University of Windsor

Final Report

Obstacle Avoiding Robot

Submitted to

Dr. Roozbeh Razavi-Far,

University of Windsor

Submitted by

Pruthvi Gabani (105159045)

Richa Singh Madnawat (105177573)

Priyanka Pradipbhai Nair (105202729)

Date:14th April 2020

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

EXECUTIVE SUMMARY:

This report presents the designing and implementation of Obstacle Avoiding Robot using MATLAB and Simulink environment. This project proposes an intelligent robot that directs itself on to a direct track if there is an obstacle in its path. It is built using Arduino UNO using the ultrasonic sensor to detect the obstacle. This sensor sends the signal to the microcontroller that controls the movement of the robot. This control is based on sensor acquired data from Arduino board sensors.

This Model uses two motors, that is, DC Motors and Servo Motors which operates simultaneously according to the detection of obstacle in the path of the robot. This invention is highly used in the field of automation to reduce the continuous human guidance in the specific task as it is highly reliable.

CHAPTER.1. INTRODUCTION

Robotics develops machines that can substitute for humans and replicate human actions. Being a branch of engineering, the applications of robotics are increasing with the advancement of technology. Robots are a combination of machines and computational intelligence. Obstacle Avoiding Robot is an autonomous robot which performs task automatically. Without continuous human guidance, autonomous robots can perform the desired task in unstructured environments.

Object Avoiding robot will automatically sense the obstacle and overcome it on its path. This robot will cover the maximum area of the provided space. Object detection is the primary requirement of this robot. The robot gets the information of the obstacle from the surrounding area through mounted sensors on the robot [2].

This obstacle avoiding robot helps to detect the obstacle or any object in the path which when applied in the automobile industry on a larger scale will reduce the accidents on the roads. It could be used in the wheelchairs to assist the physically handicapped by incorporation cutting-edge technologies. It could be used in advanced mapping devices during the exploration of unknown environments such as interplanetary exploration. Hence there is a various application of the obstacle avoiding robots in the real world which can solve numerous problems [1].

This project is built using Arduino UNO as it is a compact, comfortable and relatively cheap micro-controller. The microcontroller is used to achieve the desired operation. The microcontroller redirects the robot to move in an alternate direction depending on the input signal received [6].

CHAPTER.2. OBJECTIVE

The main aim is to build an Obstacle Avoiding Robot using the Simulink environment which involves the development and implementation of a robot that will move in the forward direction until there is no object or obstacle detected in the path. Once the obstacle is detected the robot will move sideways either in the left direction or right randomly to avoid a collision. When the object is detected the servo motor arm will change its direction either left or right. If there is no object the servo motor arm will remain in the forward direction. The entire design, implementation, and testing of the project are carried out in the hardware, that is, Arduino UNO using MATLAB and Simulink environment.

CHAPTER.3. LITERATURE REVIEW

There has been various obstacle avoiding robot in existence by different practitioners and obstacle detection has been the topic of much concern since past few decades. The most common myth of accidents being inevitable is a bane of any society. Accidents can be unplanned and could be random but are generally found to occur due to the unpredicted obstacles on the moving path.Automobile safety can be enhanced by forestalling a crash before it happens and thereby providing extra time to organize safety technologies. In previous works, a wide range of sensors and various means for sensing and avoiding obstacles for mobile robot purposes have been proposed. Based on these developed sensor systems, various methods related to this work can be grouped [6].

CHAPTER.4. REQUIREMENTS

4.1. HARDWARE REQUIREMENTS:

1. Arduino UNO - 1
2. Ultrasonic sensor - 1
3. Liquid Crystal Display (16-pin Interface) - 1
4. LED - 1
5. Servo Motor - 1
6. 3-6V DC Motor – 1
7. L293D Motor Driver Chip - 1
8. 220-ohm resistor – 1
9. 10K Ohm Potentiometer
10. Hook-up wires
11. Breadboard

4.2. SOFTWARE REQUIREMENTS:

1. MATLAB Support Package for Arduino Hardware.
2. Simulink Support Package for Arduino Hardware.
3. Arduino Engineering Kit Hardware Support.
4. Simulink library for Arduino Liquid Crystal Display.

