT’ to be displayed.
- disp_rpm: enables the string ‘RPM’ to be displayed.
- turn: enables the string ‘TURNING RIGHT’ or ‘TURNING LEFT’ or the rpm value.

Output description:

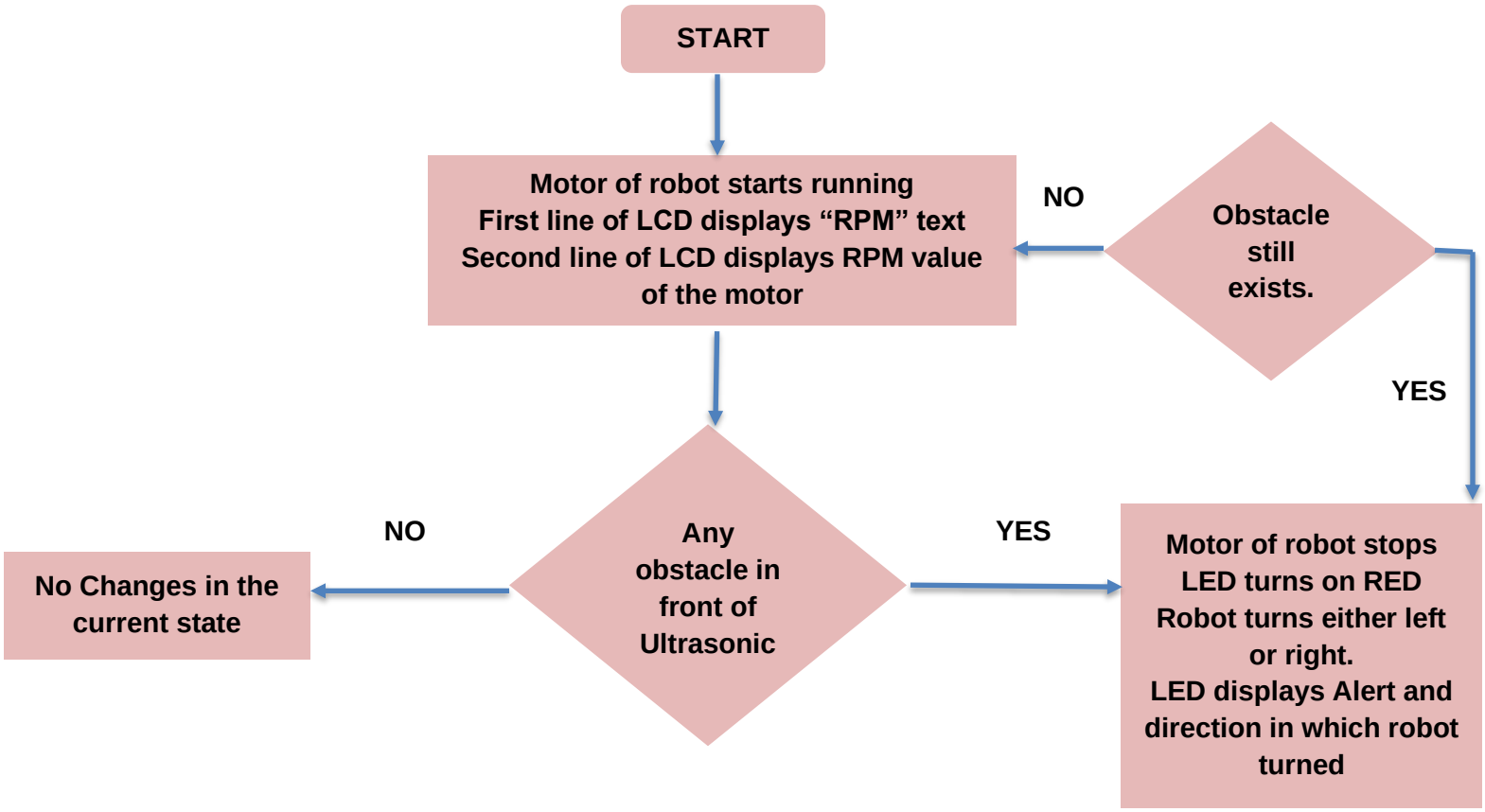
- line_1: displays the corresponding output/string in the first line of LCD.
- line_2: displays the corresponding output/string in the second line of LCD.

4. Work and Development Process

The simulation consists of two function blocks od and lcd_fn. The od function block gets input from the ultrasonic sensor. This value is initially converted to distance in the function block and thereby measures distance between obstacle and the sensor. The od function is obstacle detection function and it has conditions that provides changes on the working of robot if any obstacle is detected.

Initially the motor arm rotates and whenever an obstacle is placed in front of the sensor, od function enables lcd_alert which is helpful by lcd_fn to display alert message on the lcd. It also turns on the led light and turns off the motor by making dc_motor_1 to 0. When obstacle is removed, od goes into else block condition which is the default case where led is turned off and the motor keeps running.

