FINAL REPORT

OBSTACLE AVOIDING ROBOT

SUBMITTED BY:

Arulpiruthiviraj Arunthavaraja	105212573
Baranidharan Pasupathi	110007279
Rishi Anand	105195887

SUBMITTED TO:

PROFESSOR ROOZBEH RAZAVI-FAR

Executive Summary

With the shift of the world towards automation, and with every industry aiming to reduce human error, the obstacle avoiding robot is a system in high demand. With the ability to detect and avoid obstacles built in the system, this robot is the first step towards the automation of vehicles and industrial equipment. This project aims to build an obstacle detection system with the ability to avoid those obstacles in any unfamiliar environment. The goal of this project is to build a system that eliminates human error and saves time and money for both individuals and businesses. To achieve this, certain milestones were introduced into this project.

The first milestone was a detailed requirement analysis and development of the basic logic to implement this system. This was achieved by the means of a detailed flowchart of all basic operations. Then the hardware components were individually studied and tested using simple Simulink models. Finally, a single model was designed by integrating all software models and hardware components. Further, the model was tested for all given test cases and the outputs were recorded.

This report deals with the building and development of both a software and hardware model for an obstacle detection robot, which can detect and avoid obstacles automatically in any environment. Introducing this system into the real world would improve the efficiency of all automated equipment and eliminate human error entirely. This would result in a decrease in human dependency and help provide a more efficient workforce.

Table of Contents

List of Figures

1. Introduction

This project aims to develop and construct a cognitive machine which identifies its surroundings and functions according to the external environment. In today's world where self-autonomous vehicles are in high demand, industries are attempting to push to next generation vehicles [1]. The basic idea of autonomous vehicle projects is to avoid mistakes due to human error. Machines usually follow protocols designed by humans. The error rates of machine are very low when compared to humans. This notion drives industries to develop self-aware machines. To achieve these expectations, certain milestones were introduced into this project. The obstacle detection robot developed in this project could be applied to all industrial and commercial applications.

The initial stage was requirement analysis in which the prerequisites were studied to produce an initial design. After the preliminary study, a model was designed entirely in the Simulink application. Further, each of the blocks used in the software model were tested separately before being integrated together. Finally, the hardware components were successfully linked with the software blocks by certain MATLAB addons such as MATLAB support package for Arduino hardware and Arduino engineering kit support to develop a self-aware robot.

This report was put together to outline the development of an obstacle avoiding robot. Introducing this system into real time could increase road safety and reduce traffic congestion resulting in a reduction in human labour thus improving time management and cost-effectiveness.

2. Prerequisite Analysis

Prior to development of the robot, certain requirements were analyzed and researched. The main logic of the obstacle avoiding robot was formed and a flowchart was designed based upon this logic. Further the various hardware components such as the ultrasonic sensor and the DC motor were researched upon and were tested separately before the start of this project. Basic knowledge on using the various blocks in Simulink to simulate the hardware components was also studied. Further certain add-ons are required to be added to MATLAB and Simulink to successfully simulate the various hardware components.

Add-ons Used

To develop an obstacle avoiding robot, the following add-ons were introduced:

- MATLAB Support Package for Arduino Hardware
- Simulink Support Package for Arduino Hardware
- Arduino Engineering Kit Hardware Support
- Simulink Library for Arduino Liquid Crystal Display

3. Hardware and Software Used

To develop a model of an obstacle avoiding robot effectively, the following hardware components and software were used.

3.1 Components Used

To develop the hardware model of the obstacle detection robot, the following hardware components are used:

- Arduino UNO R3 Controller board – 1PC
- LCD 1602 Module – 1 PC
- Servo Motor SG90 – 1 PC
- Ultrasonic Sensor – 1 PC
- Fan Blade and 3-6V Motor – 1 PC
- L293D – 1PC
- Potentiometer 10K – 1 PC
- Breadboard – 1 PC
- USB Cable – 1 PC
- Resistor (220 ohm) – 1 PC
- LED (Green) – 1 PC
- Jumper Wires – Few

3.2 Software Used

MATLAB and Simulink were used to build a model of the obstacle detection robot.

4. Simulink Blocks Used

The following Simulink blocks are used in this model to simulate the components of an obstacle avoiding robot.