CHAPTER 5: DEVELOPMENT PROCEDURE

5.1. HARDWARE DEVELOPMENT:

5.1.1. CONNECT LIQUID CRYSTAL DISPLAY TO ARDUINO: This process requires the following components:

- Solderless Breadboard
- Arduino UNO
- USB Cable
- Potentiometer (10K Ohm)
- LCD Screen (16*2)
- Jumper Cables
- Resistor (220 Ohm)

We will use 16 by 2-character LCD that we will be used to display symbols. Each character is off by default and is a matrix of small dots of liquid crystal. These dots make up the numbers and letters that we display on screens. The LCDs have a parallel interface to control the display, the microcontroller manipulates several interface pins at once.[5] Following are the pins in the interface:

1. Register Select (RS): This pin controls the LCD's memory where we write the data. To hold data on the screen we can use data register. The instruction register is used to look for instructions on what to do next. 2. Read/Write (R/W) pin that selects reading mode or writing mode.

2. Read or Write (R or W): This pin selects the mode for reading and writing.

3. Enable Pin (EN): Facilitate writing to the registers.

4. Data Pins(D0-D7): There are eight data pins. The states of these pins (high or low) are the bits that we are writing to a register when we write or the values we are reading when we read.

5. Display contrast pin (Vo)

6. Power supply pins (+5V and Ground)

7. LED Backlight: Pins that powers the LCD, control the display contrast, and turn on and off the LED backlight.[5]

PIN	FUNCTION	CONNECTS TO
1.	VSS	GND
2.	VDD	5V
3.	V0	Potentiometer Input
4.	RS	Arduino Digital Pin 10
5.	RW	GND
6.	EN	Arduino Digital Pin 9
7.	D0	No Connection
8.	D1	No Connection
9.	D2	No Connection

10.	D3	No Connection
11.	D4	Arduino Digital Pin 5
12.	D5	Arduino Digital Pin 4
13.	D6	Arduino Digital Pin 3
14.	D7	Arduino Digital Pin 2
15.	A	5V
16.	K	GND

Table.1: LCD-Arduino connections [5]

5.1.2. CONNECT ULTRASONIC SENSOR TO ARDUINO: Ultrasonic sensor works by sending sound waves from the transmitter, which then bounce off an object and then return to the receiver. We can determine how far away the object is by the time it takes for the sound waves to get back to the sensor.

PIN	FUNCTION	CONNECTS TO
1.	VSS	5V
2.	TRIG	Arduino Digital Pin 11
3.	ECHO	Arduino Digital Pin 12
4.	GND	GND

Table.2: Ultrasonic sensor to Arduino

5.1.3. LED CONNECTIONS: LED is used to indicate the obstacle. Whenever the obstacle is detected the LED will turn on and if there is no obstacle the LED remains off.

PIN	FUNCTION	CONNECTS TO
1.	CATHODE	Arduino Digital Pin 13
2.	ANODE	GND

Table.3: LED Connections

5.1.4. SERVO MOTOR CONNECTIONS: Servo Motor is used to move the robot in the left or right direction randomly whenever the obstacle comes in the path.

PIN	FUNCTION	CONNECTS TO
1.	YELLOW WIRE	Arduino Digital Pin 8
2.	ORANGE WIRE	5V
3.	BROWN WIRE	GND

Table.4: Servo Motor Connections

5.1.5. DC MOTOR AND L293D CONNECTIONS: DC motor is used to move the robot in a straight path, that is, forward direction.

PIN	FUNCTION	CONNECTS TO
1.	INPUT 1	Arduino Digital Pin 6
2.	INPUT 2	Arduino Digital Pin 7
3.	ENABLE 1	5V

4.	VSS	5V
5.	VS	5V
6.	OUTPUT 1	DC Motor Pin 1
7.	OUTPUT 2	DC Motor Pin 2

Table.5: DC Motor and L293D Connections

5.2. SOFTWARE DEVELOPMENT:

5.2.1. INSTALLATION OF ADD-ONS FOR ARDUINO AND MATLAB SETUP:

To build the model for adaptive cruise control in MATLAB Simulink environment we need to install several add-ons as they provide various blocks used in the model. We can install these add-ons from the home tab in MATLAB by clicking the add-ons options. These add-ons help to link hardware and software by assigning the connections of hardware to the blocks available in these packages. Below are the Add-ons used for building the adaptive cruise control:

1. MATLAB Support Package for Arduino Hardware.
2. Simulink Support Package for Arduino Hardware.
3. Arduino Engineering Kit Hardware Support.
4. Simulink library for Arduino Liquid Crystal Display.

