International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

# Impact of Adaptive Cruise Control System in Automobiles

***Dr. A. Vijin Prabhu[1], B. Chandra Subash[2], J. Sai Kumar[3], D. Ramesh[4], B. Pavan Kumar[5], A. Kavya Sai[6]***

[1,2,3,4,5,6]Department of Mechanical Engineering, GMRIT, Andhra Pradesh, India

**ABSTRACT:**

The main concern of introducing the autonomous technology is for safety purpose. According to a survey 90% of accidents happen due to mistake of driver. In this project Adaptive cruise control feature was implemented on a prototype using ultrasonic sensor and Arduino. The adaptive cruise control system (ACC) is a system which combines cruise control with a collision avoidance system. This system can control the velocity of vehicle automatically to match the velocity of car, bus or truck in front of vehicle. If the lead vehicle gets slow down or accelerate, then ACC system automatically matches that velocity. This project focuses primarily on two tasks: the first is to implement adaptive cruise control at stop-and-go situations while the second is to maintain constant speed in accordance with the previous vehicle at a distance that the driver can adjust to his comfort. The circuit was created in Proteus 8 software, and the Arduino coding was carried out using the Arduino IDE software.

**Keywords:** Adaptive cruise control, Arduino IDE, Autonomous vehicles, Ultrasonic sensor, Arduino, Proteus 8

## 1. Introduction:

As companies strive to improve their production methods and incorporate automation technology, intelligent vehicles and related products have become essential components of automated transportation systems and flexible production organizations. Countries worldwide are actively researching the design of intelligent vehicles, which require the ability to automatically detect and avoid obstacles and measure distances to ensure timely responses. Intelligent vehicles, also known as unmanned vehicles, incorporate functions such as environment perception, planning, decision-making, and multi-scale auxiliary driving (lang et al.).

For ACC systems to be acceptable to drivers and passengers, they must be designed to provide smooth and natural vehicle behavior that resembles the driving of human drivers. The longitudinal control algorithm of the vehicle is created to incorporate the driving habits of human drivers and to achieve a natural driving experience. As ADAS systems always operate alongside human drivers, the ACC/CA system with collision avoidance and mitigation braking can be helpful to drivers, and its system control logic and operation characteristics should resemble those of a human driver's decision-making and driving operation. (Yi, Kyongsu, et al). New studies on natural driving behaviours using SAE Level 2 automation indicate that factors such as road conditions, traffic patterns, weather, and driver characteristics significantly affect how automation is utilized. Drivers tend to maintain a larger following distance when driving at a consistent speed with adaptive cruise control (ACC) activated, but they tend to shorten this distance when accelerating or decelerating. (varotoo et al.).

Current approaches usually follow a three-layer structure, similar to traditional robotics solutions. The first layer is responsible for perceiving the environment, with a particular emphasis on detecting and predicting objects, as well as detecting lanes and calculating available space. The second layer focuses on decision-making, which includes strategic, tactical, and operational planning. The third layer, which comes after the decision-making layer, handles detailed movement planning, trajectory planning, and lateral and longitudinal vehicle dynamic control to achieve accurate movement tracking. (Schrödel et al..). Research conducted in the past has shown that the use of ACC results in reduced maximum speed, mean speed and headway variance, thus creating a safer driving environment. In addition, the variance in speed and headway is also reduced, thereby decreasing the likelihood of accidents. Furthermore, the provision of ACC results in drivers paying greater attention to lateral control, which in turn leads to a reduction in lane departures compared to when ACC is not available (Lan et al..). Results from previous experiments conducted on the road indicate that early ACC systems, which are only activated when driving at medium to high speeds, significantly impact the longitudinal control of drivers. It's worth noting, however, that these findings may be affected by the conditions under which the systems are activated, such as low-medium traffic density and non-critical traffic situations, as well as the inability of the systems to be used at lower speeds. Recently, full-range ACC systems have been installed in commercially available vehicles, which can operate at both low and high speeds in dense traffic conditions. (varotoo et al..). In this project ultrasonic sensor and Arduino are used to implement adaptive cruise control feature.

