Obstacle Avoidance and

Voice Control Rover

A PROJECT REPORT

Submitted by

NITHIN U (811720104071)

MOHAMED JAMEER N (811720104061)

SHARAN SHAKTHI G (811720104092)

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

SAMAYAPURAM – 621 112

November, 2022

BONAFIDE CERTIFICATE

Certified that this project report titled **"OBSTACLE AVOIDANCE AND VOICE CONTROL ROVER"** is the bonafide work of **NITHIN U (811720104071), MOHAMED JAMEER N (811720104061) AND SHARAN SHAKTHI G (811720104092)** who carried out the project under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE	**SIGNATURE**
Dr. M. Sivakumar M.E., Ph.D.,	Mrs. S. Rahmath Nisha M.Tech., Ph.D.,
HEAD OF THE DEPARTMENT	**SUPERVISOR**
Department of CSE	ASSISTANT PROFESSOR
K. Ramakrishnan College of Technology	Department of CSE
(Autonomous)	K. Ramakrishnan College of Technology
Samayapuram – 621 112	(Autonomous)
	Samayapuram – 621 112

Submitted for the viva-voce examination held on ………………

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION

We jointly declare that the project report on **"OBSTACLE AVOIDANCE AND VOICE CONTROL ARDIUNO ROVER"** is the result of original work done by us and best of our knowledge, similar work has not been submitted to **"ANNA UNIVERSITY CHENNAI"** for the requirement of Degree of **BACHELOR OF ENGINEERING**. This project report is submitted on the partial fulfilment of the requirement of the award of Degree of **BACHELOR OF ENGINEERING**.

Signature

____

NITHIN U

____

MOHAMED JAMEER N

____

SHARAN SHAKTHI G

Place: Samayapuram

Date:

ACKNOWLEDGEMENT

It is with great pride that we express our gratitude and in-debt to our institution “**K.Ramakrishnan College of Technology (Autonomous)**”, for providing us with the opportunity to do this project.

We are glad to credit honourable chairman **Dr. K.RAMAKRISHNAN**, **B.E.,** for having provided for the facilities during the course of our study in college.

We would like to express our sincere thanks to our beloved Executive Director **Dr. S. KUPPUSAMY, MBA, Ph.D.,** for forwarding to our project and offering adequate duration in completing our project.

We would like to thank **Dr. N. VASUDEVAN, M.E., Ph.D.,** Principal, who gave opportunity to frame the project the full satisfaction.

We whole heartily thanks to **Dr. M. SIVAKUMAR**, **M.E., Ph.D.,** Head of the department, **COMPUTER SCIENCE AND ENGINEERING** for providing his encourage pursuing this project.

I express my deep and sincere gratitude to my project guide
Mrs. S. RAHMATH NISHA, **M.Tech.**, **Ph.D.,** Department of **COMPUTER SCIENCE AND ENGINEERING,** for his incalculable suggestions, creativity, assistance andpatience which motivated me to carry out this project.

I render my sincere thanks to Course Coordinator and other staff members for providing valuable information during the course.

I wish to express my special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work progress.

ABSTRACT

Obstacle detection and avoidance can be considered as the central issue in designing mobile robots. This technology provides the robots with senses which it can use to traverse in unfamiliar environments without damaging itself. In this paper an Obstacle Avoiding Robot is designed which can detect obstacles in its path and maneuver around them without making any collision. It is a robot vehicle that works on Arduino Microcontroller and employs three ultrasonic distance sensors to detect obstacles. This robot has been initiated by mobile command start. The Arduino board was selected as the microcontroller platform and its software counterpart, Arduino Software, was used to carry out the programming. The integration of three ultrasonic distance sensors provides higher accuracy in detecting surrounding obstacles. Being a fully autonomous robot, it successfully maneuvered in unknown environments without any collision. The hardware used in this project is widely available and inexpensive which makes the robot easily replicable.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

RPM	Revolution Per Minute
ACDS	Advanced Combat Direction System
ACRO	Auxiliary Cathode Readout
ALT Key	Alternate Key
A/N	Alphanumeric
APA	All Points Addressable
ASCII	American Standard Code for Information Interchange
ASTAB	Automated Status Board
EPROM	Erasable Programmable Read-Only

CHAPTER 1

INTRODUCTION

Robotics is part of Todays communication. In today's world ROBOTICS is fast growing and interesting field. It is simplest way for latest technology modification. Now a day communication is part of advancement of technology, so we decided to work on ROBOTICS field, and design something which will make human life simpler in day today aspect. Thus we are supporting this cause.

1.1 PURPOSE

An obstacle avoiding robot is an intelligent device, which can automatically sense and overcome obstacles on its path. Obstacle Avoidance is a robotic discipline with the objective of moving vehicles on the basis of the sensorial information. The use of these methods front to classic methods (path planning) is a natural alternative when the scenario is dynamic with an unpredictable behaviour. In these cases, the surroundings do not remain invariable, and thus the sensory information is used to detect the changes consequently adapting moving. It will automatically scan the surrounding for further path.

This project is basic stage of any automatic robot. This ROBOT has sufficient intelligence to cover the maximum area of provided space. It has ultrasonic sensor which are used to sense the obstacles coming in between the path of ROBOT. It will move in a particular direction and avoid the obstacle which is coming in its path. We have used two D.C motors to give motion to the ROBOT. The construction of the ROBOT circuit is easy and small the electronics parts used in the ROBOT circuits are easily available and cheap too.

Obstacle avoiding robots can be used in almost all mobile robot navigation systems. They can be used for household work like automatic vacuum cleaning. They can also be used in dangerous environments, where human penetration could be fatal.

This is one of the most important aspects of mobile robotics. Without it robot movement would be very restrictive and fragile. This tutorial explains obstacle avoidance using ultrasonic sensors. This project also presents a dynamic steering algorithm which ensures that the robot doesn’t have to stop in front of an obstacle which allows robot to navigate smoothly in an unknown environment, avoiding collisions.

1.2 PROBLEM DESCRIPTION

Robotics is the branch of technology that deals with the design, construction, operation, and application of robots. A machine capable of carrying out a complex series of actions automatically, esp. one programmable by a computers is defined as a robot.

The project is to develop a robot that will move according to the code assigned but find a free space, navigating from any obstacle on its way. This kind of obstacle is very useful in industries where automatic supervision is needed, for example, in places where it might be risky for humans to be. This robot can also be made by putting other sensors like light sensors or line sensors, ultrasonic sensors and ultrasound sensor depending on the need.

The project is design to build an obstacle avoidance robotic vehicle using ultrasonic sensors for its movement. A microcontroller (ATmega328) is used to achieve the desired operation. A robot is a machine that can perform task automatically or with guidance. The project proposes robotic vehicle that has an intelligence built in it such that it directs itself whenever an obstacle comes in its path. This robotic vehicle is built, using a micro-controller of AT mega 328 family.

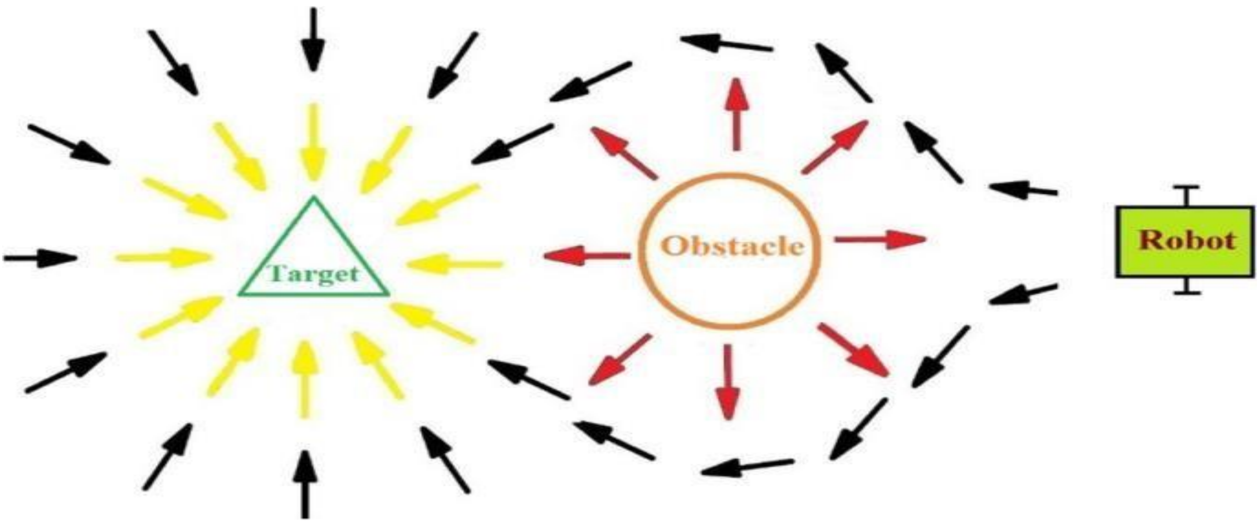

FIG 1.1

An ultrasonic sensor is used to detect any obstacle ahead of it and sends a command to the micro-controller. Depending on the input signal received, the micro-controller redirects the robot to move in an alternate direction by actuating the motors which are interfaced to it through a motor driver. Some of the project is built with the Ultrasonic Sensor has its own application so in our project those application is not compactable so we are using ultrasonic sensor.

1.3 OVERVIEW

Obstacle avoidance is one of the most important aspects of mobile robotics. Without it, robot movement would be very restrictive and fragile. This project proposes robotic vehicle that has an intelligence built in it such that it directs itself whenever an obstacle comes in its path.

So, to protect the robot from any physical damages. This can be design to build an obstacle avoidance robotic vehicle using ultrasonic sensors for its movement. A micro-controller (AT mega 328P) is used to achieve the desired operation. An ultrasonic sensor is used to detect any obstacle ahead of it and sends a command to the micro-controller.

Depending on the input signal received, the micro-controller redirects the robot to move in an alternate direction by actuating the motors which are interfaced to it through a motor driver.

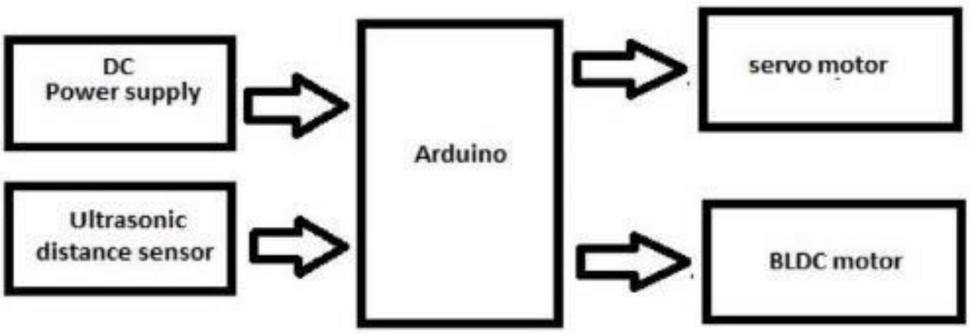

FIG 1.2

It is an autonomous robot which will be able to avoid every obstacle in its path. It will use an ultrasonic distance sensor and a servo motor. The robot will check how far the nearest obstacle is (in every direction) and then decide upon the actions to be taken.

1.4 SCOPE

This task work has been limited to short range Bluetooth module. Utilizing a long range modules and other availability gadgets will bring about network with the robot for significant distances. Picture preparing can be executed in the robot to distinguish the shading and the items.

A warm camera can be introduced to detect the warmth produced by bodies valuable in military purposes to distinguish foes on the lines. Programmed Targeting System can be executed in the robot for following the objective.

Further upgrade in venture can be utilized for Home security and military purposes where the orders can be given to robot without chance by expanding the range and by introducing cameras.

The robot is valuable in places where people discover hard to reach however human voice comes to. For example, in fire circumstances, in profoundly poisonous zones.

1.5 DEFINITION AND ACRONYMS

This obstacle avoiding machine uses an HC-SR04 sensor mounted on top of a servo to locate walls in a maze using echolocation. The 4 DC motors are powered by a 9V battery back that runs from an L298 chip.

The Arduino, servo, and sensor are powered by a separate 9V battery. Some important Acronyms are

ACDS	Advanced Combat Direction System
ACRO	Auxiliary Cathode Readout
ALT Key	Alternate Key
A/N	Alphanumeric
APA	All Points Addressable
ASCII	American Standard Code for InformationInterchange.

ASTAB	Automated Status Board
EPROM	Erasable Programmable Read-Only

Table 1.1

Obstacle avoiding robots can be used in almost all mobile robot navigation systems. They can be used for household work like automatic vacuum cleaning.

They can also be used in dangerous environments, where human penetration could be fatal. Some Common Definition

Voice Detection - Voice or speaker recognition is the ability of a machine or program to receive and interpret dictation or to understand and carry out spoken commands.

Automation - Automation is a term for technology applications where human input is minimized. This includes business process automation (BPA), IT automation, personal applications such as home automation and more.

Obstacle - Obstacle Avoiding Robot is an intelligent device that can automatically sense the obstacle in front of it and avoid them by turning itself in another direction. This design allows the robot to navigate in an unknown environment by avoiding collisions, which is a primary requirement for any autonomous mobile robot.

Recognition - The act of identifying someone or something because of previous knowledge, or to formally acknowledge someone.

1.6 AIM AND OBJECTIVES:

It is an autonomous robot which will be able to avoid every obstacle in its path. It will use an ultrasonic distance sensor and a servo motor. The robot will check how far the nearest obstacle is (in every direction) and then decide upon the actions to be taken.

The aim of this project is to implement an obstacle avoiding robot using ultrasonic sensor and Arduino. All the connections are made as per the circuit diagram. The working of the project is explained below.

When the robot is powered on, both the motors of the robot will run normally and the robot moves forward. During this time, the ultrasonic sensor continuously calculate the distance between the robot and the reflective surface.

In mobile robotics, the objective of obstacle avoidance is generally to navigate from one location to another while avoiding collisions with obstacles in the environment.

These obstacles can be structural or static, such as walls or doors in the environment, but also dynamic such as other robots or humans moving in the same environment.

Often this problem is solved in the control algorithm of the robot in a reactive manner, in contrast to (global) path planning where a precomputed obstacle-free path is followed by the robot.

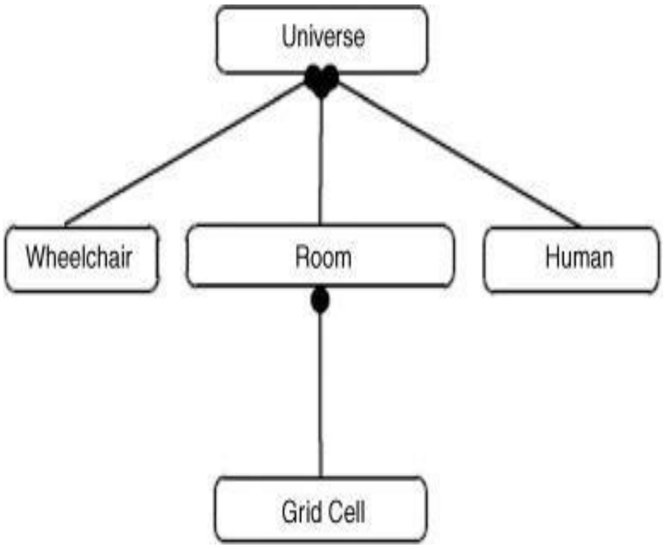

FIG 1.3

The robot controller reactively alters the robot's course to avoid the obstacle and afterwards steers back to the original trajectory. The obstacle avoidance problem can be interpreted as a resource allocation problem in which the environment is divided into a set of grid cells and the desired trajectory is represented by the allocation of a sequence of these grid cells.