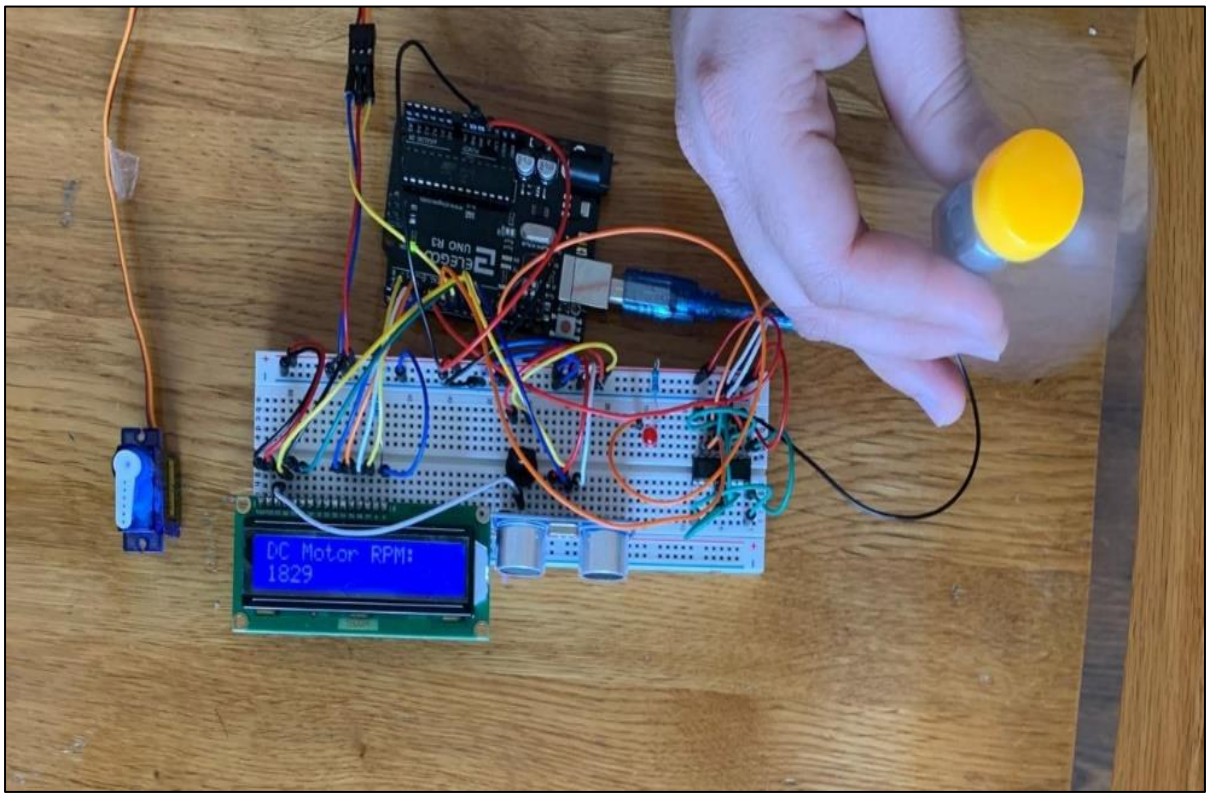

Fig.1. Hardware Setup

5.2.2. BLOCK DIAGRAM OF THE MODEL:

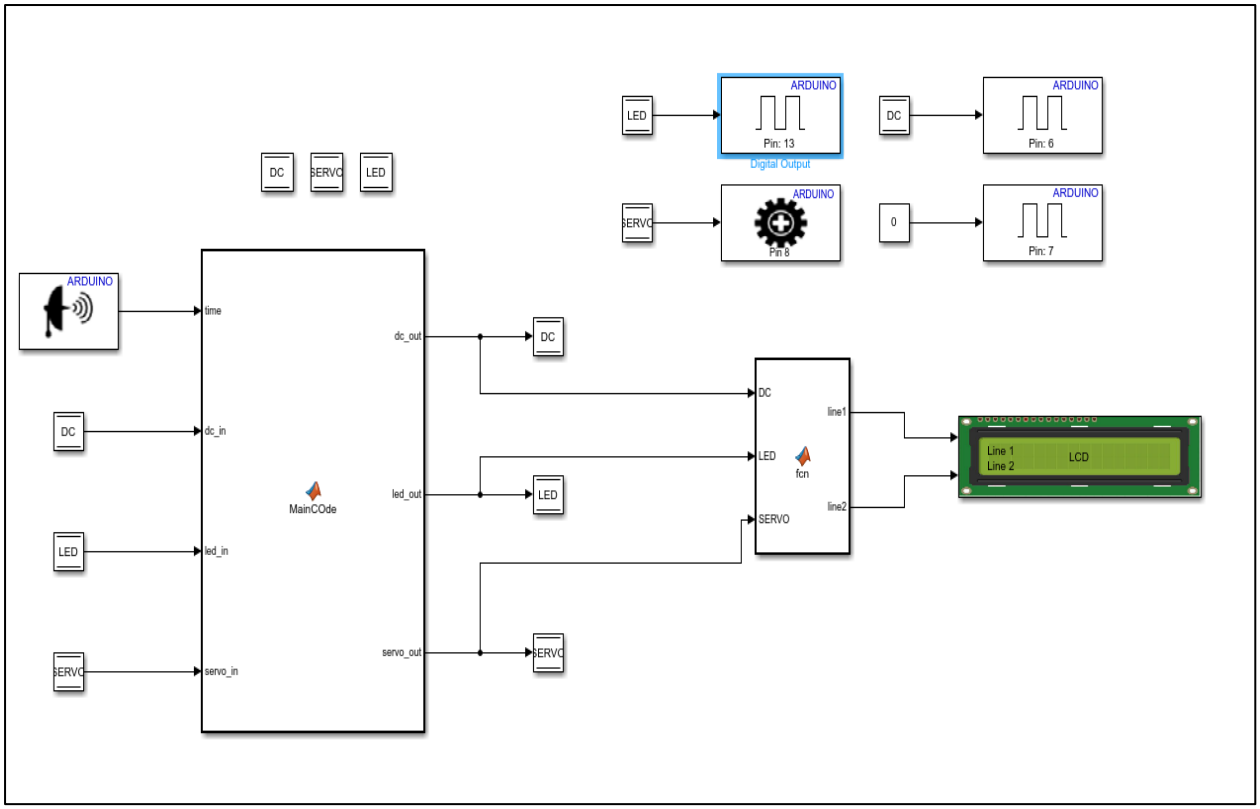

Fig.2. Block Diagram of the Model

5.2.3. BLOCKS USED IN THE SIMULINK ENVIRONMENT TO BUILD THE MODEL:

1. Ultrasonic Sensor: This is used to sense the obstacle and then the distance is calculated which will be further used to move the arm. **Distance = (Time*Speed of Sound)/2**

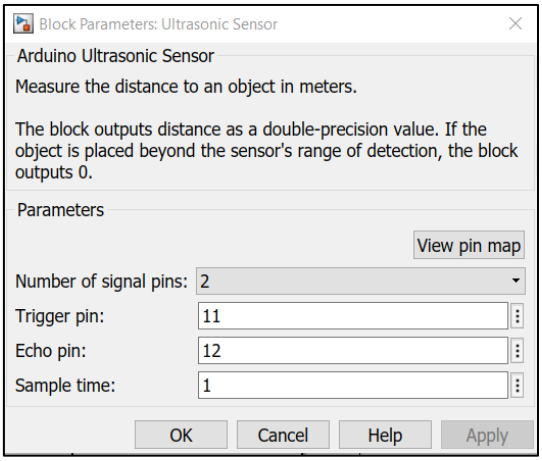

Fig.3. Ultrasonic Sensor Inputs

2. Data Store Memory: The Data Store Memory block defines and initializes a named shared data store, which is a memory region functioning by Data Store Read and Data Store Write blocks with the same datastore name. It stores the data for the further execution process. We have used three data store memory blocks each for **DC Motor, Servo Motor and LED**.

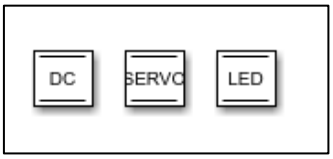

Fig.4. Data Store Memory

3. Data Store Read: The Data Store Read block copies data from the named data store to its output. So, there are respective data store read blocks for DC Motor, Servo Motor and LED where they read the value from the specific data store.