5. Flowchart:



6. Test Cases

Case 1: The robot is running in the forward direction and no obstacle is encountered.

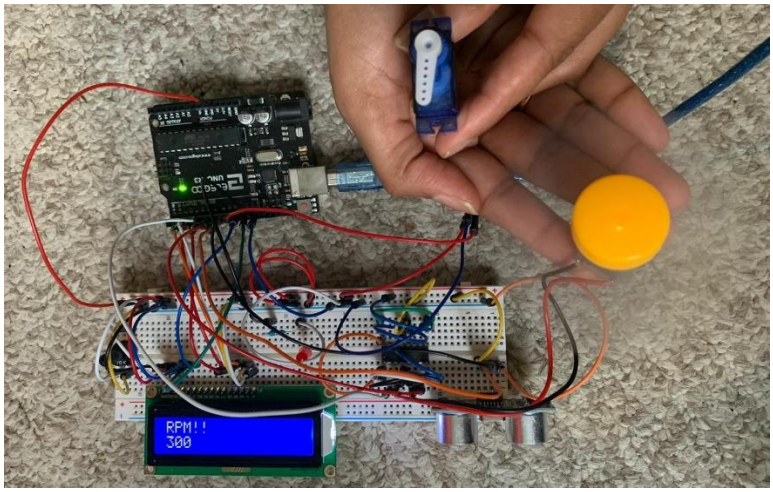


Figure 15. No obstacle

Case 2: The robot has detected an obstacle, the DC Motor stops, and the servo arm turns towards the LEFT direction.

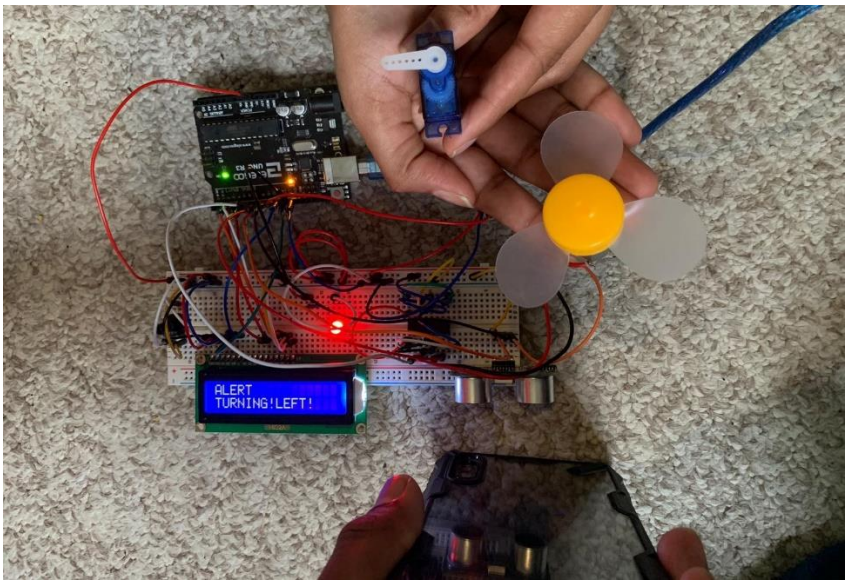


Figure 16. Obstacle detected and robot turns left

Case 3: The robot has detected an obstacle, the DC Motor stops, and the servo arm turns towards the RIGHT direction.