4.1 Semantic Blocks

- Data store memory block

This block is used to simulate a variable in the software Simulink. Values can be stored in memory and can be retrieved upon when required. The data store write block is used to write the value into memory and a data store read block is used to retrieve the value from memory

Figure 1: Data store memory

- Compare to constant block

This block is used to compare a value with a constant in Simulink. If the condition is true, the output from the block is 1, else the output is 0.

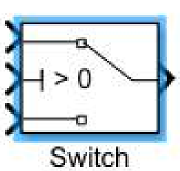
Figure 2: Compare to constant block

- Logical operator block

This block is used to simulate any logical operators. The options are AND, OR, NAND, NOR, XOR, NXOR and NOT.

Figure 3: Logical operator

- Switch block



This block is used to simulate the functions of a switch. In this model the main use of the switch block is to compare the value generated in the block to 0.

Figure 4: Switch

- Constant block

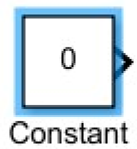


Figure 5: Constant

This block is used in Simulink to pass a constant value as input to any other blocks in the model. In this model this block is majorly used to pass constant values to the DC motor and servo motor.

- Uniform random number generator

This block is used to generate a uniformly distributed random number. The output is repeatable for a given seed value. In this block a minimum and a maximum value can be set for the output. This block is used in this project to control the motion of the servo motor.

Figure 6: Uniform random number generator

- Input block

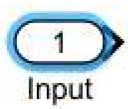


Figure 7: Input

This block signifies the input port of a subsystem or model. In this project the input block is used to simulate the input to the gate subsystem of the servo motor.

- Output block

Figure 8: Output

This block provides an output port for a subsystem or model. In this system, the output block is used to provide the output port for the gate subsystem of the servo motor.

4.2 Hardware Implementation in Simulink

In this project 5 major hardware components are simulated in Simulink: ultrasonic sensor, servo motor, LED, LCD display and the DC motor.

- Ultrasonic Sensor

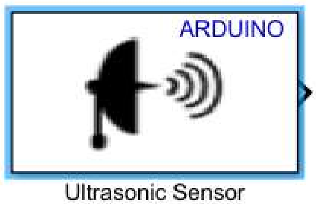


Figure 9: Ultrasonic sensor

The ultrasonic sensor is simulated using the Ultrasonic Sensor block in Simulink. This block outputs the distance from an object in meters. The trigger and echo pin numbers must be set in the block corresponding to the hardware connections.

- Servo Motor

Figure 10: Servo motor

A servo motor can be simulated in Simulink using the servo read and servo write blocks. To set the shaft position of the servo motor, a standard servo write block is used. The digital pin number must be set in the block as per the hardware connections.

- LED

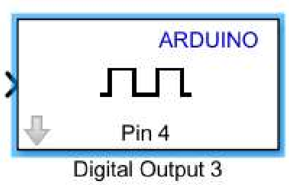


Figure 11: LED

The LED component can be simulated using a digital output block. This block is used to set the logical value of a specified digital output pin. The pin number should be set corresponding to the pin of the Arduino board to which the LED is connected.

- LCD Display

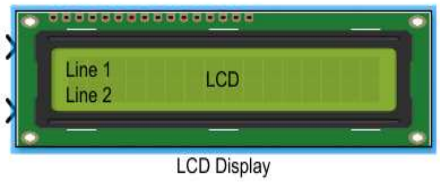


Figure 12: LCD

The LCD component can be simulated using the LCD display block in Simulink. The block displays two rows of maximum 16-character strings. The block accepts inputs only of uint8 format. In this block 6 parameters have to be set corresponding to the hardware connections – register select, data enable and the 4 data pins.

- DC Motor

Figure 13: DC motor

The DC motor can be simulated in the Simulink model using 2 digital output blocks. This block is used to set the logical value of a specified digital output pin. The pin numbers of the two input ports of the DC motor must be updated in the two digital output blocks respectively.