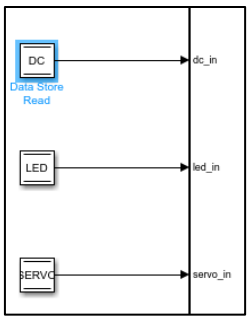

Fig.5. Data Store Read

4. Data Store Write: The Data Store Write block duplicates the value at its input to the named data store. Each write operation done by a Data Store Write block writes over the data store, swapping the prior contents. This helps to note all the present values. So again, there are three different blocks for DC Motor, Servo Motor and LED.

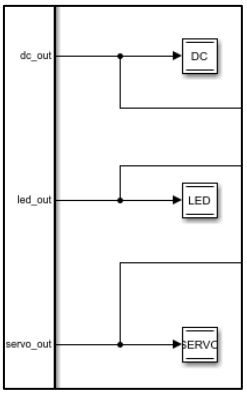

Fig.6. Data Store Write

5. MATLAB Function (SERVO AND DC MOTOR):

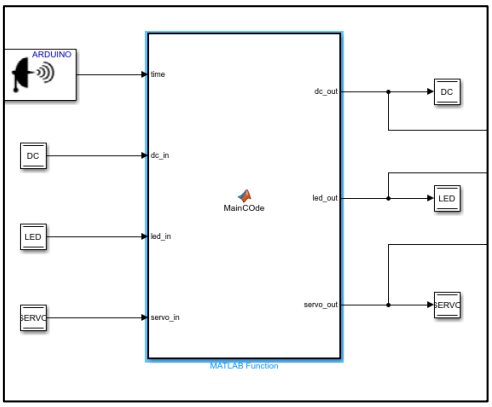

Fig.7. MATLAB Function (DC and Servo Motors)

The execution of this model is done with the help of coding in MATLAB function "MainCOde" for turning on/off DC motor and move servo motor arm along with the on/off functioning of the LED concerning the obstacles in the path. Below is the code execution explanation step by step:

```
function [dc_out,led_out,servo_out]  = MainCOde(time,dc_in,led_in, servo_in)
```

```
distance = (time*343)/2;
```

- MATLAB function is used where the output is going to tell the details about the object with the help of an ultrasonic sensor used to calculate the distance for the further process.

```
if distance > 40
servo_out = 90;
dc_out=1;
led_out = 0;
```

- Above, code is setting initial arm position to 90 degrees, if no obstacle is there which is specified by randomly taking the distance more than 40, that is, the range of the obstacle, the robot will move forward. We are setting dc_out = 1; which means it will store the value in Data Store Write and this will help in turning on DC motor. Initially DC motor is on, but LED is off, so we are setting led_out = 0 and storing it to data store write.

```
elseif distance > 0 && distance < 20
   servo_out = 180;
   dc_out = 0;
   led_out=1;
elseif distance > 20 && distance < 40
   servo_out = 0;
   dc_out = 0;
   led_out=1;
```

- For both elseif loops above, dc motor is off when the obstacle is in the range between 0 to 40 and LED is on so we are passing 1 to Digital Output, so it will light up the LED.

And since the obstacle is in the path, the servo motor arm will move 180 degrees 'LEFT' and 0 degree 'RIGHT', if the range of the obstacle is between 0 to 20 or distance, is between 20 and 40 respectively.

```
else
   servo_out = servo_in;
   dc_out= dc_in;
   led_out=led_in;
end
end
```

- If None of the condition is true then it will just set input is equal to output so nothing will happen, this condition is just to complete the 'if' loop as all the time one of the above will be true.

6. LED Data Store Read: This will take LED data read store as input and when '1' is set form the code used in MATLAB function, it will light up the LED and when '0' it will turn off the light.

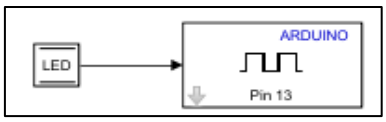

Fig.8.LED Data Store Read

7. Servo Motor Data Store Read: We have used standard servo read, to change the arm position, it will change according to the distance od the obstacle from the robot, that is, will turn left when in the between '0 to 20' and right when in the range between '20 to 40'.