The resource Holon's representing the environment are, thus, divided into grid cell resource Holon's that can be individually allocated. Figure 6.3 shows the relevant resources in this use case.

CHAPTER 2

GENERAL DESCRIPTION

In 2003, worldwide speculation in modern robots up 19%. In 2004, orders for robots were up another 18% to the highest level ever recorded. Overall development in the period 2004-2007 conjecture at a normal yearly pace of about 7%. More than 600,000 family unit robots being used several millions in the next few years. Various researches have been made by different researchers in developing this project. Be that as it may, they serve an alternate application and have various innovations actualized. Some of those papers are mentioned below stating their technology and application. Robot control design using android smartphone authors: Mrumalk pathak, Javed khan, Aarushi koul, Reshma kalane Raunak Varshney.

The motivation behind this paper is to furnish amazing computational android stages with less difficult robot equipment design. This paper depicts how to control a robot utilizing portable through Bluetooth communication, a few highlights about Bluetooth innovation, segments of the versatile and robot.

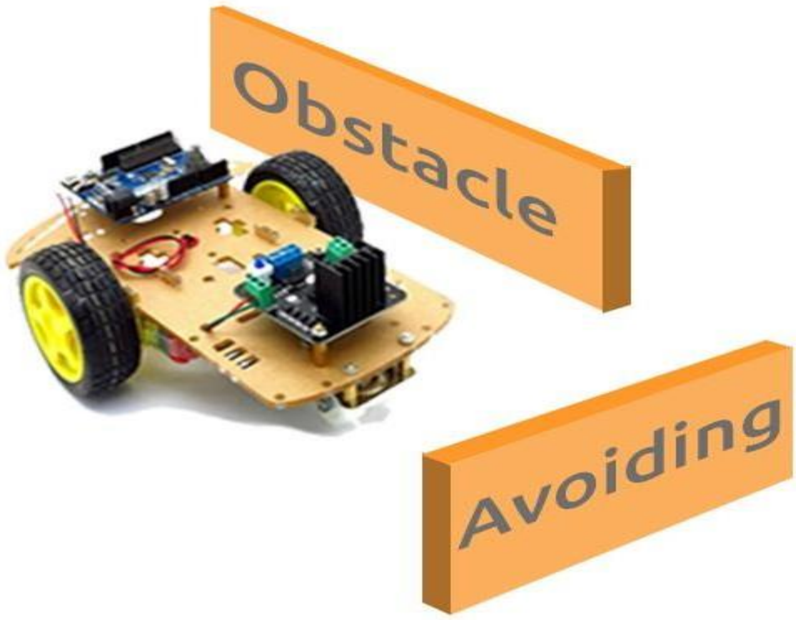

FIG 2.1

It present an audit of robots constrained by smart phone by means of moving the robot upward, reverse, left and right side by the android application, for example, Arduino, Bluetooth Smart Phone Controlled Robot Using ATMEGA328

Microcontroller. Author's: Aniket R. Yeole, Sapana M. Bramhankar, Monali D, Wani Mukesh P. Mahajan. In this paper have structured a robot that can be controlled using an application running on an android smartphone.

It sends control order by means of Bluetooth which has certain highlights like controlling the speed of the engine, detecting and sharing the data with telephone about the bearing and separation of the robot from the closest hindrance**.**

2.1 User Needs

Obstacle Avoiding Robot is an intelligent device that can automatically sense the obstacle in front of it and avoid them by turning itself in another direction. This design allows the robot to navigate in an unknown environment by avoiding collisions, which is a primary requirement for any autonomous mobile robot

The working principle of the robot is transmitting sensed signal to the microcontroller to control the DC motors for obstacle avoidance. The H-bridge L293D controls the direction of the motors to move either clockwise or anti-clockwise directions as provided by the microcontroller.

. A more general and commonly employed method for obstacle avoidance is based on edge detection. A disadvantage with obstacle avoidance based on edge detecting is the need of the robot to stop in front of an obstacle in order to provide a more accurate measurement.

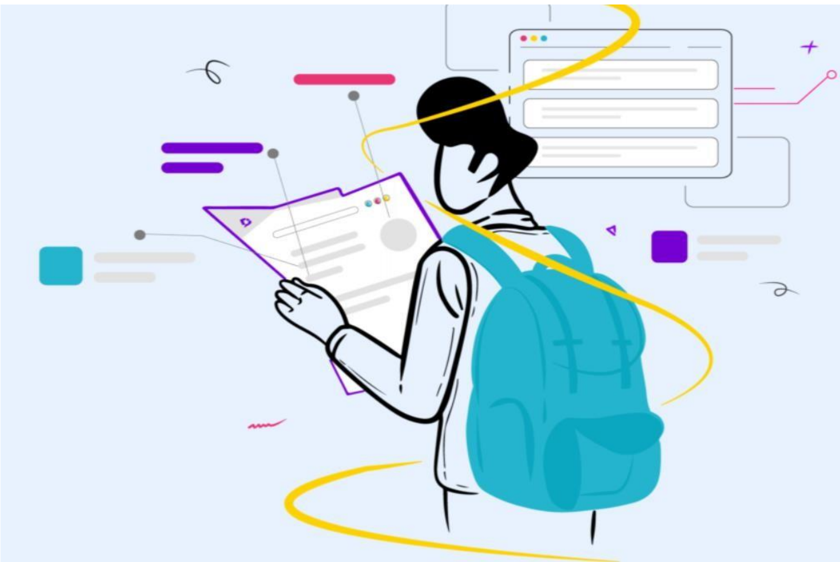

FIG 2.2

Obstacle Avoiding Robot is an intelligent device that can automatically sense the obstacle in front of it and avoid them by turning itself in another direction. This design allows the robot to navigate in an unknown environment by avoiding collisions, which is a primary requirement for any autonomous mobile robot

2.2 ASSUMPTIONS AND DEPENDENCIES

We reviewed different obstacle detecting robot mechanisms that have been built by a lot of students and other practitioners that are in existence. For an autonomous mobile robot performing a navigation-based task in a vague environment, to detect and to avoid encountered obstacles is an important issue and a key function for the robot body safety as well as for the task continuity.

Obstacle detection and avoidance in a real world environment that appears so easy to humans is a rather difficult task for autonomous mobile robots and is still a well-researched topic in robotics. In many previous works, a wide range of sensors and various methods for detecting and avoiding obstacles for mobile robot purpose have been proposed. Good references related to the developed sensor systems and proposed detection and avoidance algorithms can be found.

Based on these developed sensor systems, various approaches related to this work can be grouped Robots need miscellaneous of sensors to obtain information about the world around them. Sensors will help detect position, velocity, acceleration and range for the object in the robot's workspace. There is a variety of sensors used to detect the range of an object.

One of the most common range finders is the ultrasonic transducer. Vision systems are also used to greatly improve the robot’s versatility, speed and accuracy for its complex and difficult task. Electronic signals are sent to a mobile robot’s motor controllers and auditory signals can guide the blind traveller around the obstacles the developed robot uses ultrasonic range finder for detection and mapping to avoid collision with the unexpected obstacles. “Obstacle-avoiding robot with IR motion Sensors” has been designed and developed, robot platform was not designed for specific task but as a general wheeled autonomous platform.

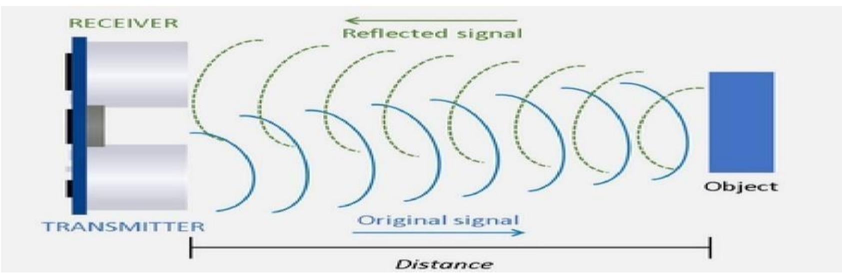

FIG 2.3

It can therefore be used for educational, research or industrial implementation. Students can use it to learn the microcontroller programming using C or C++, Arduino Uno compiler, Ultrasonic sensors characteristics, motor driving circuit and signal condition circuit design. Research on obstacle avoidance robot at the polytechnic level can help students to develop communication, technical skills and teamwork.

The design of such robot is very flexible and various methods can be adapted for another implementation. It shows that Ultrasonic sensors are more sensitive compared to Ultrasonic sensors while detecting human being.

2.3 WORKING PRINCIPLE

The working principle of the robot is transmitting sensed signal to the microcontroller to control the DC motors for obstacle avoidance. The H-bridge L293D controls the direction of the motors to move either clockwise or anti-clockwise directions as provided by the microcontroller.

If Ultrasonic sensor detect a moving object while IR sensor does not detect any object, the robot will move backward (motor 1 and motor 2 counter clockwise). On the other hand, if PIR detect object and IR sensor also detect object, the robot will stop (motor 1 and motor 2 OFF).

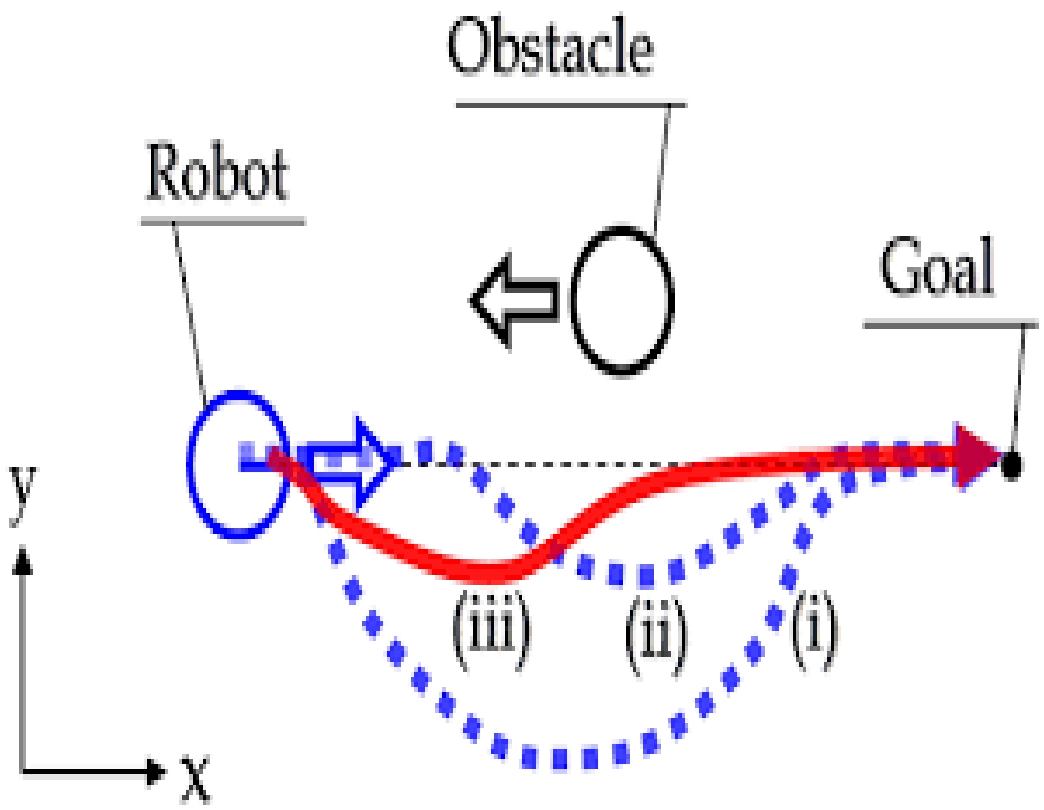

FIG 2.4

After 50ms, motor 1 will move clockwise and the robot will turn left. After 500ms, the robot will move forward (motor 1 and motor 2 clockwise) and after 1000ms, both motors will stop. From the flow chart in Figure 10, it shows that the Ultrasonic sensors are very effective in sensing signals in their path for the obstacle avoiding robots to evade obstacles in its path.

2.4 BLOCK DIAGRAM

The working principle of the robot is transmitting sensed signal to the microcontroller to control the DC motors for obstacle avoidance. The H-bridge L293D controls the direction of the motors to move either clockwise or anti-clockwise directions as provided by the microcontroller.

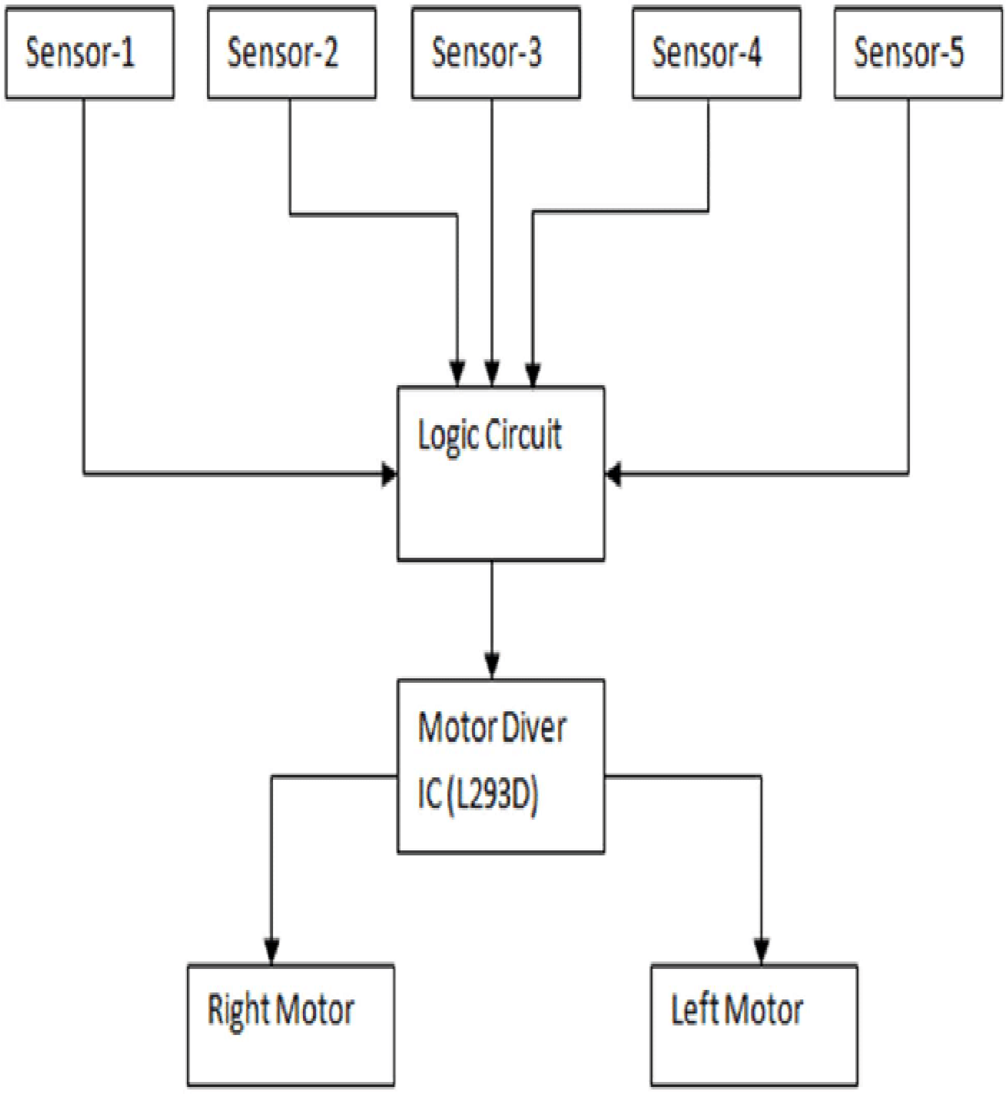

FIG 2.5

There is an ultrasonic sensor around a vehicle that is used to recognise any obstacle. The ultrasonic sensor transmits sound waves and reflects sound from an object. At the point where an object is episode of ultrasonic waves, energy impression occurs up to 180 degrees.