5. Hardware Model

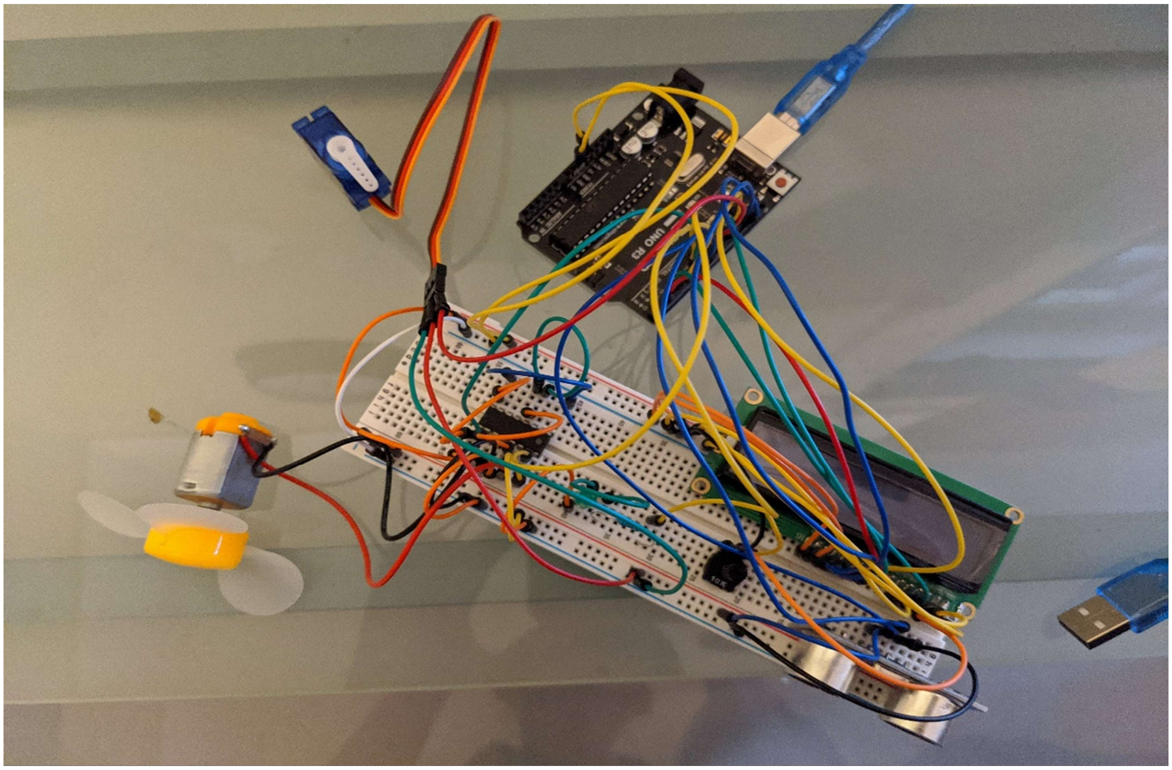


Figure 14: Hardware model of the obstacle avoiding robot

The hardware model of the obstacle avoiding robot has five major hardware modules namely, the ultrasonic sensor, LED, servo motor, LCD, DC motor and L293D driver. The pin connections of the hardware components are as follows:

- Ultrasonic sensor – The trigger pin is connected to pin 7 and echo pin to pin 8 of the Arduino board. The Vcc pin is connected to 5V and the ground pin of the sensor is connected to ground on the Arduino board.
- LED – The positive pin of the green LED is connected to pin 4 of the Arduino board through a 220ohm resistor, and the negative pin of the LED is connected to ground.
- Servo motor – The positive pin of the servo motor is connected to 5V while the negative pin is connected to ground. The third pin is connected to pin 9 of the Arduino board.
- LCD – Pin 1 and pin 2 of the LCD are connected to 5V and ground respectively and pin 15 and 16 are similarly connected to 5V and ground to facilitate backlight of the LCD. Pin 3 is connected to a potentiometer and pin 5 is connected to ground. Pins 4 and 6 are connected to the Arduino board through pins 3 and 2 respectively. Finally, four data pins of the LCD are connected to pins 10, 11, 12 and 13 of the Arduino board.
- DC motor – The positive and negative wires of the DC motor are connected to the L293D driver and pins 2 and 7 of the driver are connected to pins 5 and 6 of the Arduino board. Pins 1, 8, 16 and 9 of the L293D driver are connected to 5V and pins 4, 5, 12 and 13 are connected to ground.

6. Implementation and Working

6.1 Flowchart

Figure 15 depicts the flowchart of the logic used to design and develop an obstacle avoiding robot.