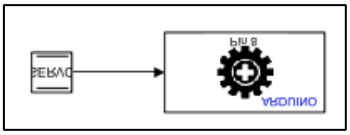

Fig.9.Servo Motor Data Store Read

8. DC Motor Data Store Read: This will take DC data to read the store as input and when '1' is set form the code used in MATLAB function then it will is turn on the DC motor making the robot move in the forward direction. And when DC is '0' then it will turn off the motor.

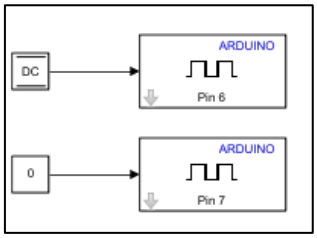

Fig.10. DC Motor Data Store Read

9. MATLAB FUNCTION (LCD DISPLAY): The top line of the LCD will display "DC Motor RPM:" and the bottom line of the LCD will display the numerical RPM of the motor.

Once the object is detected the top line of the LCD will display “ALERT”, and the bottom line of the LCD will indicate which direction the robot is moving.

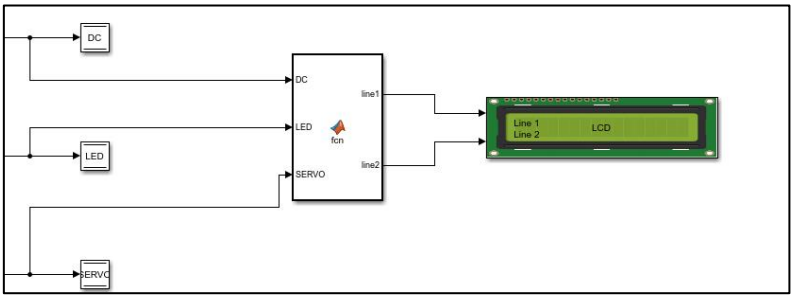

Fig.11.LCD Display MATLAB Function

Below is the complete code in MATLAB Function for the display:

function [line1, line2] = fcn(DC,LED,SERVO)

1. **Below is the code explanation for the first line of LCD**

```
if DC == 1
   line1= [[68],[67],[32],[77],[111],[116],[111],[114],[32],[82],[80],[77],[58]];
```

- DC data write store is used as input, if this condition is true then, LCD line 1 will display: “DC Motor RPM:”

```
elseif LED == 1
  line1 = [[65],[76],[69],[82],[84],[33],[32],[32],[32],[32],[32],[32],[32]];
```

- If DC is false then it will check for the above condition if LED = 1, same we have used LED data write store as input, if this is true then LCD line 1 will display: “ALERT!”

```
else
   line1 = [[32],[32],[32],[32],[32],[32],[32],[32],[32],[32],[32],[32],[32]];
end
```

- If none is true, then it will display Space which means blank.

2. **Below is the Code explanation for the second line of LCD**

```
if SERVO == 180
   line2= [[84],[85],[82],[78],[32],[76],[69],[70],[84],[32],[32]];
```

- If Servo data write store will have 180 stored in it then line 2 will display: “TURN LEFT”

```
elseif SERVO == 0
   line2= [[84],[85],[82],[78],[32],[82],[73],[71],[72],[84],[32]];
```

- If Servo data write store will have 0 stored in it then line 2 will display: “TURN RIGHT”

```
elseif DC == 1
   line2 = [[49],[56],[50],[57],[32],[32],[32],[32],[32],[32],[32]];
```

- DC data write store is used as input, if this condition is true then, LCD line 2 will display: “1829”

```
else
   line2= [[32],[32],[32],[32],[32],[32],[32],[32],[32],[32],[32]];
end
end
```

- If none is true, then it will display space.