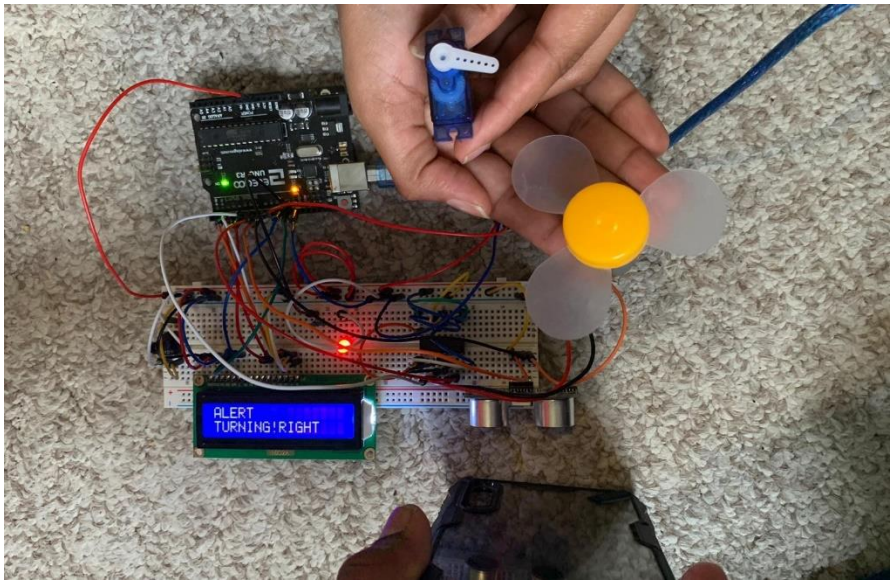


Figure 17. Obstacle detected and robot turns right

7. Code

7.1 Obstacle Detection Function(od):

```
function
[flag_out,rand_out,dist,dc_motor_1,dc_motor_2,led,lcd_alert,servo_on,turn,lcd_rpm]  =
od(flag_in,rand_in,usonic,rand)
dist=usonic/0.0686;% Conversion to distance(cm)
if(dist>0 && dist<=3)
   lcd_alert=1; % 'ALERT' to enable its functionality in lcd_fn
   lcd_rpm=0; % 'rpm' to enable its functionality in lcd_fn
   led=1;
   dc_motor_1=0; % output 1 of dc motor which later goes to DC motor driver to run/stop
the motor
   dc_motor_2=0; % output 2 of dc motor dc motor which later goes to DC motor driver to
run/stop the motor
   turn=2; % switches to Case 2 in the 'lcd_fn'
   if(flag_in==0) % initial condition check
      rand_out=rand; % rand value to the data write block
      flag_out=1; % flag is 1 when there is an obstacle
      if(rand_out>0) % checking rand value
      servo_on=[90]; % rotates servo to 90 degrees
      turn=1; % right turn
      else
         servo_on=[-90]; % rotates to -90 degrees
         turn=0; % left turn
      end
   else
      if(rand_in>0)
      servo_on=[90];  % rotates servo to 90 degrees
      turn=1; % right turn
      else
         servo_on=[-90]; %  rotates to -90 degrees
         turn=0; % left turn
      end
      flag_out=1; % flag=1 and else condition starts working
      rand_out=rand_in; % stored rand number remains unchanged until the obstacle is
removed
   end
 else
   servo_on=[0];
   flag_out=0; % flag=0 implies that intial value stored to the flag
   rand_out=rand; % rand_out is updated
```

```
    lcd_alert=0;
    led=0;
    dc_motor_1=1; % output 1 of dc motor
    dc_motor_2=0; % output 2 of dc motor
    lcd_rpm=1;
    turn=2;
end
```

7.2 LCD Display Function(lcd_fn):

```
function [line_1,line_2] = lcd_fn(lcd_alert,lcd_rpm,turn)
if(lcd_rpm==1 && lcd_alert==0) % Cases where obstacle detected or not.
    line_1=[[82],[80],[77],[33],[33]]; % "RPM!!" in ASCII
else
    line_1=[[65],[76],[69],[82],[84]]; % "WELCOME" in ASCII
end
    switch turn % switch case works for clockwise,anticlockwise motions of the servo arm and
also to display the rpm value of the dc motor
        case 0
        line_2=[[84],[85],[82],[78],[73],[78],[71],[33],[76],[69],[70],[84],[33]]; % "TURNING
LEFT" in ASCII
        case 1
        line_2=[[84],[85],[82],[78],[73],[78],[71],[33],[82],[73],[71],[72],[84]]; % "TURNING
RIGHT" in ASCII
        case 2
        line_2=[[51],[48],[48],[32],[32],[32],[32],[32],[32],[32],[32],[32],[32]]; % "300" rpm
value in ASCII
        otherwise
          line_2=[[51],[48],[48],[32],[32],[32],[32],[32],[32],[32],[32],[32],[32]]; % "300" rpm
value in ASCII
    end
end
```

8. Analysis of Results

The Obstacle Avoiding Robot project has been successfully designed and implemented with the help of MATLAB functions, Simulink, and Arduino kit, also all the 3 test cases have been successfully carried out as well.

9. Conclusion

The project is successfully implemented and has achieved all its objectives. Although it is a prototype, there is a scope of improvement in its design by using multiple sensors to increase its efficiency and further improvement can be done using Artificial Intelligence. Then this robotic system could be used in all aspects of life ranging from general labor work at industries to Automatic Vehicles etc. It will save time and money, but most importantly it can be utilized to carry out the tasks which are out of scope for a human and are often unsafe.

10.References

1. Pins Description and Hardware Components on Arduino Uno," Electronics Curiosities, 21-Jan-2020. [Online]. Available: https://www.electronicscuriosities.com/2019/08/pins-description-and-hardware_31.html

2. Reese, L., Sravya, & Kalnoskas, A. (n.d.). The working principle, applications and limitations of ultrasonic sensors. Retrieved August 11, 2020, from https://www.microcontrollertips.com/principle-applications-limitations-ultrasonic-sensors-faq/

3. Controlling a DC Motor using L293D Motor Driver Board - projectsriza. (n.d.). Retrieved August 11, 2020, from https://sites.google.com/site/projectsriza/controlling-a-dc-motor-using-l293d-motor-driver-board