The development of intelligent vehicles has led to the utilization of various types of sensors (e.g. cameras, LIDAR, radars at 79 GHz or 24 GHz, ultrasonic sensors, etc.) for perceiving the traffic environment surrounding road vehicles. Out of these sensors, ultrasonic sensors have several advantages such as being cost-effective, having a wide detection angle, a small blind zone in the near-field, and being robust to varying environmental conditions (such as lighting, fog, etc.). Essentially, an ultrasonic sensor functions by using a transducer or transmitter to emit ultrasonic waves or pulses and then receiving the reflected waves to determine an object's proximity (stiawan et al.). The transmitter calculate the vehicle speed according to the time taken to transmit the ultrasonic waves (stiawan et al.). When waves hitting an object, part of their energy reflected to receiver of the sensor as echo signal (alessio et al.).

Arduino is a set of open-source microcontroller-based kits that can be used extensively in the digital world and for building interactive devices that can sense and control physical devices. This technology is based on a design of microcontroller boards, which can implement various functions. It includes digital and analog input/output (I/O) pins that allow interfacing with various external boards and circuits. Additionally, Arduino provides features such as Universal Serial Bus (USB) for loading programs from computer systems. The technology also comes with an integrated development environment (IDE), which is based on programming languages such as C and C++ (lang et al.).

## 2. Literature survey:

The ACC system has a direct impact on the longitudinal control of drivers as it maintains a target speed and time gap. Previous on-road experiments have indicated that early ACC systems, which are only activated when driving at medium to high speeds, significantly influence the longitudinal control of drivers. However, these results may be affected by the circumstances in which the systems are activated, such as low-medium traffic density and non-critical traffic situations, as well as the inability of the systems to be used at lower speeds. Recently, full-range ACC systems that can operate at both low and high speeds in dense traffic conditions have been installed in commercially available vehicles. Recent studies involving naturalistic driving with SAE Level 2 automation have revealed that automation usage is significantly impacted by road characteristics, traffic conditions, weather conditions, and driver characteristics. With ACC activated, drivers tend to maintain a larger time gap in stable speed conditions and a shorter time gap when accelerating and decelerating (varotoo et al.).

The results of the experiment demonstrated that time-gaps exceeding 1.60 seconds were deemed more acceptable and effective in preventing collisions. However, in situations where drivers were distracted by secondary tasks, the acceptable time gap increased to 2.08 seconds. This resulted in a significant difference of 0.48 seconds or 13.1 meters, which was considered as the time required for drivers to disengage from the secondary task. Additionally, it was observed that drivers maintained shorter minimum gaps while driving with ACC, but preferred time-gaps greater than 2.08 seconds when engaged in secondary tasks. In this study, regaining control was a demanding task that required high levels of visual and manual attention, as the driver had to observe the lead vehicle and react to emergency braking. The visual, auditory, and cognitive loads resulted in longer reaction times. Hence, the secondary tasks designed for the study had a significant impact on the results, which were obtained through simulations conducted on buses and may vary when applied to cars (Lan et al..).

Despite their benefits, it is important to note that current ACC systems have limitations, including their speed and acceleration range, as well as their braking ability. However, the next generation of ACC systems is being designed to operate in all speed ranges and in various traffic situations, including stop-and-go traffic. These new systems have the potential to actively prevent rear-end collisions, improving safety on the roads. However, it is important to keep in mind that ACC systems only control longitudinal driving and do not account for merging, lane changing, or creating gaps for other vehicles. As a result, the driver must still intervene when necessary and retain full responsibility for the vehicle's actions. Microscopic simulations have shown that using ACC vehicles on a road section with an on-ramp bottleneck during rush hour can improve traffic stability and increase the dynamic road capacity (kesting et al.).

However, it is important to note that the results of simulator studies may not always directly translate to real-world driving situations. Real-world factors such as weather conditions, road conditions, and driver behaviour can greatly impact the performance of both manual driving and ACC systems. Therefore, it is necessary to conduct further studies and experiments in real-world driving situations to fully understand the benefits and limitations of ACC systems compared to manual driving (Lang et al.).