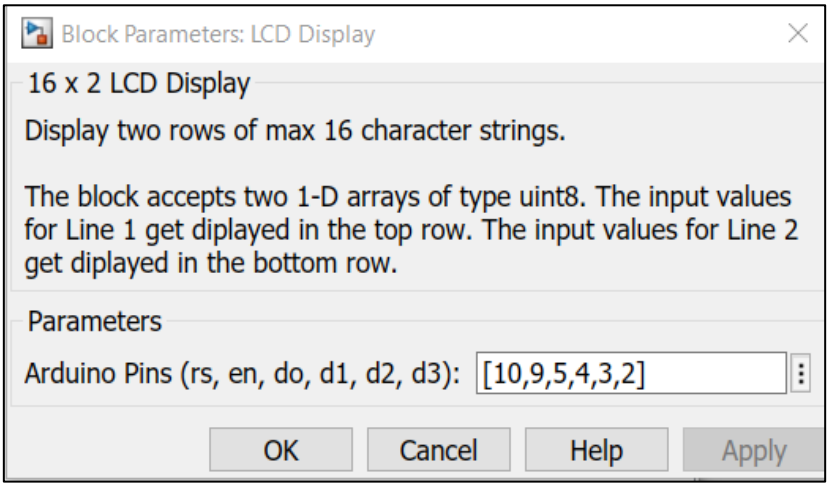

Fig.12. LCD Display Inputs

CHAPTER.6. FLOWCHART FOR THE WORKING MODEL

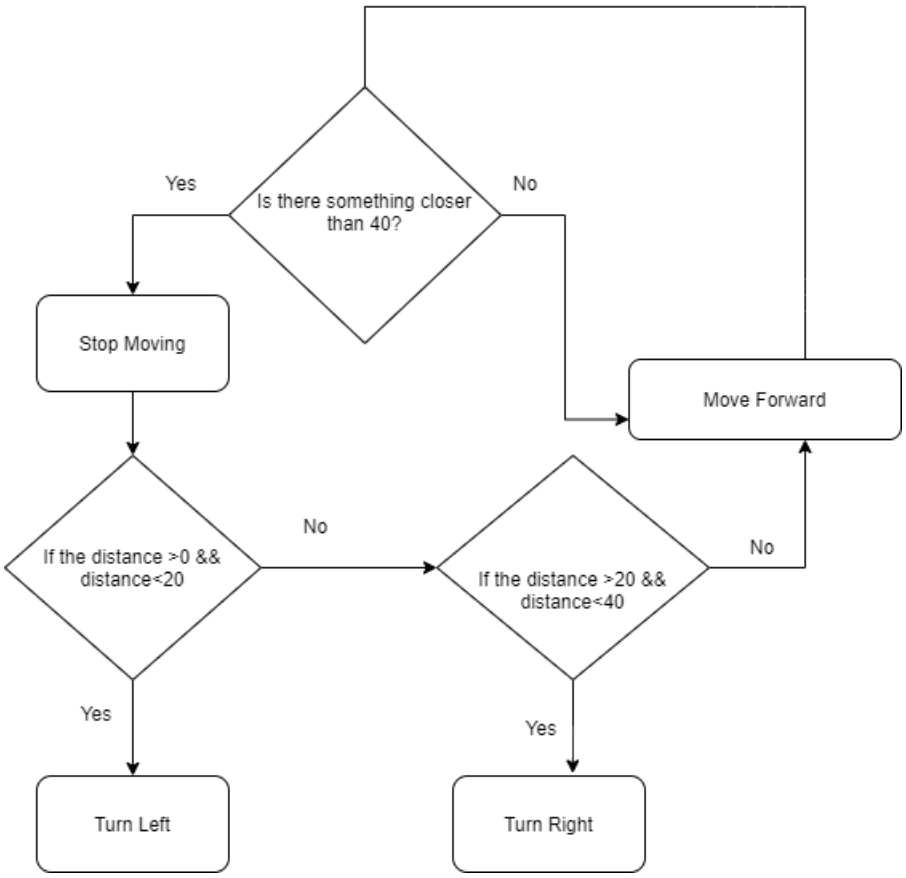

Fig.13. Flowchart

CHAPTER.7. WORKING OF THE MODEL

As per the flowchart, when the system is initialized the robot is moving in the forward direction as there is no obstacle in the path, which means the input data value of DC Motor is '1'. If the sensor detects the obstacle the system will stop, that is, DC motor gets off and turns on the servo motor and calculate the distance of the obstacle from the robot with the help of the ultrasonic sensor. If the range of the obstacle is between 0-20 the arm rotates in the left direction as the input data store for this range is '180'. If the range of the obstacle from the robot is between 20-40 the arm rotates in the right direction as the input data store for this range '0'.Once the obstacle is removed from the path, the DC motor gets on again and the servo arm gets back to its initial position, that is, in the forward direction. Only one of the motors operate at a time. When there is no obstacle DC Motor comes into action else in the presence of an obstacle servo motor starts rotating its arm respectively as per the distance between the robot and the obstacle.