Fig 2.5 is another model uses four sensors to detect object and perform operations. Using instead of IR Sensor Ultrasonic plays major role to detect object in an optimal way by emitting and receiving Ultrasonic ways.

The sonar system is used in HC-SR04 ultrasonic sensor to determine distance to an object like bats do. It offers excellent non-contact range detection from about 2 cm to 400 cm or 1feet to 13 feet.

Its operation is not affected by sunlight or black material. The ultrasonic sensor emits the short and high frequency signal. If they detect any object, then they reflect back echo signal which is taken as input to the sensor through Echo pin.

Firstly user initialize Trigger and Echo pin as low and push the robot in forward direction. When obstacle is detected Echo pin will give input as high to microcontroller. Pulse In function is used for calculating the time of distance from the obstacle. Every time the function waits for pin to go high and starts timing, then timing will be stopped when pin go to low.

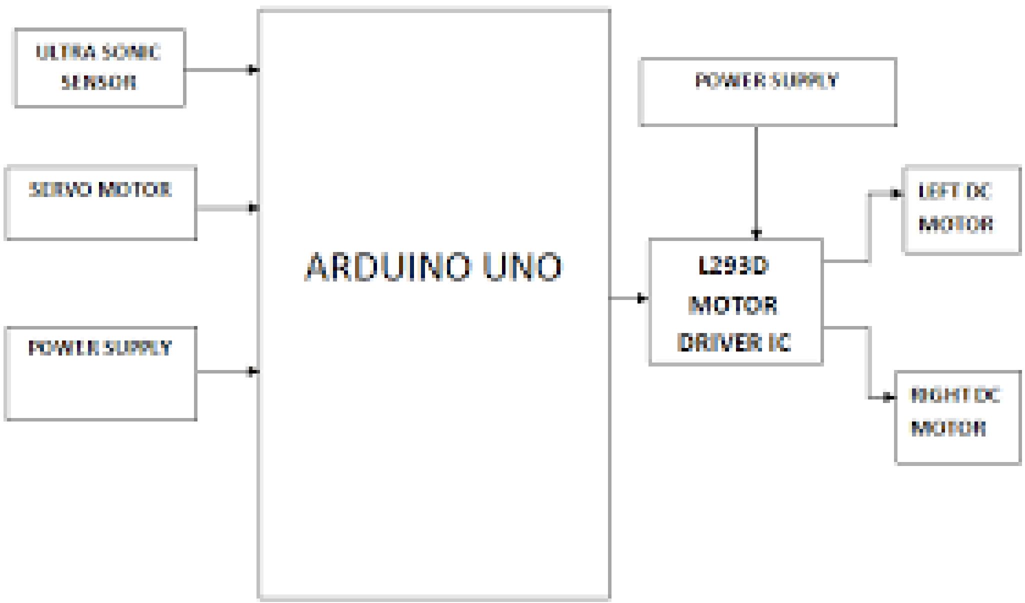

FIG 2.6

It returns the pulse length in microseconds or when complete pulse was not received within the timeout it returns. The timing has been determined means it gives length of the pulse and will show errors in shorter pulses.

Pulses from 10microseconds to 3 minutes in length are taken into consideration. After determining the time, it converts into a distance.

If the distance of object is moderate then speed of robot get reduced and will take left turn, If obstacle is present in left side then it will take right turn.

If the distance of object is short then speed of robot get reduced and will turn in backward direction and then can go in left or right direction. This robot was built with an Arduino development board on which microcontroller is placed.

MOVEMENT	**PIN 10**	**PIN 11**	**PIN 12**	**PIN 13**
FORWARD	1	0	0	1
BACKWARD	0	1	1	0
LEFT	0	0	1	0
RIGHT	0	1	0	1

Table 2.1

CHAPTER 3

SYSTEM FEATURES AND REQUIREMENTS

Ultrasonic sensor based obstacle detection SDM-IO Ultrasonic sensor used for range detection range between obstacle and sensor is 20 cm. Automatic moving system Arduino based Robot car on board motor driver Two no’s of motor wheel and single motor less wheel Battery operated robot car Four no’s of AA Battery (+6V).

3.1 INTEGRATED SYSTEM

When an electrical pulse of high voltage is applied to the ultrasonic transducer it vibrates across a specific spectrum of frequencies and generates a burst of sound waves. Whenever any obstacle comes ahead of the ultrasonic sensor the sound waves will reflect in the form of echo and generates an electric pulse.

It calculates the time taken between sending sound waves and receiving the echo. The echo patterns will be compared with the patterns of sound waves to determine the detected signal’s condition.

The obstacle detection and avoidance robot now successfully detects and obstacle and then turns right to avoid the obstacle, enough though it comes across an obstacle then it would completely come to a halt.

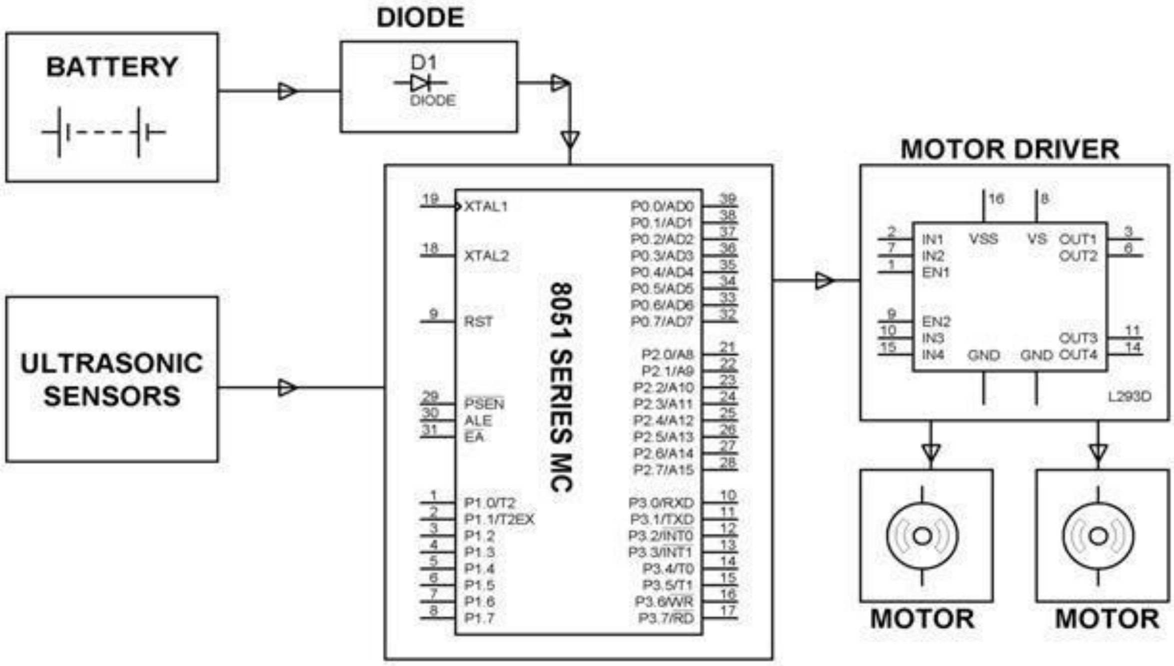

FIG 3.1

The ultrasonic receiver shall detect signal from the ultrasonic transmitter while the transmit waves hit on the object.

The combination of these two sensors will allow the robot to detect the object in its path. The ultrasonic sensor is attached in front of the robot and that sensor will also help the robot navigate through the hall of any building.

3.2 THEORY OF OPERATION

The ultrasonic sensor emits the short and high-frequency signal. These propagate in the air at the velocity of sound. If they hit any object, then they reflect an echo signal to the sensor. The ultrasonic sensor consists of a multi-vibrator, fixed to the base.

The multi-vibrator is a combination of a resonator and a vibrator. The resonator delivers ultrasonic wave generated by the vibration. The ultrasonic sensor consists of two parts;

The emitter which produces a 40 kHz sound wave

The detector detects a 40 kHz sound wave and sends an electrical signal back to the microcontroller.

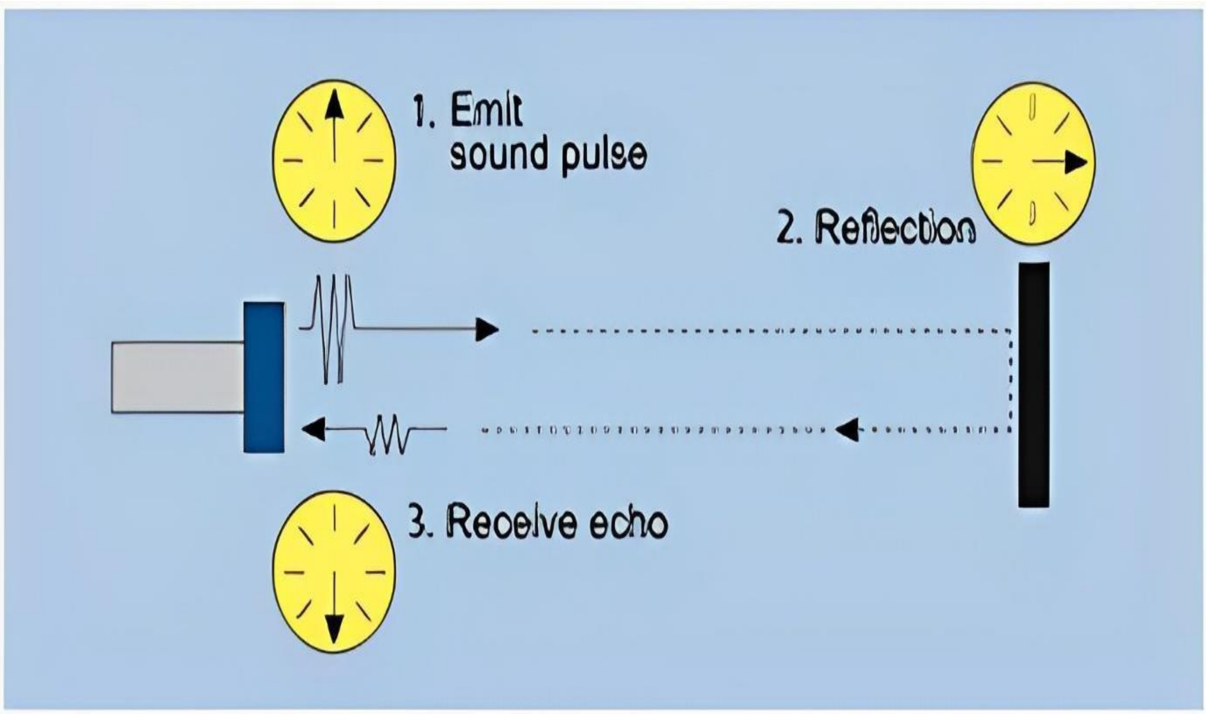

FIG 3.2

The ultrasonic sensor enables the robot to virtually see and recognize an object, avoid obstacles, measure distance. The operating range of the ultrasonic sensor is 10 cm to 30 cm.

When the robot is powered on, both the motors of the obstacle avoiding robot will run normally and the robot will move forward. During this whole time, the ultrasonic sensor will be continuously calculating the distance between the reflecting surface and the robot.

This information is processed by the Arduino from the sensor. If the distance between the robot and the obstacle are less than limit set in the Arduino, the Robot will stop and scans in right and left directions for new distance by using Ultrasonic sensor.

If the left distance is more than the right distance, the robot will turn in left direction by commanding the left wheel to move in forward motion and the right wheel to move in backward direction.

Similarly, if the right distance is more than left distance, the robot will turn in right direction. The robot will not collapse with any obstacle.

Whenever the robot is going on the desired path the ultrasonic sensor transmits the ultrasonic waves continuously from its sensor head. Whenever an obstacle comes ahead of it the ultrasonic waves are reflected from an object and that information is passed to the microcontroller. The microcontroller controls the motors left, right, back, front, based on ultrasonic signals. To control the speed of each motor pulse width modulation is used (PWM).

3.3 PLATFORM STRUCTURE

Obstacle avoiding robot, which is the another example of arduino project. Basically an Obstacle avoiding robot consist of some set of sensors like ultrasonic sensor (distance measurement sensor), which measure the distance of an object from a particular distance and then it returns the distance as an input to the microcontroller arduino.

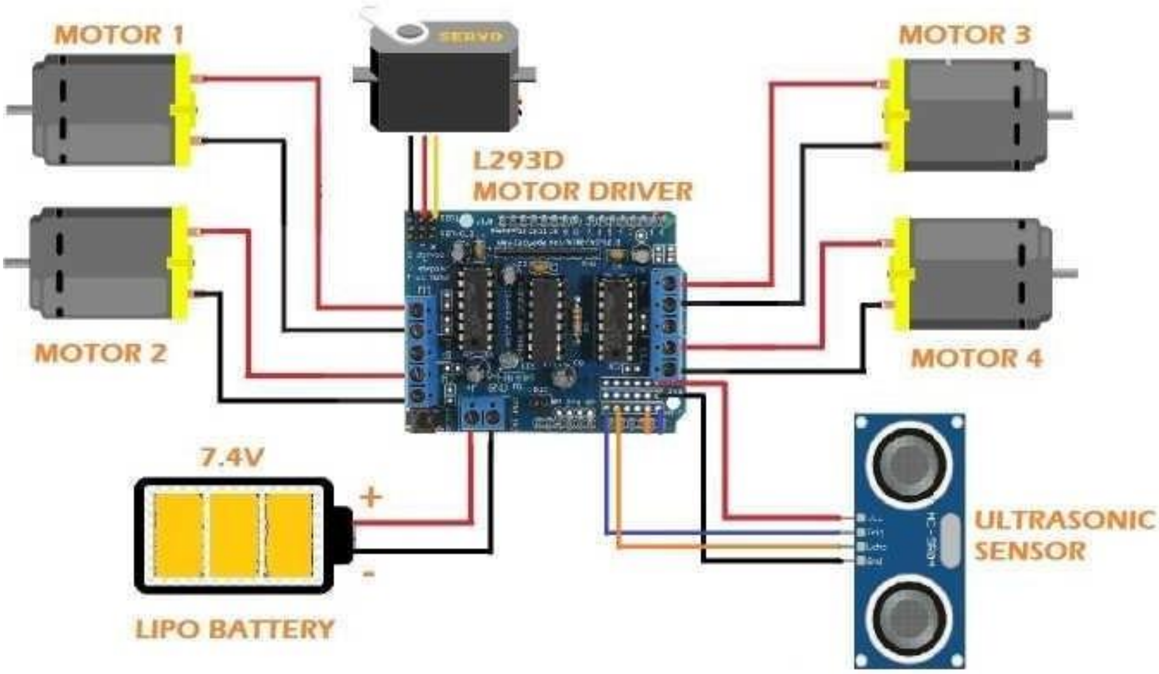

FIG 3.3

Arduino will further process the input for executing some set of instruction. Before making the robot you have to know about some basic important things. Obstacle avoiding robot is a basic example of automation projects, where we are building a car by the help of arduino uno and ultrasonic sensor.