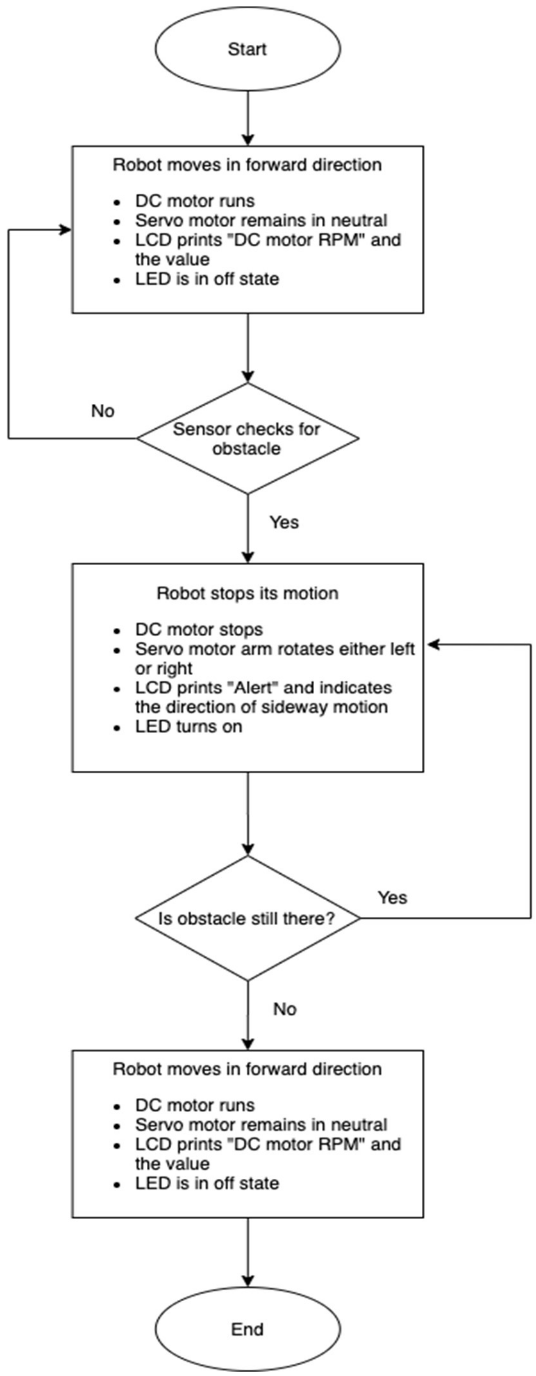


Figure 15: Flowchart of the obstacle avoiding robot

6.2 Simulink Design

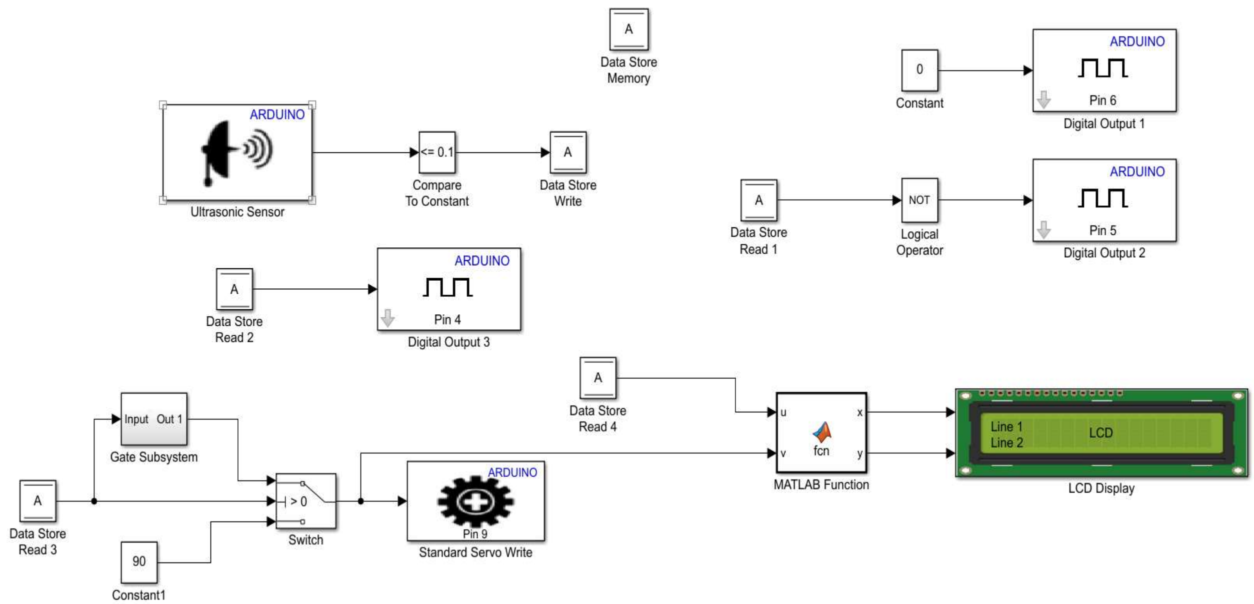


Figure 16: Simulink model of the obstacle avoiding robot

Figure 16 depicts the Simulink model of the obstacle detection robot used in this project. To simulate this model a data store memory block is used in Simulink to simulate a variable. This variable is set to 1 if the ultrasonic sensor detects an obstacle and 0 if the sensor does not detect any object using the data store write block. The value stored in the data store block is read using the data store read block. This model can be broadly classified into five modules.

- Module 1

Module 1 is involved with the ultrasonic sensor. To simulate this the ultrasonic sensor block is used in Simulink. In this block the number of signal pins is set as 2 – the trigger pin and the echo pin. In this project, the trigger and echo pins are set as pins 7 and 8 in the Arduino board respectively. The ultrasonic sensor block outputs the distance to an object as a double-precision value. That value is compared with 0.1 meters using the compare to constant block used in series with the ultrasonic sensor. This value is then set into the data store block using the data store write block.

- Module 2

Module 2 is dedicated to the DC motor. To simulate the DC motor digital output blocks are used. The two inputs to the DC motor are connected to pins 5 and 6 of the Arduino board. To pin 6, 0 is given as input using a constant block and to the pin 5, the data store read value is passed through a NOT gate and then given as input using a digital output block. This is because the DC motor should rotate when no obstacle is detected and should stop when an obstacle is detected.