CHAPTER.8. APPLICATIONS

1. Obstacle avoiding robots can be used in mobile robot navigation systems.
2. This robot can be used for evading concealed paths, especially in factories where heavy types of equipment are manufactured to avoid any injuries to the workers.
3. It can also be used in parking systems, assembling automobiles and in chemical industries.
4. They have huge scope in scientific investigation and emergency rescue, there may be places that are unsafe for people or even impossible for humans to reach directly, then we should use robots to help us gather data to about their surrounding challenging atmospheres [4].

CHAPTER.9. CONCLUSIONS

The goal of our project was to implement an intelligent autonomous robot that detects the obstacles in its path to avoid any accidents or unwanted dangerous situations which may cause any harm to the living creature of the surrounding. This will also help us navigate the process of any task that we set-up to avoid any delay or hindrance in the task [3]. Obstacle Avoiding Robot is one of the greatest inventions to protect the surrounding from any loss. The building of the model was done on Arduino UNO and its design is implemented in MATLAB Simulink environment using two motors, which are DC and Servo Motor. When the system is positioned in an unfamiliar environment with obstacles, it moved while avoiding all obstacles with significant accuracy. To enhance the working of the robot, we have many considerations for improvement. However, most of these ideas will cost more money and time as well.

REFERENCES

[1] Srijan Srivastava, Samarth Mehta, Shashank Singh, Aakriti Mittal," Obstacle Avoiding Smartcar using Arduino and Ultrasonic Sensors", Vit University, October 2017. [Accessed 10 April 2020]

[2] R.Vairavan, S.Ajith Kumar, L.Shabin Ashiff, C.Godwin Jose," Obstacle Avoidance Robotic Vehicle Using Ultrasonic Sensor, Arduino Controller", International Research Journal of Engineering and Technology (IRJET), Volume: 05,e-ISSN: 2395-0056,p-ISSN: 2395-0072, Issue: 02, Feb 2018.[Accessed 10 April 2020]

[3] S.B. Burrewar, S.P. Joshi, A.P. Shingade, A.N. Shire," Obstacle Avoidance Robot: A Review", International Advanced Research Journal in Science, Engineering and Technology Vol. 4, Issue :3 Jan.2017[Accesses 11 April 2020]

[4] Pavithra A C, Subramanya Goutham V," Obstacle Avoidance Robot Using Arduino", International Journal of Engineering Research & Technology (IJERT),2018[Accesses 11 April 2020]

[5] SM," Hello World", Arduino tutorials, Examples from libraries, Liquid Crystal, Hello World. [Accessed: 12 Apr. 2020].

[6] Utkarsh Bingewar, Shubham Thakur, Rupesh Rote "Obstacle Avoiding Car", Savitri Phule Pune University,2018[Accessed: 12 Apr. 2020].

APPENDICES

1. Code for DC and Servo Motor in MATLAB function:

```
function [dc_out,led_out,servo_out]  = MainCOde(time,dc_in,led_in, servo_in)
distance = (time*343)/2;

if distance > 40
   servo_out = 90;
   dc_out=1;
   led_out = 0;
elseif distance > 0 && distance < 15
   servo_out = 180;
   dc_out  = 0;
   led_out=1;
elseif distance > 15 && distance < 30
   servo_out = 0;
   dc_out  = 0;
   led_out=1;
else
   servo_out = servo_in;
   dc_out= dc_in;
   led_out=led_in;
end
end
```

2. Code for LCD:

```
function [line1, line2] = fcn(DC,LED,SERVO)
 if DC == 1
    line1=[[68],[67],[32],[77],[111],[116],[111],[114],[32],[82],[80],[77],[58]];
 elseif LED == 1
   line1 = [[65],[76],[69],[82],[84],[33],[32],[32],[32],[32],[32],[32],[32]];
else
   line1 = [[32],[32],[32],[32],[32],[32],[32],[32],[32],[32],[32],[32],[32]];
end
if SERVO == 180
   line2= [[84],[85],[82],[78],[32],[76],[69],[70],[84],[32],[32]];
elseif SERVO == 0
   line2= [[84],[85],[82],[78],[32],[82],[73],[71],[72],[84],[32]];
elseif DC == 1
   line2 = [[49],[56],[50],[57],[32],[32],[32],[32],[32],[32],[32]];
else
   line2= [[32],[32],[32],[32],[32],[32],[32],[32],[32],[32],[32]];
end
end
```