The robot will automatically control itself by avoiding the obstacles coming in front of it. The robot will use the data of ultrasonic sensor and find its own path by avoiding the obstacles.

3.4 FEATURES

The concept of Robotics is now used in every sector whether it is in manufacturing industry, medical, transport etc. Obstacle avoidance is one of the features that is needed for the automated mobile robots.

In this there is a robot that consist of Arduino UNO (Microcontroller) and sensor that detect presence of obstacles. Programming is done by the Arduino software. The ultrasonic sensor is highly accurate in detecting obstacles in the surroundings. This is a wheeled robot.

There are some very common methods for robot navigation like wall-following, edge detection, line following. One of the commercial systems uses wall following

Method on a floor cleaning robot for long hallways. A more general and commonly employed method for obstacle avoidance is based on edge detection. A disadvantage with obstacle avoidance based on edge detecting is the need of the robot to stop in front of an obstacle in order to provide a more accurate measurement.

FIG 3.4

All mobile robots feature some kind of collision avoidance, ranging from primitive algorithms that detect an obstacle and stop the robot in order to avoid a collision, using some sophisticated algorithm, that enable the robot to detour obstacles.

The latter algorithms are more complex, since they involve detection of an obstacle as well as some kind of quantitative measurements concerning the obstacle's dimensions. Once these have been determined, the obstacle avoidance algorithm needs to steer the robot around the obstacle and resume motion toward the original target.

CHAPTER 4

INTERFACE REQUIREMENT

Since the launch of the Arduino open-source platform, the brand has established themselves at the centre of an expansive open-source community. The Arduino ecosystem is comprised of a diverse combination of hardware and software.

The versatility of Arduino and its simple interface makes it a leading choice for a wide range of users around the world from hobbyists, designers, and artists to product prototypes.

The Arduino board is connected to a computer via USB, where it connects with the Arduino development environment (IDE). The user writes the Arduino code in the IDE, then uploads it to the microcontroller which executes the code, interacting with inputs and outputs such as sensors, motors, and lights.

Both beginners and experts have access to a wealth of free resources and materials to support them. Users can look up information on how to set up their board or even how to code on Arduino.

The open source behind Arduino has made it particularly friendly to new and experienced users. There are thousands of Arduino code examples available online. In this post, we'll take you through some basic principles of coding for Arduino.

4.1 MOBILE PLATFORM

An Obstacle Avoiding Robot is a type of autonomous mobile robot that avoids collision with unexpected obstacles. In this project, an Obstacle Avoiding Robot is designed.

It is an Arduino based robot that uses Ultrasonic range finder sensors to avoid collisions. Here to control the rover need to use mobile application Arduino Bluetooth Controller.

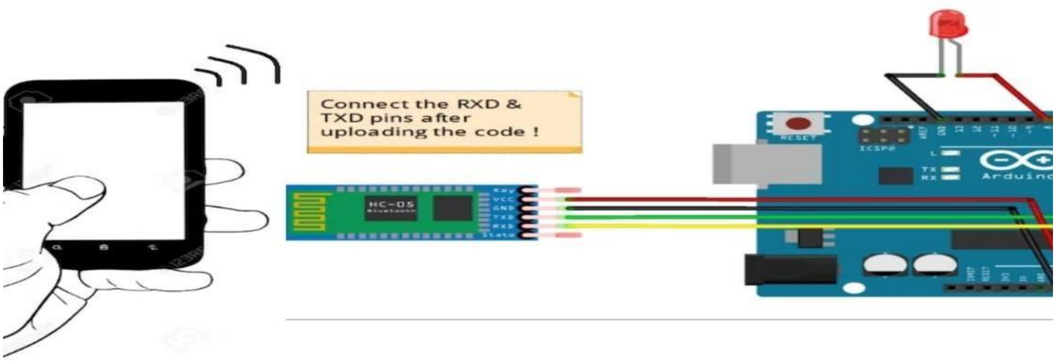

FIG 4.1

If the communication is over a short range, then Bluetooth would be more helpful. WIFI is fast and secure than Bluetooth. Moreover, you need an extra module for wireless communication. You can use a USB dongle, which allows the Arduino to wirelessly communicate with any computer or the web. You can use multi-button key fobs to add very cool remote control capability to your Arduino projects

4.2 Connection Table

Servos are mainly used on angular or linear position and for specific velocity, and acceleration.

Arduino Uno	**Servo Motor**
PIN 3	Single Pin Orange wire
VCC	VCC Red Wire
GND	GND Black Wire

Table 4.1

Ultrasonic sensors are used primarily as proximity sensors. They can be found in automobile self-parking technology and anti-collision safety systems. Ultrasonic sensors are also used in robotic obstacle detection systems, as well as manufacturing technology.

Arduino Uno	**Ultrasonic Sensor**
VCC	VCC
PIN 5	Trig
PIN 6	Echo
GND	GND

Table 4.2

Motor drivers acts as an interface between the motors and the control circuits. Motor require high amount of current whereas the controller circuit works on low current signals.

So the function of motor drivers is to take a low-current control signal and then turn it into a higher-current signal that can drive a motor.

Arduino Uno	Motor Driver
PIN 7	IN 1
PIN 8	IN 2
PIN 9	IN 3
PIN 10	IN 4

Table 4.3

These are the components that are connected with Arduino UNO board which perform instructions which implemented by the particular component available in the Arduino rover.

4.3 HARDWARE ASSEMBLY

Second Module of the project is to assemble components that can be performed by following steps

Step 1: How Ultrasonic Sensor Can Be Used to Avoid Obstacles.

Step 2: Attach the Motor and Wheel to the Chassis.

Step 3: Attach Arduino in Chassis.

Step 4: Connect Motor Wire in Arduino.

Step 5: Ultrasonic Sensor.

Step 6: Mount the ULTRASONIC Sensor with Servo

FIG 4.2

In the below paragraph how does the sensor work. This obstacle-avoiding robot using Arduino Uno works as a distance measurer.

- An ultrasonic sensor has two parts one is a transmitter and another is the
- Receiver that is known as a trigger and echoes.
- The trigger is the transmitter part and transmits the ultrasonic wave.
- Another part that is receiver an echo receive that transmitter ultrasonic waves transmitted from the trigger.
- Now we calculate that in how much time the ultrasonic waves return back to the receiver and divide by 2 because the time travel is double.

Hardware need to Interface these pins which are

1. 5V input power supply
2. Trigger Pulse Input
3. Echo Pulse Output
4. Ground (0v)

Here power supply is given to VCC, trigger pulse input is given to port p1.7 of msp430, echo pulse output is connected to port p1.6 of msp430 and ground pin is connected to 0v ground.

Specifications

- Speed: 1000rpm
- Operating voltages:

12V Hardware Interfacing DC motor is connected to the output ports 3, 6, 11, 14 of H-Bridge L293D.

And further H-Bridge is connected to 3600 DC BATTERY is the main power supply for the circuit. The input power supply is 12V.

4.4 ADVANTAGES AND APPLICATION

The sensing can be increased. A Bluetooth module and a camera can be attached, so that the user can see the obstacle and take pictures and also can take videos of it. As the sensor can detect only the obstacles with reflective surface, so in future work can be carried out to detect and avoid obstacles of absorbing surface.

Obstacle avoiding robots can be used in almost all mobile robot navigation systems. They can be used for household work like automatic vacuum cleaning. They can also be used in dangerous environments, where human penetration could be fatal. Obstacle avoidance is the task of satisfying some control objective subject to non-intersection or non-collision position constraints.

What is critical about obstacle avoidance concept in this area is the growing need of usage of unmanned aerial vehicles in urban areas for especially military applications where it can be very useful in city wars.

Normally obstacle avoidance is considered to be distinct from path planning in that one is usually implemented as a reactive control law while the other involves the pre-computation of an obstacle-free path which a controller will then guide a robot along. With recent advanced in the autonomous vehicles sector, a good and dependable obstacle avoidance feature of a driverless platform is also required to have a robust obstacle detection module.

4.4.1 ADVANTAGES OF OBSTACLE AVOIDANCE

It is an autonomous robot which will be able to avoid every obstacle in its path. It will use an ultrasonic distance sensor and a servo motor. The robot will check how far the nearest obstacle is (in every direction) and then decide upon the actions to be taken.

Obstacle avoider robot is the important part of mobile robotics. Obstacle avoidance is task which is used for detecting the presence of object in a path of robot or any vehicle.

Obstacle avoiding robot is an intelligence device, which is used to protect the robot from any physical damages. It automatically sense and overcome the obstacles on its path.

In this project, we will study how to design and simulate an obstacle avoider robot using AVR ATmega16 microcontroller and Analog IR Sensor. This robot designed is an automatic robot i.e. no manual control is required for the operation of robot.

FIG 4.3

In obstacle avoider robot we use ADC of the ATmega16 microcontroller to convert Analog signals to digital values. After converting the IR sensors signal into digital equivalent signal

I.e. Threshold value (3V in our project case). According to the output of above comparison, the microcontroller send the control signal to the DC motor driver Integrated Circuit (L293D) of robot to move the robot in forward, left or right direction. The above operation of sensors output comparison, ADC conversions and robot control continues till the power source is applied.

4.4.2 DRAWBACKS OF EXISTING SYSTEM

A more general and commonly employed method for obstacle avoidance is based on edge detection. A disadvantage with obstacle avoidance based on edge detecting is the need of the robot to stop in front of an obstacle in order to provide a more accurate measurement.

All mobile robots feature some kind of collision avoidance, ranging from primitive algorithms that detect an obstacle and stop the robot in order to avoid a collision, using some sophisticated algorithm, that enable the robot to detour obstacles.

The latter algorithms are more complex, since they involve detection of an obstacle as well as some kind of quantitative measurements concerning the obstacle's

Dimensions . Once these have been determined, the obstacle avoidance algorithm needs to steer the robot around the obstacle and resume motion toward the original target.

In simple robot, steering algorithm is used for robotic actions in which driver or a human being is controlling the robot using remote. Here driver is present, who can see the obstacle and navigate robot accordingly.

4.4.3 APPLICATIONS

- Obstacle avoiding robots can be used in almost all mobile robot navigation systems.
- They can be used for household work like automatic vacuum cleaning.
- They can also be used in dangerous environments, where human penetration could be fatal.

CHAPTER 5

ALGORITHM TO IMPLEMENT

Obstacle avoidance algorithms can usually be classified into two categories: off-line planning algorithms and on-line adjustment algorithms. Off-line obstacle avoidance algorithms solve obstacle avoidance problem by path planning, such as rapidly-exploring random tree (RRT), probabilistic road map (PRM).

5.1 KHATIB'S ALGORITHM

Khatib introduced an algorithm called APF in 1985 [6]. This algorithm considers the robot as a point in potential fields and then combines stretching toward the target and repulsion of obstacles. The final path of the output is the intended path.

Fuzzy logic and neuro-fuzzy networks can also be used to avoid obstacles. In this regard, fuzzy logic can be helpful in solving the problems caused by the ocean flow in the underlying surfaces.

It should be noted that the algorithms discussed in the research background for avoiding an obstacle are different from each other, and each algorithm has proposed a separate and new method for avoiding obstacles. So far, novel methods to avoid obstacles in addition to the mentioned methods are also presented. In the following section, we first investigate the artificial potential field algorithm and then modify this algorithm in FIG 4.5. In, the state-dependent Ricotta equation (SDRE) algorithm is studied on the motion design of cable-suspended robots with uncertainties and moving obstacles.

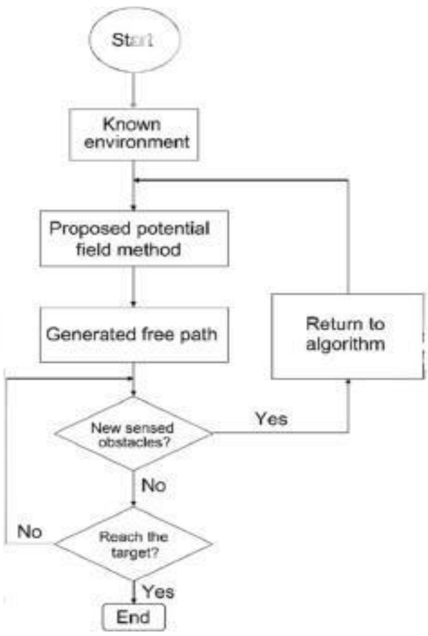

FIG 5.1

A method for controlling the tracing of a robot has been developed by the SDRE. In the fuzzy logic method is used to predict the movement of obstacles and accelerate the linear velocity of the robot.

The movement trend of the obstacle in approaching the robot is divided away from the robot and translated by the robot. With respect to the three trends, the robot can increase the velocity, decrease the velocity to pass by the obstacle, or stop to wait for the obstacle to pass.

Some simulation results show that the proposed method can help the robot avoid an obstacle without changing the initial path. In FIG 4.5 have a four- dimensional unmanned aerial vehicle's platform equipped with a miniature computer equipped with a set of small sensors for this work. The platform is capable of accurately estimating the precision mode, tracking the speed of the user, and providing uninterrupted navigation.

The robot estimates its linear velocity by filtering a Kalman filter from the indirect and optical flow with the corresponding distance measurements. In a new method for defining the path in the presence of obstacles is proposed, which describes the curve as a two-level intersection.

The planner, based on the definition of the path along with the cascaded control architecture, uses a non-linear control technique for both control loops (position and attitude) to create a framework for manipulating multivariate behavior. The method has been shown to be able to create a safe path based on perceived obstacles in real time and avoiding collisions.

Refer to the tracking control of Euler-Lagrange system problems with external impediments in an environment containing obstacles. According to a new sliding discovery, a new tracking controller is proposed to determine the tracing, convergent errors reach zero as infinite time.

In addition, based on the non-destructive sliding terminal, a simultaneous control algorithm with time constraints has also been developed to ensure that tracking errors are approaching a restricted area close to the source at a given time. By introducing multi-purpose collision avoidance functions, both controllers can ensure that the obstacle is avoided.