- Module 3

Module 3 deals with the LED component. The function of the LED component is to turn ON when an obstacle is detected and to remain in OFF condition when no obstacle is detected by the ultrasonic sensor. This is achieved by giving input to the LED using a digital output block set to pin 4 of the Arduino board. The input to this output block is given from the data store memory using a data store write block.

- Module 4

Module 4 is involved with the servo motor. The servo motor arm is maintained at 90 degrees, which is considered as the neutral position. When an obstacle is detected, the servo arm is shifted to 180 degrees or 0 degrees randomly. This is achieved in the gate subsystem.

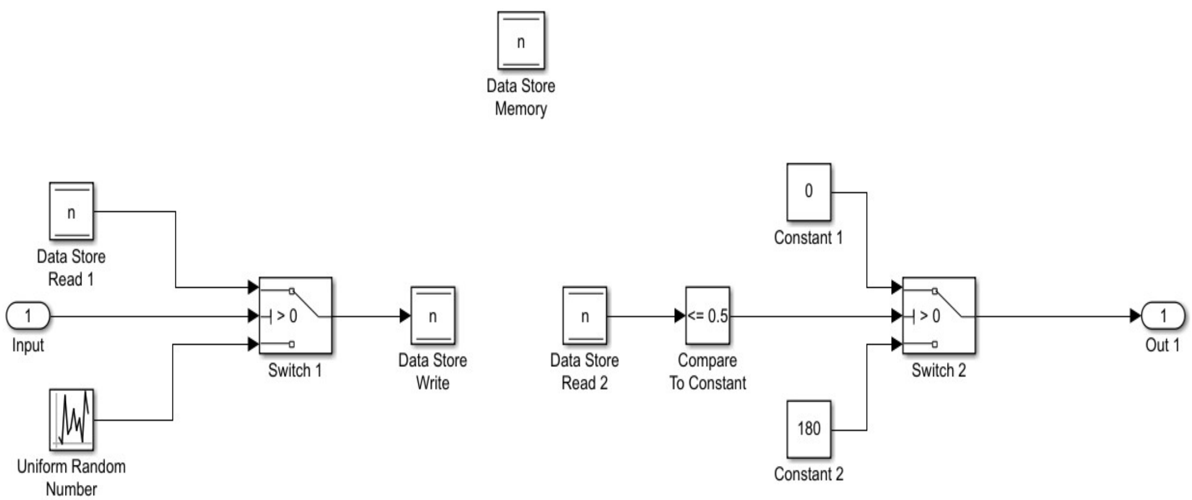


Figure 17: Gate subsystem to control servo motor arm

Figure 17 depicts the control system for the servo motor; when an obstacle is detected by the ultrasonic sensor, the servo arm is maintained at 0 degrees or 180 degrees, signifying the left or right movement of the robot. This is achieved by using a uniform random number block in Simulink. The minimum and maximum values in the uniform random number block are set as 0 and 1 respectively. This value is stored in a data store block. When an obstacle is detected, the value stored in the data store memory is compared with 0.5 using a compare to constant block. If the value is less than or equal to 0.5, the servo arm is shifted to 0 signifying the right turn of the robot and if the value is greater than 0.5 then the servo arm is set at 180 degrees. This implies that the obstacle avoiding robot turns left.