In the mobile sensor deployment (MSD) problem had been studied in mobile sensor networks (MSNs), aiming at deploying mobile sensors to provide target coverage and network connectivity with requirements of moving sensors. This problem is divided into two sub-problems, the Target Coverage (TCOV) problem and Network Connectivity (NCON) problem.

For the TCOV problem, it is proved that it is NP-hard. For a special case of TCOV, an extended Hungarian method is provided to achieve an optimal solution; for general cases, two heuristic algorithms are proposed based on the clique partition and Verona diagram, respectively.

For the NCON problem, at first, an edge constrained Steiner tree algorithm is proposed to find the destinations of mobile sensors, and then, we used extended Hungarian to dispatch rest sensors to connect the network.

Theoretical analysis and simulation results have shown that compared to extended Hungarian algorithm and basic algorithm, the solutions based on TV-Greedy have low complexity and are very close to the optimum.

In a novel approach is presented to obtain the optimal performance bounds for a multi-hop multi-rate wireless data network. First, the optimal relay placements are determined for a target terminal located at a distance D away from the access point.

Second, for a general analytical PHY layer throughput model, the maximum achievable MAC throughput is determined as a function of the number of relays for a target located at distance D. Researchers around the world have been researching about artificial potential field defects and have presented suggestions for improving this method.

Here, a regulative agent has been added to improve the artificial potential field algorithm with the target of overcoming the minimum local and inaccessible target.

When the robot is close to the target, this regulative agent reduces the attraction control as a linear function, and the repulsion decreases as a higher-order function ($M \geq 3$). A flowchart illustrating the steps of the modified APF algorithm is given in Fig 4.5.

In this section, a comparison is firstly done between the simulation results of the artificial potential field algorithm and the modified artificial potential field, then the obstacle avoidance of the mobile robot is displayed against a variety of obstacles that are simulated using the modified artificial potential field algorithm.

Potential field algorithm introduced by Khatib is well-known in path planning for robots. The algorithm is very simple yet provides real-time path planning and effective to avoid robot’s collision with obstacles. The purpose of the paper is to implement and modify this algorithm for quadrotor path planning.

This paper considers the integration of gap-based local navigation methods with artificial potential field (APF) methods to derive a local planning module for hierarchical navigation systems that has provable collision-free properties. Large-scale flocking behavior has been an area of significant research in multi-agent systems and control. To date, the structure of flocking has been predominantly studied through the application of artificial potential fields.

By the below graph working of this project can be analyse and get an optimal result that can be further implemented to various development of same project by adding more specifications on it. Also this can provide visual analysis of the system which perform the required task.

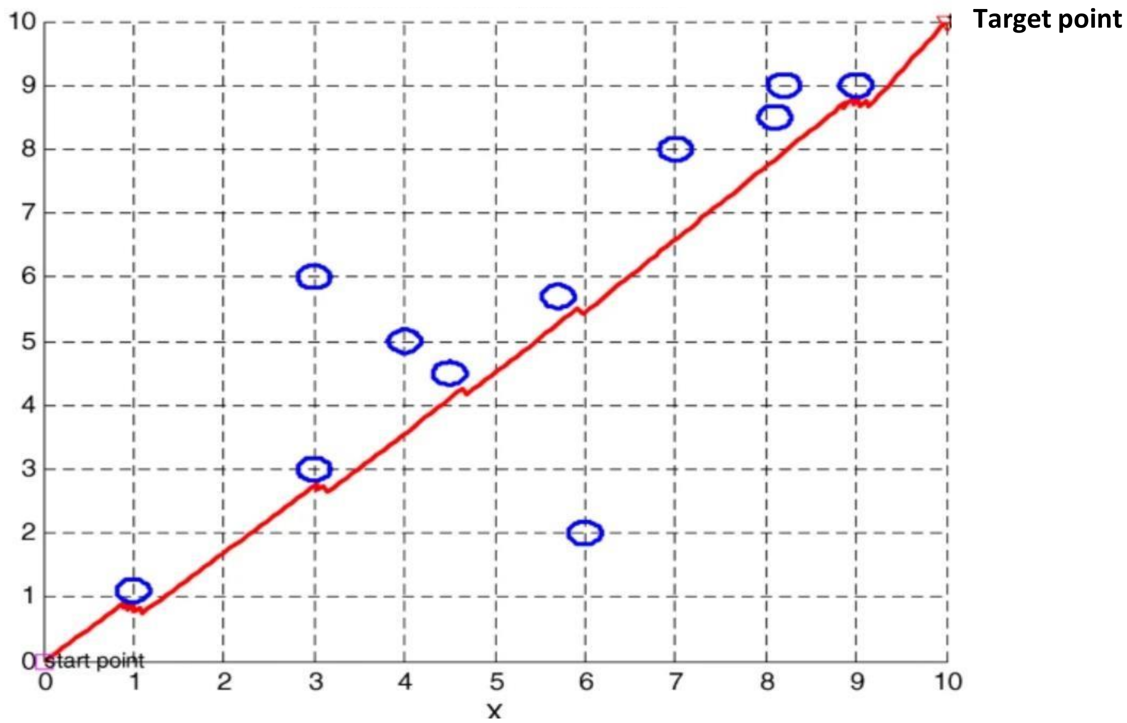

FIG 5.2

In the artificial potential field algorithm in which its relation was said in Section 2, if the value to the parameter be considered as Z = 100, M = 2, kS = 1.1, G0 = 0.11, and L = 0.1 and obstacles on a circle of the radius R = 0.18, the result is as shown in FIG 4.6; the robot is trapped and stands at the local minimum.

FIG 4.6 shows that the repulsive force is more than the attraction at all stages, which means the robot is trapped at the local minimum. If these parameters are changed to any desired extent other than these values, there will be no change to the status of the robot at the local minimum. The mesh plot of this simulation as shown in FIG 4.6. It describes the obstacles like summit, target act cavity and absorb the robot inside itself.

5.2 Efficient Algorithm

Obstacle avoidance is one of the prime issues related to autonomous navigation of mobile robots. It is not a viable option to model entire environment in which robot moves. In order to maneuver in dynamic environment robot has to be equipped with obstacle avoidance algorithm to deal with hurdles which are not known beforehand.

Bug Algorithms

Bug 1 algorithm is the simplest of all algorithms discussed in this work. It reaches to the goal all most all the times giving high reliability. But the matter of concern with this method is efficiency.

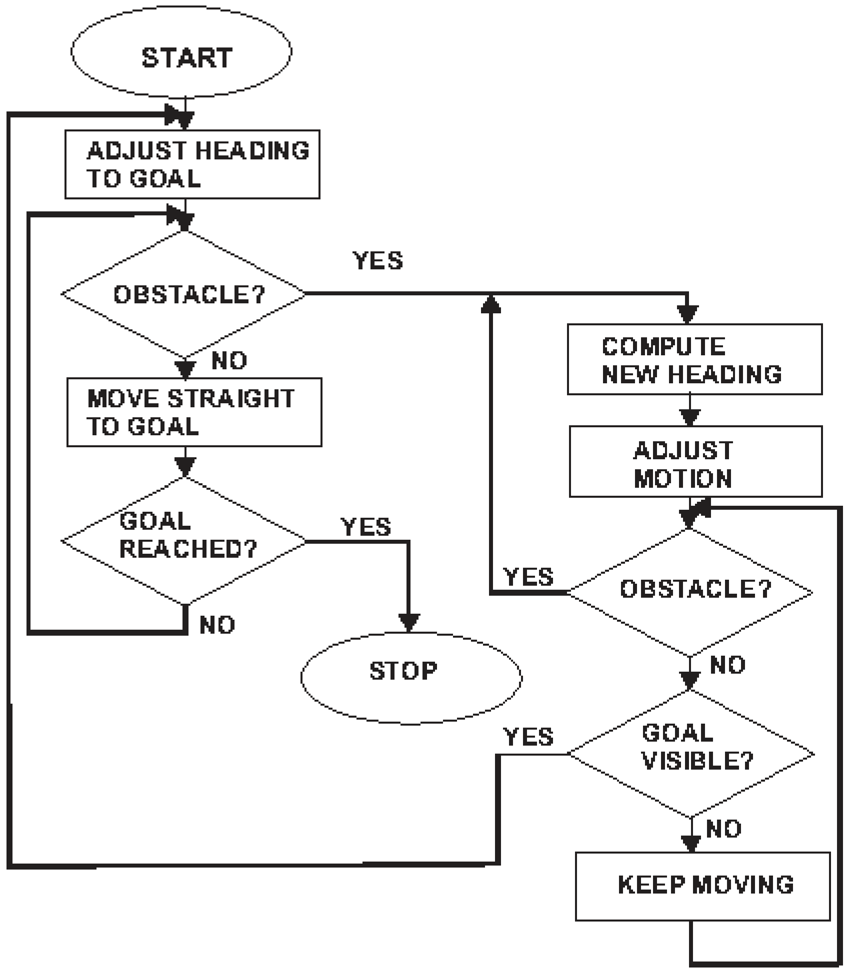

FIG 5.3

Robot moves on the shortest path joining robot's position X and goal location until it encounters a hurdle in the path. When it is confronted by an obstacle, it starts revolving around its surface and calculates distance from destination point. After one complete revolution it figure outs the point of departure which is closest to goal.

It maintains or changes direction of motion depending on distance of leaving point from hit point. This method can be illustrated in three steps

- Head towards the goal
- If an obstacle is encountered, circumnavigate it and remember how close you get to the goal
- Return to that closest point and continue Robot revolves around each and every obstacle in the way towards goal increasing computational efforts.

But ease of implementation makes it worth when only completion of task is required irrespective of time.

Bug 2 algorithm is modified form of Bug 1 algorithm. Instead of searching for minimum distance point, Bug 2 algorithm focuses on maintaining direction of motion towards goal. It calculates slope of the line joining initial point and desired point. When robot encounters an obstacle, it starts moving along the edge of the obstacle until it finds a point with the same slope. It starts moving on the line joining point of departure and goal. This algorithm is easy to program and provides better performance than Bug 1 algorithm in majority of cases. Robot does not need to encircle the entire Object as in the Bug 1 algorithm.

Distance Bug Algorithm

Robot travels comparatively shorter distance than earlier versions of Bug algorithm allowing the robot to reach destination in less time. This algorithm exhibit different obstacle avoidance behavior.

When confronted by an obstacle, it starts following the edge of the obstacle simultaneously calculating and storing the distance from its current and next position to destination.

The leaving point, where it switches the behavior from obstacle avoidance to move to goal, is selected based on the condition that the distance of destination from its next position is greater than the corresponding distance from its current position. The robot continues its obstacle avoidance behavior other-wise.

Tangent Bug Algorithm

Tangent Bug algorithm finds tangents to the obstacle and calculates distances of robot from points where they touch the obstacle. These points are denoted by Oi , where i denotes index number. In this approach robot takes path of the tangent which maximally decreases heuristic distance

I.e. D(robot position, Oi)+D(Oi, goal). At times it has to act as bug algorithm by following the edge of the obstacle if heuristic distance starts increasing while moving towards Oi.

It is a memory type algorithm which exhibits motion to goal and boundary following behaviour. A value Dmin which is the shortest distance observed so far between the sensed boundary of the obstacle and the goal and Dleave which is the shortest distance between any point in the currently sensed environment and the goal are continuously updated.

It terminates boundary following behaviour when Dleave < Dmin

5.3 DESIGN AN ALGORITHM

An intelligent mobile robot must reach the designated targets at a specified time. The robot must determine its location relative to its objectives at each step and issue a uitable strategy to achieve its target. Information about the environment is also needed to avoid obstacles and design optimal routes.

In general, the objectives of this study are improving the APF algorithm in order to trace he path and avoid obstacles by a mobile robot and comparing the performance and efficiency of the proposed algorithm with the previous method, APF algorithm.

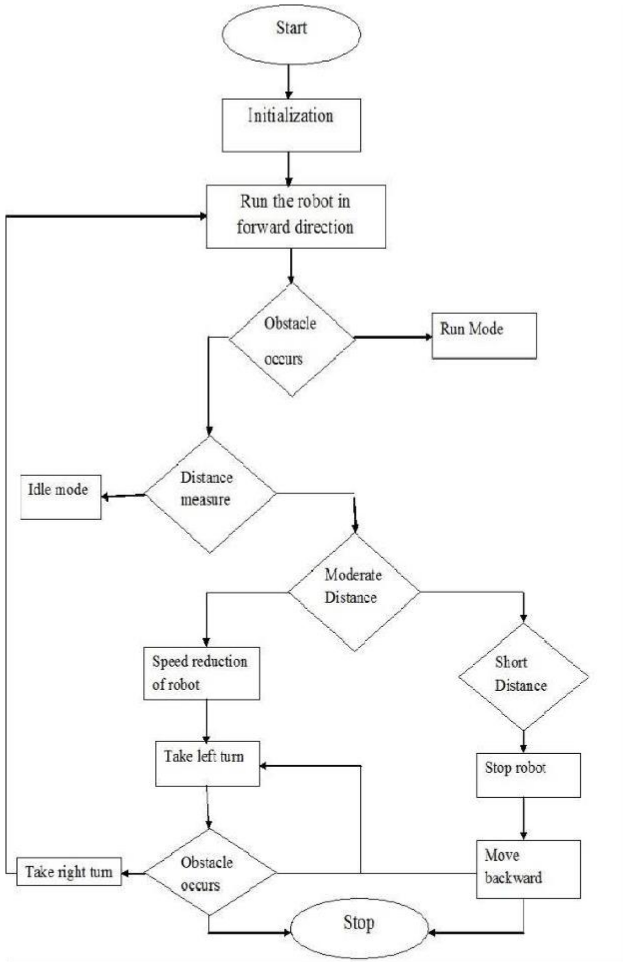

FIG 5.4

An intelligent mobile robot must reach the designated targets at a specified time. The robot must determine its location relative to its objectives at each step and issue a suitable strategy to achieve its target. Information about the environment is also needed to avoid obstacles and design optimal routes.

In general, the objectives of this study are improving the APF algorithm in order to trace the path and avoid obstacles by a mobile robot and comparing the performance and efficiency of the proposed algorithm with the previous method, APF algorithm.

CHAPTER 6

CODING

In Obstacle avoiding robot we place an ultrasonic sensor that can scan for the obstacles and free path. It can measure the distance from the obstacle and send the signals to the Arduino. There is servo motor which helps the sensor to look left and right.

Arduino code is written in C++ with an addition of special methods and functions, which we'll mention later on. C++ is a human-readable programming language. When you create a 'sketch' (the name given to Arduino code files), it is processed and compiled to machine language.

The Arduino Integrated Development Environment (IDE) is the main text editing program used for Arduino programming. It is where you'll be typing up your code before uploading it to the board you want to program. Arduino code is referred to as sketches.

6.1 ARDUINO IDE SOFTWARE

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus.

Writing code and uploading it to the board while offline is simple with the Arduino Software (IDE). For those with a bad or non-existent internet connection, we advise it. Any Arduino board can be used with this software.

The Arduino IDE is now available in two different versions: 1.x.x and 2.x. The IDE 2.x is a brand-new major update that outperforms the IDE 1.x.x in terms of speed and power. It also has more sophisticated capabilities to aid users in their coding and debugging in addition to a more contemporary editor and a responsive UI.

You can use either the offline IDE (IDE 1.x.x or IDE 2.x) by following the instructions below:

1. Get the Arduino Software IDE and install it:

IDE for Arduino 1.x.x (Windows, Mac OS, Linux, Portable IDE for Windows and Linux, ChromeOS).

2. IDE for Arduino 2.x 2. Connect your gadget to your Arduino board.

3. Launch the Arduino programme (IDE).

4. The text console shows all text output produced by the Arduino Software (IDE), including comprehensive error messages and other data.

The configured board and serial port are visible in the bottom right corner of the window.

Now that you are all set up, let's try to make your board blink!

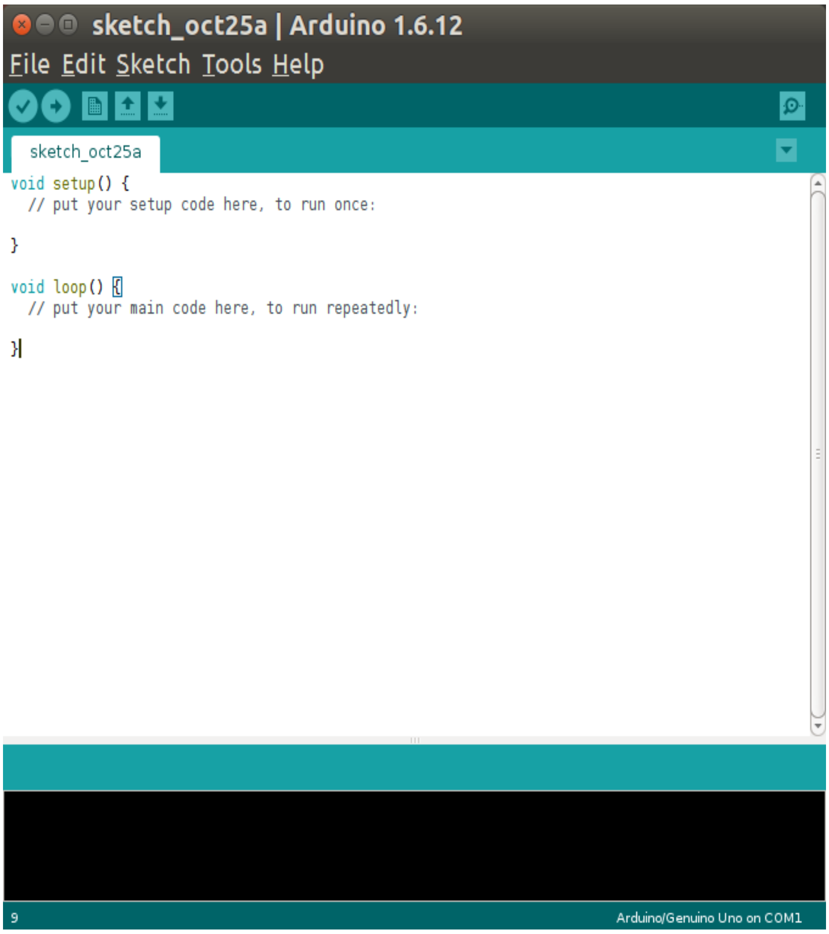

FIG 6.1

5. Connect your Arduino or Genuino board to your computer.

6. Now, you need to select the right core & board. This is done by navigating to Tools > Board > Arduino AVR Boards > Board. Make sure you select the board that you are using. If you cannot find your board, you can add it from Tools > Board > Boards Manager.

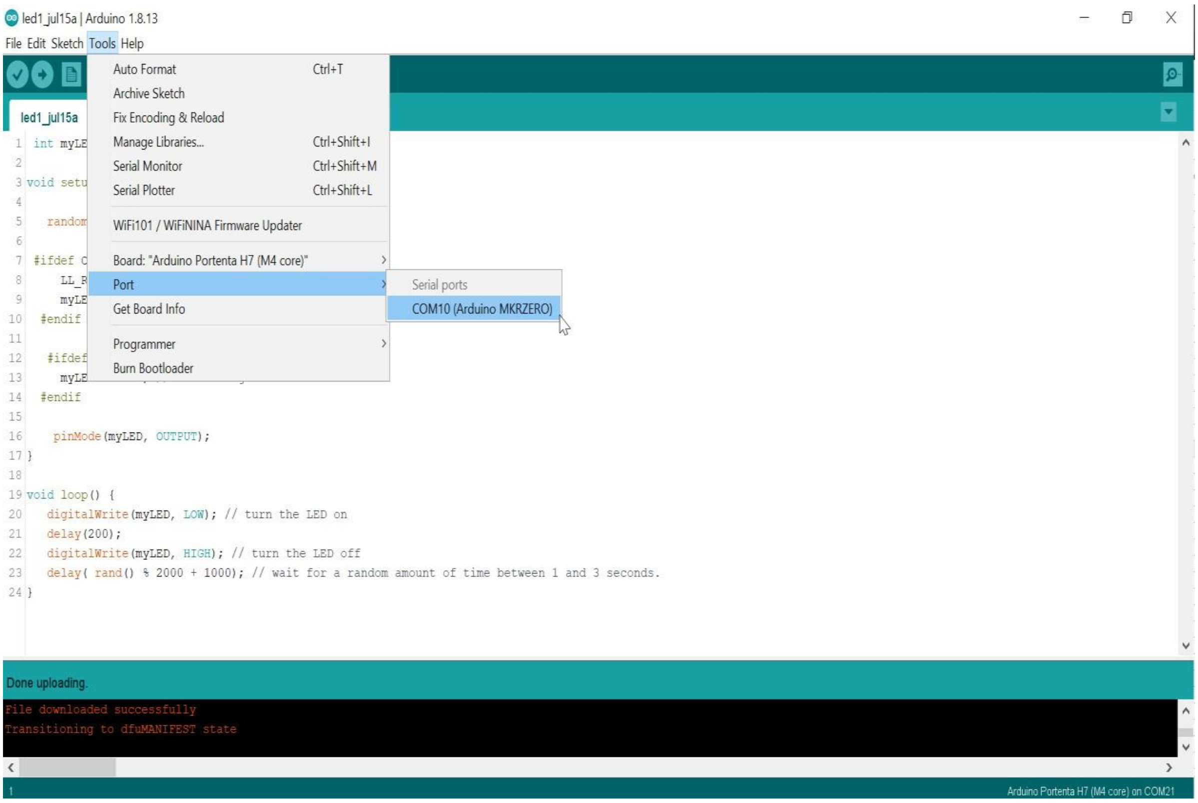

FIG 6.2

7. Now, let's make sure that your board is found by the computer, by selecting the port. This is simply done by navigating to Tools > Port, where you select your board from the list.

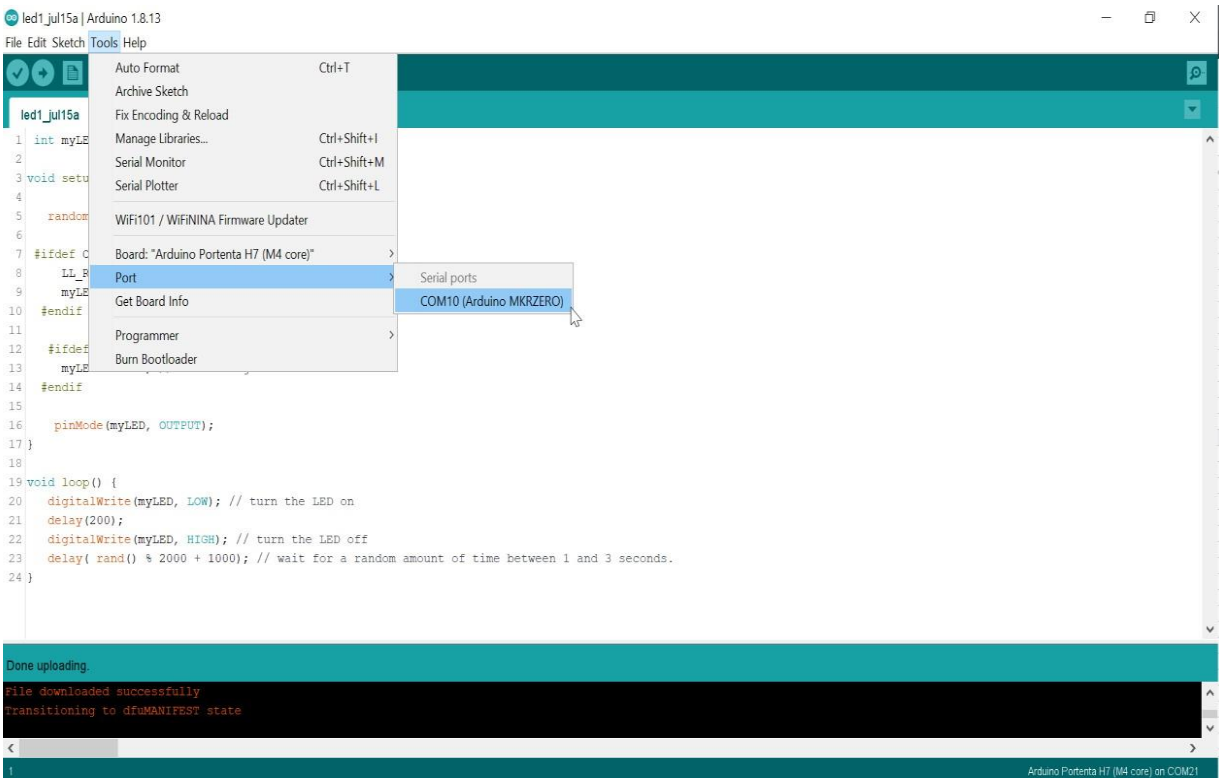

FIG 6.3

6.2 PROGRAMMING STRUCTURE

Arduino IDEs, and Installation process of the Arduino software. We learned that Arduino IDE

IDE (Integrated Development Environment) allows us to draw the sketch and upload it to the various Arduino boards Using code. The code is written in a simple programming language similar to C and C++.

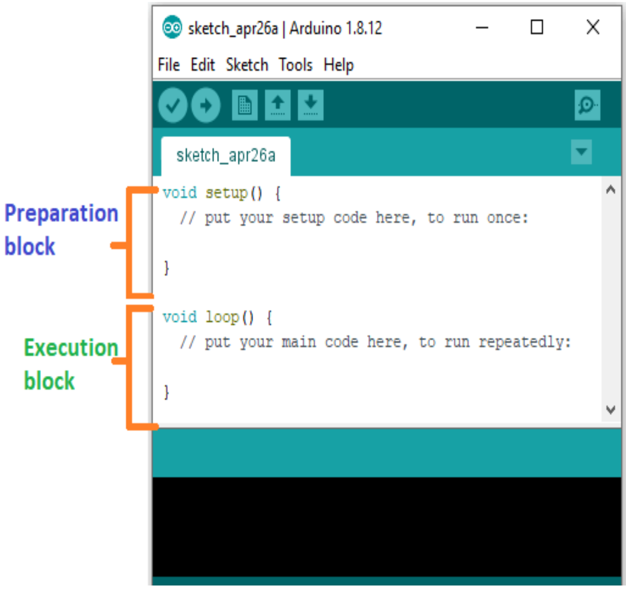

FIG 6.4

The coding screen is divided into two blocks. The setup is considered as the preparation block, while the loop is considered as the execution block

Let’s try an example: navigate to File > Examples > 01.Basics > Blink.

To upload it to your board, simply click on the arrow in the top left corner. This process takes a few seconds, and it is important to not disconnect the board during this process. If the upload is successful, the message "Done uploading" will appear in the bottom output area.

It contains an initial part of the code to be executed. The pin modes, libraries, variables, etc., are initialized in the setup section. It is executed only once during the uploading of the program and after reset or power up of the Arduino board.

Zero setup () resides at the top of each sketch. As soon as the program starts running, the code inside the curly bracket is executed in the setup and it executes only once.

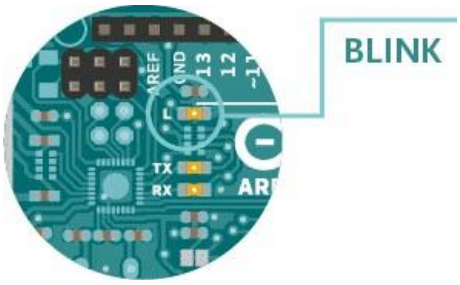

FIG 6.5

Once the upload is complete, you should then see on your board the yellow LED with an L next to it start blinking.

You can adjust the speed of blinking by changing the delay number in the parenthesis to 100, and upload the Blink sketch again. Now the LED should blink much faster.

The time in Arduino programming is measured in a millisecond.

Where, 1 sec = 1000 milliseconds

We can adjust the timing according to the milliseconds.

For example, for a 5-second delay, the time displayed will be 5000 milliseconds.

6.3 BASIC LEVEL OF PROGRAMMING

The Arduino programming language is based on a very simple hardware programming language called processing, which is similar to the C language. After the sketch is written in the Arduino IDE, it should be uploaded on the Arduino board for execution.

If you just got your Arduino Uno board, you'll first have to install the Arduino IDE (Integrated Development Environment) on another computer. The code is typed into the IDE and sent to the Arduino via a USB cable.

Visit arduino.cc to download the most recent Arduino IDE version for your computer. There are different versions for Mac, Windows, and Linux OS.

At the download page, click on the "Installer" option for the easiest installation then

1. Save the .exe file to your disk drive.

2. Open the .exe file.
3. Click the button to agree to the licensing agreement
4. Decide which components to put in, then click “Next”
5. Select which folder to put in the program to, then click “Install”
6. Wait for the program to complete installing, then click “Close”

BASIC PROGRAM

```
void setup ()
{
pinMode ( 13, OUTPUT);  // to set the OUTPUT mode of pin number 13.
}
void loop ()
{
digitalWrite (13, HIGH);
delay (4000);  // 4 seconds = 4 x 1000 milliseconds
digitalWrite (13, LOW);
delay (1500);  // 1.5 seconds = 1.5 x 1000 milliseconds
}
```

EXPLAINATION

1. Pinmode( )

The specific pin number is set as the INPUT or OUTPUT in the pinMode () function.
The Syntax is: pinMode (pin, mode)

2. digitalWrite( )

The digitalWrite ( ) function is used to set the value of a pin as HIGH or LOW.
Where,
HIGH: It sets the value of the voltage. For the 5V board, it will set the value of 5V, while for 3.3V, it will set the value of 3.3V.

LOW: It sets the value = 0 (GND).
If we do not set the pinMode as OUTPUT, the LED may light dim.

The syntax is: digitalWrite(pin, value HIGH/LOW)

3. delay ( )

The delay () function is a blocking function to pause a program from doing a task during the specified duration in milliseconds.
For example, - delay (2000)

6.4 OBSTACLE AVOIDANCE PROGRAMMING

Programming Arduino for Obstacle Avoiding Robot. First define trig and echo pin of HC-SR04 in the program. In this project the trig pin is connected to GPIO9 and echo pin is connected to GPIO10 of Arduino NANO. Define pins for input of LM298N Motor Driver Module.