- Module 5

Module 5 deals with the LCD component. To simulate this block, an LCD display block is used in Simulink, with the respective pin numbers for register select, data enable, and the four data pins set according to the hardware connections. The value to be printed in the LCD display is set using a MATLAB function block. The inputs to this function block are the value stored in the data store memory and the input given to the servo motor.

```
% Create a MATLAB function [x,y] = fcn(u,v)
% Variable u represents the value from the data store memory
% Variable v represents the input to the servo motor
If u==0, substitute x = ‘DC motor RPM’ and y = ‘300’
Else, substitute x = ‘Alert’
If v==0, substitute y = ‘Turning Right’
Else, substitute y = ‘Turning Left’
End second if
End first if
% x represents the value to be printed in first line of LCD
% y represents the value to be printed in second line of LCD
```

The output from the MATLAB function block is given as input to the LCD display block to be printed in the hardware model.

7. Challenges Faced During Development

- Given that this project does not require soldering, certain connections appear to loosen over time, requiring care and attention.
- Connecting the DC motor with the Arduino board seemed quite complicated as it had to be connected through an L293D driver.
- Checking the connections was quite challenging due to the number of components used to build this robot.
- Using the random number generator to control the motion of the servo motor was complicated and required a data store block.

8. Benefits of Obstacle Avoiding Robot

- The obstacle avoiding robot has a high level of performance and reliability due to the absence of human error [2].
- Different sensors can be used instead of the ultrasonic sensor depending on the requirement.
- The obstacle detection robot can be scaled in manufacturing industries by building complicated functions upon them according to the requirement.
- This project improves the efficiency of automation in vehicles and enables high performance navigation systems [3].

9. Conclusion

This project develops a model of an obstacle avoiding robot by using MATLAB and Arduino technologies to introduce a new automated obstacle detection system. Obstacle avoiding robot is an autonomous device that can automatically detect an obstacle and avoid it by turning in a different direction. This design enables the system to avoid collisions during operation in any environment. With a system that satisfies the prime prerequisite for any autonomous robot, the obstacle detection robot has multiple applications in various fields, particularly in the manufacturing industry where the percentage of human error is high.

With scalability being one of the biggest benefits of this system, advanced functions can be built upon the existing model based upon the requirement. Enabling this mode of automation would result in saving of money and time for both individuals and businesses alike. The project objectives were met successfully, and all the provided test cases were satisfied. Any potential improvements could be added upon the existing system to improve the efficiency and performance standards of the project.

10. Appendix

This appendix section contains the code used for converting the display value into a uint8 value to be displayed in the LCD display component.

```
% LCD display function
function [x,y] = fcn(u,v)
if u==0
    x = [68 67 32 109 111 116 111 114 32 82 80 77 58 32 32 32];
    y = [51 48 48 32 32 32 32 32 32 32 32 32 32 32 32 32];
else
    x = [65 108 101 114 116 33 32 32 32 32 32 32 32 32 32 32];
    if v==0
        y = [84 117 114 110 105 110 103 32 82 105 103 104 105 32 32 32];
    else
        y = [84 117 114 110 105 110 103 32 76 101 102 116 32 32 32 32];
    end
end
```

11. References

[1] Michele Bertoncello and Dominik Wee, "Ten ways autonomous driving could redefine the automotive world" [Online]. Available: https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/ten-ways-autonomous-driving-could-redefine-the-automotive-world. [Accessed: April 3, 2020].

[2] ELECTRONICS HUB, "Obstacle Avoiding Robot using Arduino" [Online]. Available: https://www.electronicshub.org/obstacle-avoiding-robot-arduino/. [Accessed: April 5, 2020].

[3] PANTECH Solutions, "Obstacle Avoidance Robot" [Online]. Available: https://www.pantechsolutions.net/obstacle-avoidance-robot. [Accessed: April 5, 2020].