It has following predefined function for robot motion-

1. forward() : forward movement of robot.

2. backward() : backward movement of robot.

3. turn_left() : for turning left.

4. turn_right(): for turning right.

5. halt() : for stopping robot.

STEPS TO IMPLEMENT THE OBSTACLE AVOIDANCE CODING

1. Open the new Arduino IDE page.

2. Delete everything on the page.

3. Make a code using above algorithms and required functions.

4. Paste empty Arduino IDE page.

5. First Install the Arduino Library.

6. The AF_Motor Arduino library.

Before you can use the Motor shield, you must install the AF_Motor Arduino library. Download the library which can implemented through tools option and install it. Check that inside AFMotor is AFMotor.cpp and AFMotor.h files

Place the AFMotor folder into your arduinosketchfolder/libraries folder. For Windows, this will probably be something like MY Documents/Arduino/libraries for Mac it will be something likeDocuments/arduino/libraries.

If this is the first time you are installing a library, you'll need to create the libraries folder. Make sure to call it libraries exactly, no caps, no other name. Rename the uncompressed folder AFMotor

7. The NewPing Arduino Library

Download this library function with the same procedure for AFMotor. Move to the Arduino library folder

8. Restart the IDE

9. Now you can see the libraries you have installed

10. When you verify, you will not see any errors

Using DC Motors with the Motor Shield

The motor shield can drive up to 4 DC motors bi-directionally. That means they can be driven forwards and backwards. To connect a motor, simply solder two wires to the terminals and then connect them to either the M1, M2, M3, or M4.

Make sure you #include AFMotor.h

Create the AF_DCMotor object with AF_DCMotor(motor#, frequency), to setup the motor H-bridge and latches. The constructor takes two arguments.

1. The first is which port the motor is connected to, 1, 2, 3 or 4. frequency is how fast the speed controlling signal is.

2. For motors 1 and 2 you can choose MOTOR12_64KHZ, MOTOR12_8KHZ, MOTOR12_2KHZ, orMOTOR12_1KHZ. A high speed like 64KHz wont be audible but a low speed like 1KHz will use less power. Motors 3 & 4 are only possible to run at 1KHz and will ignore any setting given.

Then you can set the speed of the motor using setSpeed(speed) where the speed ranges from 0 (stopped) to 255 (full speed). You can set the speed whenever you want.To run the motor, call run(direction) where direction is FORWARD, BACKWARD or RELEASE. Of course, the Arduino doesn't actually know if the motor is 'forward' or 'backward', so if you want to change which way it thinks is forward, simply swap the two wires from the motor to the shield.

Using the HC-SR04 with the NewPing Library

NewPing sonar(trigger_pin, echo_pin [, max_cm_distance]);

Example; NewPing sonar(12, 11, 200);

This initializes NewPing to use pin 12 for trigger output, pin 11 for echo input, with a maximum ping distance of 200cm.

max_cm_distance is optional [default = 500cm]. If connecting using a single pin, specify the same pin for both trigger_pin and echo_pin as the same pin is doing both functions.

CHAPTER 7

RESULT

7.1 EXPECTED OUTPUT

The naive robot starts with default behaviour, i.e., it drives straight ahead. When an obstacle is put in its way, the robot hits it and the reflex triggers a backward movement. The exact direction of this retreat depends on which tactile sensors have been touched.

When hitting obstacles, the robot learns coincidences of short-term memory representations and unconditioned stimuli. After about 700 time steps (about 80 s of robot action), the robot was generally able to avoid obstacles of high contrast relative to the background.

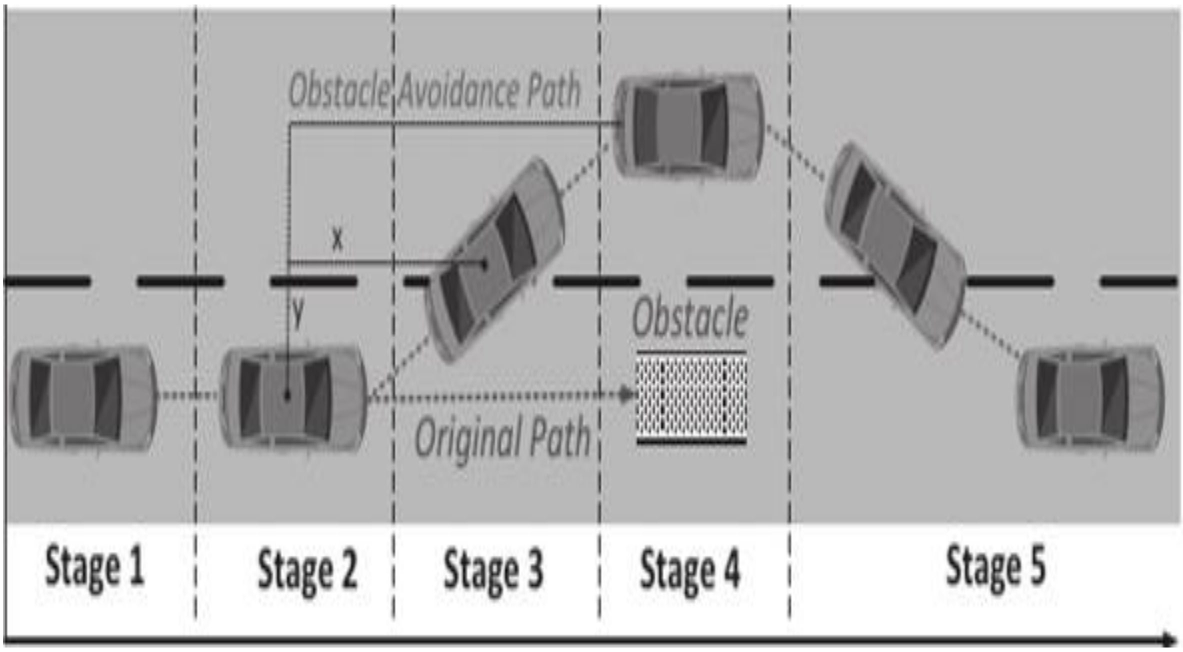

FIG 7.1

However, the robot could not detect the obstacle if the contrast edge was presented exactly in the middle of the visual field and the robot steered straight toward the obstacle so that the edge was kept always in the middle. In this case there is no optical flow to be measured and no CS signal can be used to anticipate the reflex movements.

Using Ultrasonic Sensor can avoid this problem. Most runs showed that middle EMD representations are suppressed up to less than 40% of their initial weight (“hippocampal synapses”), which is due to a rare occurrence of these stimuli representations in coincidence with collisions. Thus, the network correctly learned to pay less attention to irrelevant stimuli.

7.2 PROJECT OUTPUT

Major task of this project is to make and Arduino robot using ultrasonic and AF Driver to sense obstacle and move the robot. After Sensing an obstacle it will react over it by sensing other directions and move to an optimal way.

An obstacle avoidance robot vehicle made by using an ultrasonic sensor for its movement. A micro-controller (AT mega 8) is used to achieve the desired operation.

A robot is a machine that can perform task automatically or with guidance. Robotics is a combination of computational intelligence and physical machines (motors). Computational intelligence performs the programmed instructions.

The obstacle avoiding robot directs itself whenever an obstacle comes in its path. This robotic vehicle is built, using a micro-controller of AT mega 8 family. An ultrasonic sensor is used to finds any obstacle ahead of it and sends a command to the microcontroller.

Depending on the input signal received, the micro-controller change the direction of the robot by actuating the motors which are interfaced to it through amotor drive.

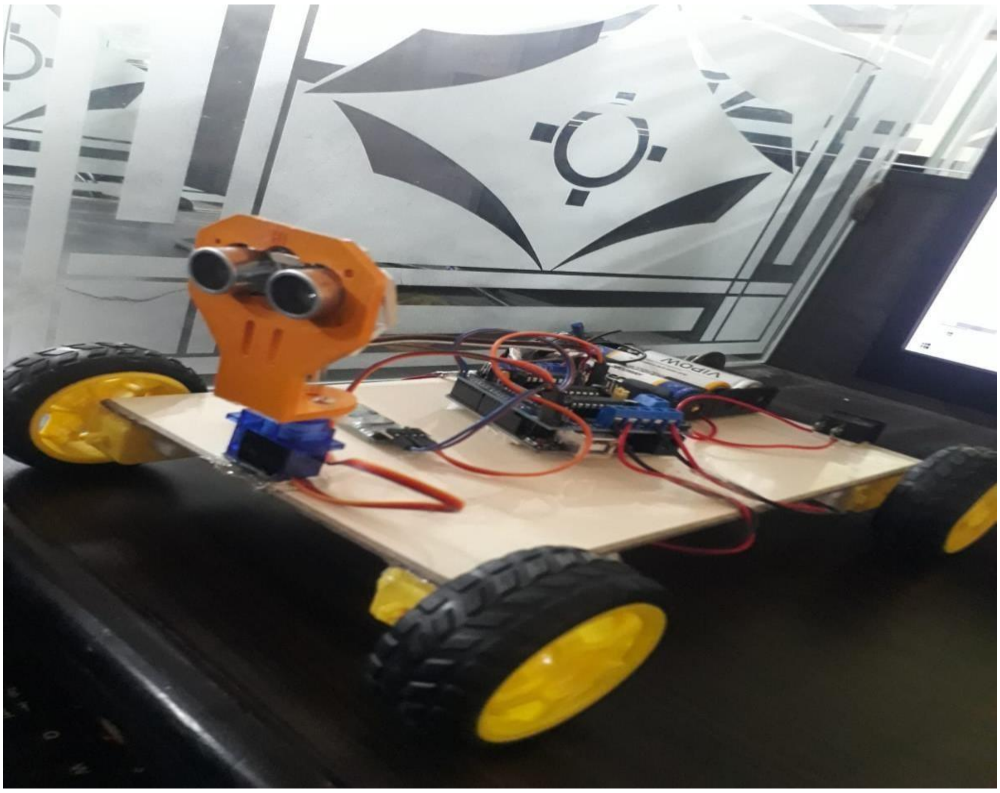

FIG 7.2

Above FIG 7.2, expressing the completion of task that need to perform obstacle avoidance voice reorganisation.

To initiate the rover we use mobile phone to say a command “START” run.

Steering algorithm is used for robotic actions in which driver or a human being is controlling the robot using remote. Here driver is present, who can see the obstacle and navigate robot accordingly.

The project proposes an autonomous robotic vehicle, in which no remote is used for controlling the robotic actions. It intelligently detects obstacles present on its path through the sensors, avoid it and take decision on the basis of internal code that we set for it.

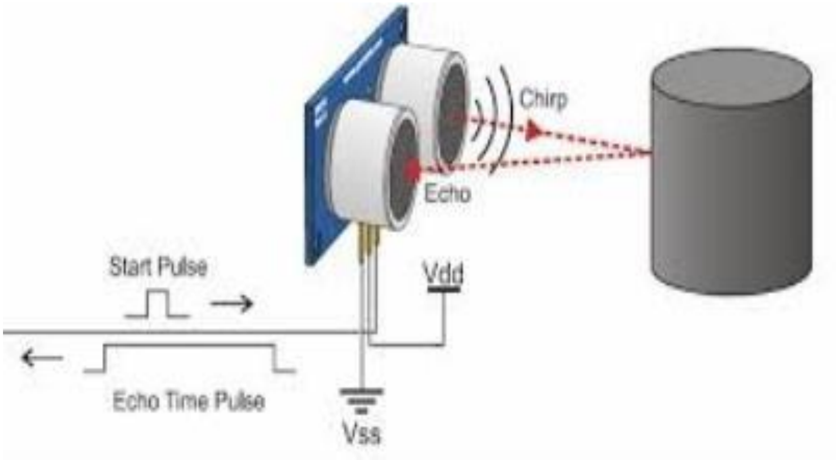

FIG 7.3

This technique of obstacle avoidance makes the robot to:

- Detect obstacles
- Avoid obstacles

The technique that is adopted in the project is that when robot detects an object at 30cm or less, it moves backward, and checks again for the obstacle with the delay of 15 micro seconds. If the obstacle distance is again less than 30cm then it turns left, otherwise keep moving in forward direction.

7.3 FUTURE UPDATE

Further the project will upgraded to recognise the voice command provided by the user to control the robot. The major task can be performed by this rover to work under major projects in future.

To safely avoid dynamic obstacles we rely on a motion planner that generates a set of candidate actions for the vehicle and selects from this set one that is collision-free with respect to these obstacles.

In our implementation each of these actions is a dynamically-feasible trajectory that can be directly executed by the vehicle. The length of these trajectories varies based on the current speed of our vehicle and is designed to ensure the vehicle could, if necessary, come to a stop over the course of the trajectory.

These trajectories are generated using a model-based trajectory generator developed by Howard and Kelly that incorporates a high-fidelity vehicle model to produce an accurate prediction of the vehicle's movement as it executes the trajectory.

We can then use this prediction along with our dynamic obstacle predictions to determine whether a candidate trajectory for our vehicle will cause a future collision with any of the dynamic obstacles.

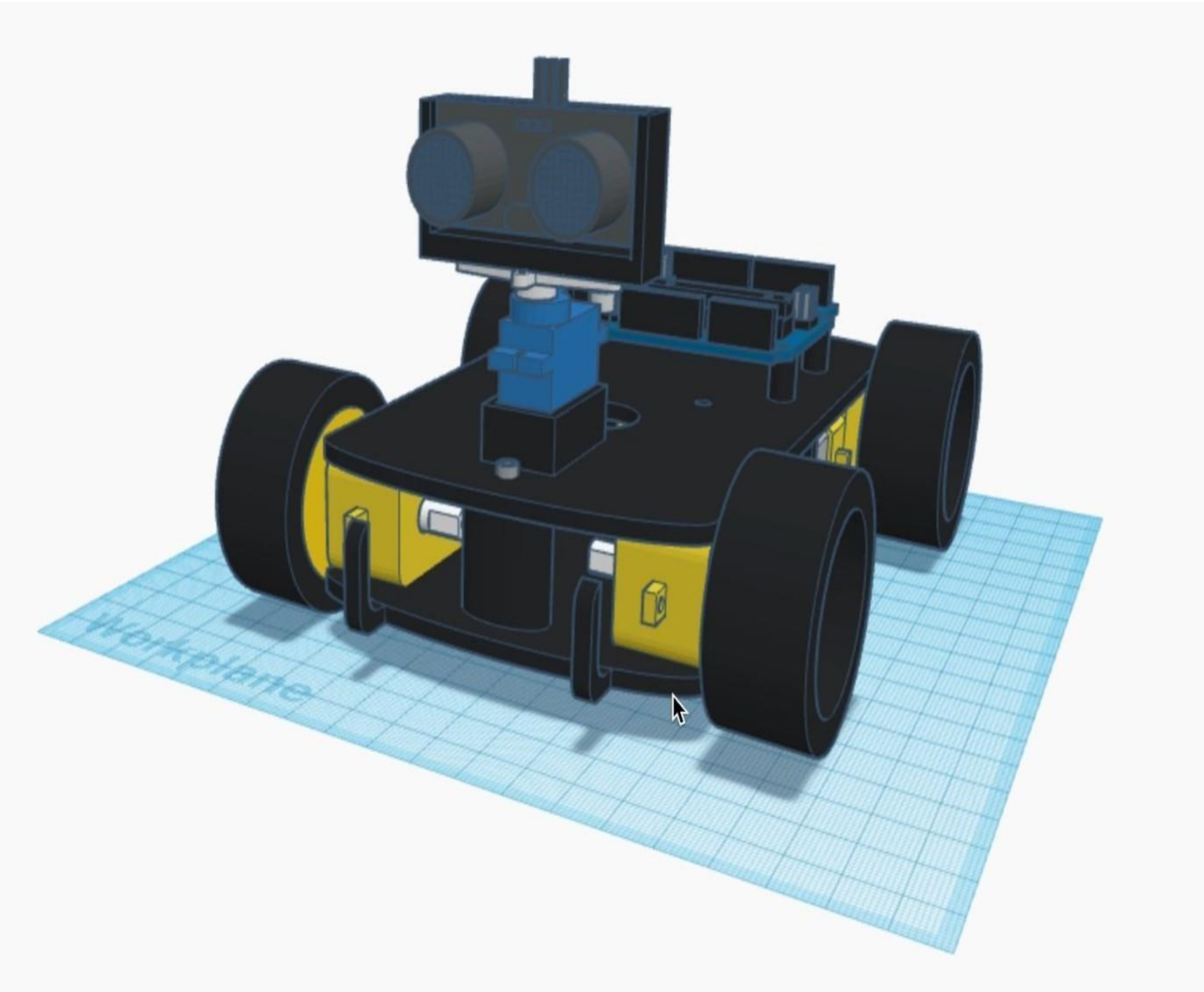

FIG 7.4

The obstacle detection algorithm was run off-line and the detected position was added manually to the switched guidance and control system described in FIG 7.2.

However, the algorithm also has the potential to be fully autonomous as part of the online control system; the detection algorithm is sufficiently fast that runtime will not be a concern in an online implementation.

The implementation of the necessary communication and control structure required to achieve this is a topic of future work. However, in the presented results the obstacle detection algorithm was run and the detected position added to the control system in one operation without removing the robot from the pool or turning it off.

7.4 MODIFICATIONS

In recent years, topics related to robotics have become one of the researching fields. In the meantime, intelligent mobile robots have great acceptance, but the control and navigation of these devices are very difficult, and the lack of dealing with fixed obstacles and avoiding them, due to safe and secure routing, is the basic requirement of these systems.

As this project can further implemented by adding camera to detect the type of object and implement different code to make a task. These task has different features to regulate the performance of the robot. It is a miniature of an automation robot to perform various task needed to be implemented to a user.

FIG 7.5

The modified artificial potential field (APF) method is proposed for that robot avoids collision with fixed obstacles and reaches the target in an optimal path; using this algorithm, the robot can run to the target in optimal environments without any problems by avoiding obstacles, and also using this algorithm, unlike the APF algorithm, the robot does not get stuck in the local minimum.

Project is looking for an appropriate cost function, with restrictions that we have, and the goal is to avoid obstacles, achieve the target, and do not stop the robot inlocal minimum.

The previous method, APF algorithm, has advantages, such as the use of a simple math model, which is easy to understand and implement.

However, this algorithm has many drawbacks; the major drawback of this problem is at the local minimum and the inaccessibility of the target when the obstacles are in the vicinity of the target. Therefore, in order to obtain a better result and to improve the shortcomings of the APF algorithm, this algorithm needs to be improved.

Here, the obstacle avoidance planning algorithm is proposed based on the improvement of the artificial potential field algorithm to solve this local minimum problem.

In the end, simulation results are evaluated using MATLAB software. The simulation results show that the proposed method is superior to the existing solution. Which may use to improve quality of services need to perform by the machine.

CHAPTER 8

CONCLUSION

This robot is able to produce the basic walking movements using two gear motors. They developed the robot with a very good intelligence which is easily capable to sense the obstacle and by processing the signal coming from the sensor it is perfectly avoiding the obstacle coming in the path. Robot takes left right or forward backward movement of the robot smooth. Ultrasonic sensor is better to detect obstacle coming in between the in the path.

An obstacle detection circuit was successfully implemented using infrared and ultrasonic sensors modules which were placed at the front of the robot to throw both light and sound waves at any obstacle and when a reflection is received, a low output is sent to the Arduino microcontroller which interprets the output and makes the robot to stop.

8.1 ACHIEVEMENTS

The obstacle detection and avoidance robot now successfully detects and obstacle and then turns right to avoid the obstacle, enough though it comes across an obstacle then it would completely come to a halt.

This robot was completed as it was proposed in the class earlier along with few simple modifications like using RGB LEDs instead of simple IR LEDs and also adding a buzzer and adjusting the volume depending on the distance of the obstacle and its interfacing with MSP430 was one of the challenges faced by the team along with the integration of Ultrasonic sensors.

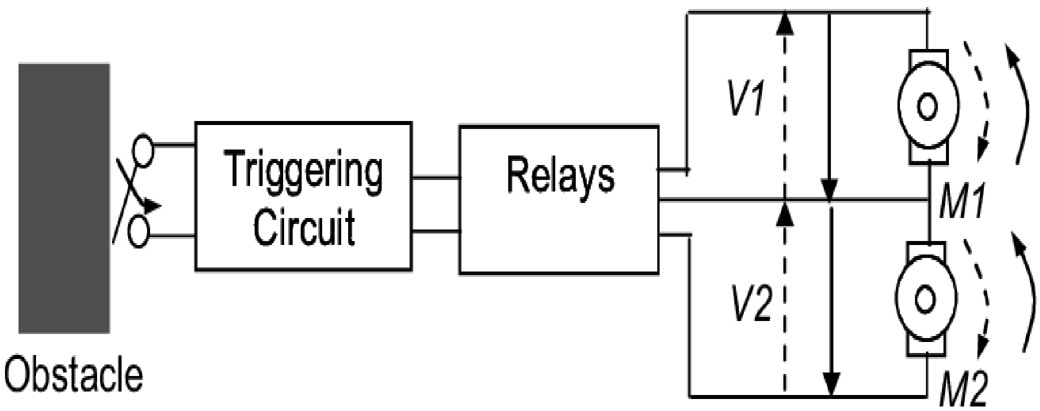

FIG 8.1

Development of an obstacle avoiding robot model is required as a fundamental step towards a bigger goal, for example development of an autonomous vehicle. An obstacle avoiding robot uses a proximity sensor module, besides other parts.

In this case, this robot uses a proximity sensor developed by ourselves. The robot is controlled by a program that is embedded into a microcontroller. The logics produced by the microcontroller are further processed by an interface module, in this case, also developed by ourselves.

The interface module translates microcontroller's logics into voltage and current that can practically drive the two motors. This article provides a report on the project activity, consisting of summary of the design, summary of the development process and report on the running test of the robot.

Following the test and program fine-tuning, it has been proven that the robot model operated well just as programmed. Index Terms— obstacle avoiding, obstacle avoidance, proximity sensor, mobile robot, interface module for robot

8.2 FUTURE DEVELOPMENT

Obstacle detection using Ultrasonic sensor can be designed even more sophisticated by using touch pads, where the robot can start with a touch or voice detector where voice commands can be set and make them function according to the command. Or may be an android application can be developed in order to control the functioning of the robot using our phone.

The working with msp430 and interfacing all the components, using the resources to the maximum extent was a bit challenging. Interfacing UART and PWM and ADC was something very helpful.

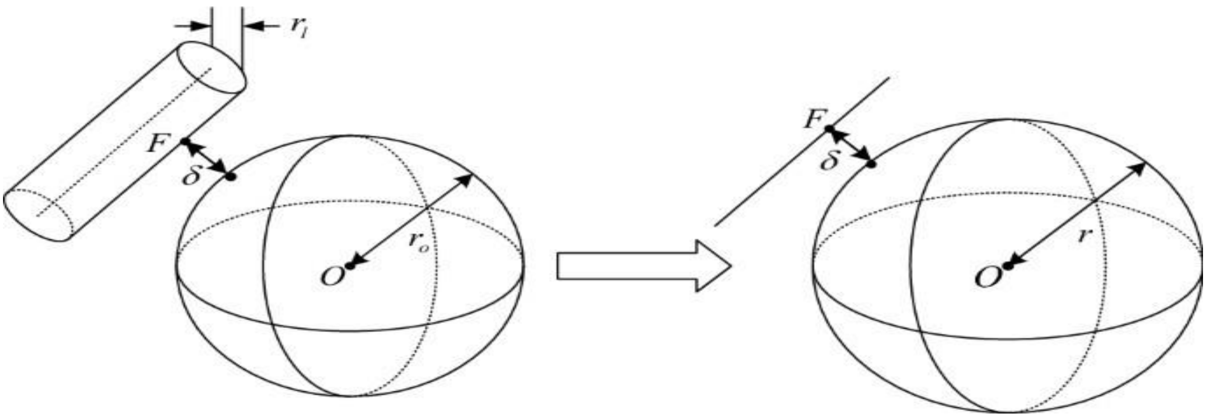

FIG 8.2

Soldering and wiring was the other thing learned, it was fun working with all those instruments. Developing code where all the other components following the inputs of the controller was difficult at first, took more time, but finally accomplished the task.

This FIG 8.2 represent the physics law that can be implemented in project to make the device to be more optimal also provide more interactive to people who handle this system.

8.3 APPLICATION

In the future, the sensing can be increased. A Bluetooth module and a camera can be attached, so that the user can see the obstacle and take pictures and also can take videos of it. As the sensor can detect only the obstacles with reflective surface, so in future work can be carried out to detect and avoid obstacles of absorbing surface.

Obstacle avoiding robots can be used in almost all mobile robot navigation systems. They can be used for household work like automatic vacuum cleaning. They can also be used in dangerous environments, where human penetration could be fatal.

To develop an obstacle avoidance robot using proximity sensors. The robot is constructed to look small enough with light-weight material. The sensor is placed in a lower position. So that the robot can sense the smaller obstacles in an efficient way.

To sense the obstacle, three proximity sensors are used. Generally, a proximity sensor senses or detects the presence of the obstacle by emitting the electromagnetic radiation and it will get reflected back from the obstacle to the receiver. Based on this, the distance will be calculated. The robot is designed with two DC motors at the rear end and a ball-shaped bearing is placed in the front. Based on the force produced by the motorized wheels, the robot will move.

Then the ATmega328P microcontroller is used for the robot controller. This system uses two lithium-ion batteries of 4.2V. And the code for the motor rotation and the whole robot operation procedure is written in the C++ language with Arduino compiler.

The design of the robot can also be developed by using assumption based architecture, a reactive control system for real time obstacle avoidance vehicles. Generally, the reactive robot is designed to explore the unknown environment. Based on the sensor feedback, the robot can move in an unknown environment.

Direct detection and avoidance methods include proximity sensor, tilt sensor, gyroscope, accelerometer which gives the response when the robot is in contact with obstacles. Indirect detection and avoidance method is a common method for obstacle detection and avoidance robots because of the range of detection. Sensors like proximity, infra-red sensor, ultrasonic sensors are used for this method. Obstacle detection and avoidance is possible by either autonomous or remote control.

By using an android application, the robot can be remotely controlled. The data will be sent from the application and it will be received by the Arduino with the Bluetooth module. For the obstacle avoidance robot, the controller is an important module.

For multi purposes like pick and place, obstacle detection and avoidance and higher end applications, ARM microcontroller can be used. To sense the presence of the obstacles, IR and PIR sensors are also used but there is a limitation to use these sensors.

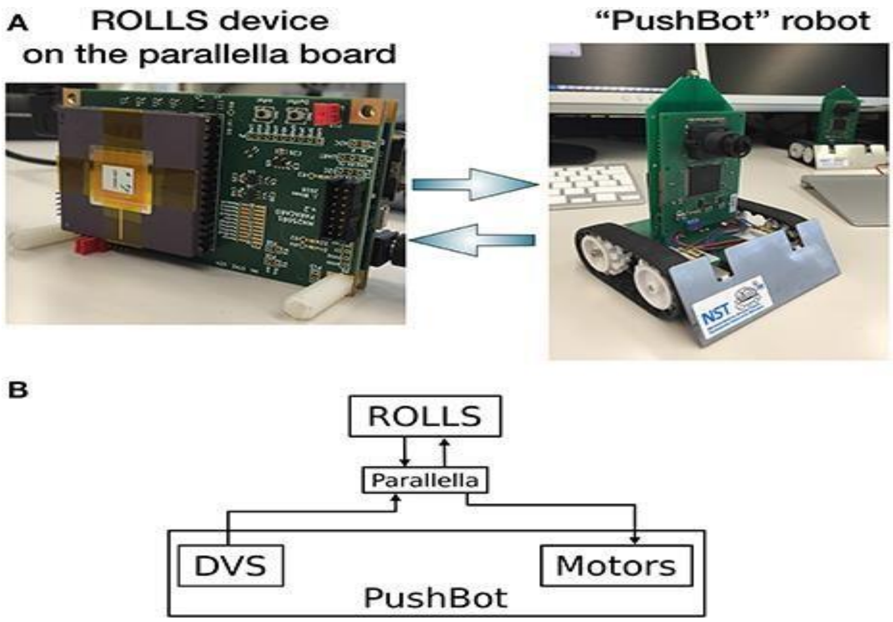

FIG 8.3

Obstacle avoidance vehicles have been very helpful for handicapped people. The vehicle can be controlled through voice-command input which is stored in the application. By interfacing the Bluetooth module with the controller, this process works. Suppose, if the obstacle robots are used in military applications, the robot does not know which direction to navigate.

By using a digital compass and sending the information to the controller, the robot can make movement in the correct direction. In real-time the obstacle detection and avoidance is a serious issue. By using "The Bug algorithm", the robot can be able to find the smallest distance to reach the destination.

REFERENCES

1. J. Seja and M. Banshidhar. 2013. Obstacle detection and avoidance by a mobile robot. National Institute of Technology, Rourkela. B.Sc. thesis. pp. 1-9.

2. L. Navarro-Serment, C. Paredis and P. Khosla. 1999. A beacon system for the localization of distributed robotic teams. Proceedings of the International Conference on Field and Service Robotics. pp. 232-237.

3. T. Bailey, E. Nebot, J. Rosenblatt and H DurrantWhyte. 1999. Robust distinctive place recognition for topological maps. Proceedings of the International Conference on Field and Service Robotics. pp. 347-352.

4. N. Harper and P. McKerrow. 1999. Detecting plants for landmarks with ultrasonic sensing. Proceedings of the International Conference on Field and Service Robotics, pp. 144-149, 1999.

5. Sanjyoti Jain, Dr. Isha Malhotra, “A review on Obstacle Avoidance techniques on Self-Driving Vehicle”, International Journal of Advanced Science and Technology, vol. 26, no. 3, pp. 5159-5167, January, 2020