The ability of a system to mimic human behavior is key to identifying a fully adaptive system, which can adjust not only to current traffic conditions but also to the driver's attitudes and preferences. Acceptance by users is critical not only for vehicle manufacturers, to increase market share, but also for society as a whole, to promote actual adoption of the system. The importance of fully adaptive systems has been widely acknowledged in scientific literature, and researchers have attempted to identify drivers' preferences using appropriate parameters. Building on these ideas, this study introduces significant advancements towards a human-like, fully adaptive ACC system, which relies on a linear car-following model to estimate and apply adjustments in real-time and on-demand, without requiring vehicle-to-vehicle communication. The system is designed to operate independently. (bifulco et al.).

Ultrasonic Sensors (UsS) can encounter difficulties because sound waves are affected by materials such as sponges, cotton, and irregular objects, as well as by environmental factors like wind, heat, and humidity. Additionally, objects come in many shapes, so this study compares the reflection of sound waves from cubic, cylinder, and cone shapes to determine which is best for reflecting the waves at different distances and sizes using the HC-RS04 sensor's properties. The module emits eight pulses at 40KHz and waits for a pulse to return. If a reflection is detected, the echo takes over these pulses, and the time taken from sending to receiving is divided by two, giving an approximate distance based on the sound velocity of the environment. However, using two ultrasonic sensors sequentially with a 15cm gap between them reduces the error rate, thereby increasing accuracy and efficiency by at least twice as much. (haider et al.).

This article introduces an ultrasonic sensor that measures the distance between specific points of a motor vehicle and the ground. The sensor operates by measuring the time of flight of an ultrasonic pulse reflected by the ground. The sensor is mounted at the rear of the vehicle, which is also fitted with four potentiometer sensors to measure the spring heights while in motion. A portable digital recorder captures the outputs from both the ultrasonic sensor and the potentiometers. The sensor output is updated every 20 milliseconds, and an additional digital output facilitates easy implementation of smoothing techniques via the car computing system. The obtained measurements are precise enough to level the headlights and provide important information for active suspension systems. (alessio et al..).

The Motor Driver IC L293D is an integrated circuit that provides dual H-bridge motor drivers. These drivers amplify current by taking a low current control signal and providing a higher-current signal to drive motors. L293D includes two built-in H-bridge driver circuits that can drive two DC motors simultaneously in both forward and reverse directions. When the enable input is high, the associated driver becomes active, and its outputs work in phase with the inputs. Conversely, when the enable input is low, that driver is disabled, and its outputs are in the high impedance state.(subodh et al..).

After receiving data from the HC-05 module, the MCU makes decisions based on the received information and controls the intelligent car to move forward, backward, turn left, and turn right. However, the ARDUINO cannot directly drive the DC motor and requires a motor driver chip. To meet the car's requirements, the L298N was chosen as the drive chip due to its ability to facilitate both positive and reverse rotation of the motor and adjust its speed through the use of PWM. (lang et al..).

This high amplitude pulse is used to brake the motor by stopping the flow of current through it. The reverse braking mode is used to quickly stop the rotation of the motor in the opposite direction. The speed control and braking of the DC motor using PWM technique is widely used in various applications such as robotics, automation, and industrial control systems. It provides precise control over the speed of the motor and also increases the efficiency and reliability of the system. The use of PWM technique in motor control also helps in reducing power consumption and minimizing heat dissipation, thereby increasing the life of the motor and the system as a whole. (subodh et al..).

## 3. Methodology:

Initially, some journals were collected from IEEE and Science direct to know more details about working and new developments of adaptive cruise control. After studying journals picking some key features from them is done and started working on those features and finally found one feature which is less concentrated in recent developments. Working on this feature has started and finally found a way to develop a prototype with this feature. In order to develop the prototype first research on components is done and then started writing code according to the components. Then after writing code, it is dumped into Arduino board and assembled all the components to prepare the prototype. After preparing the prototype, testing it under different scenarios has been done.

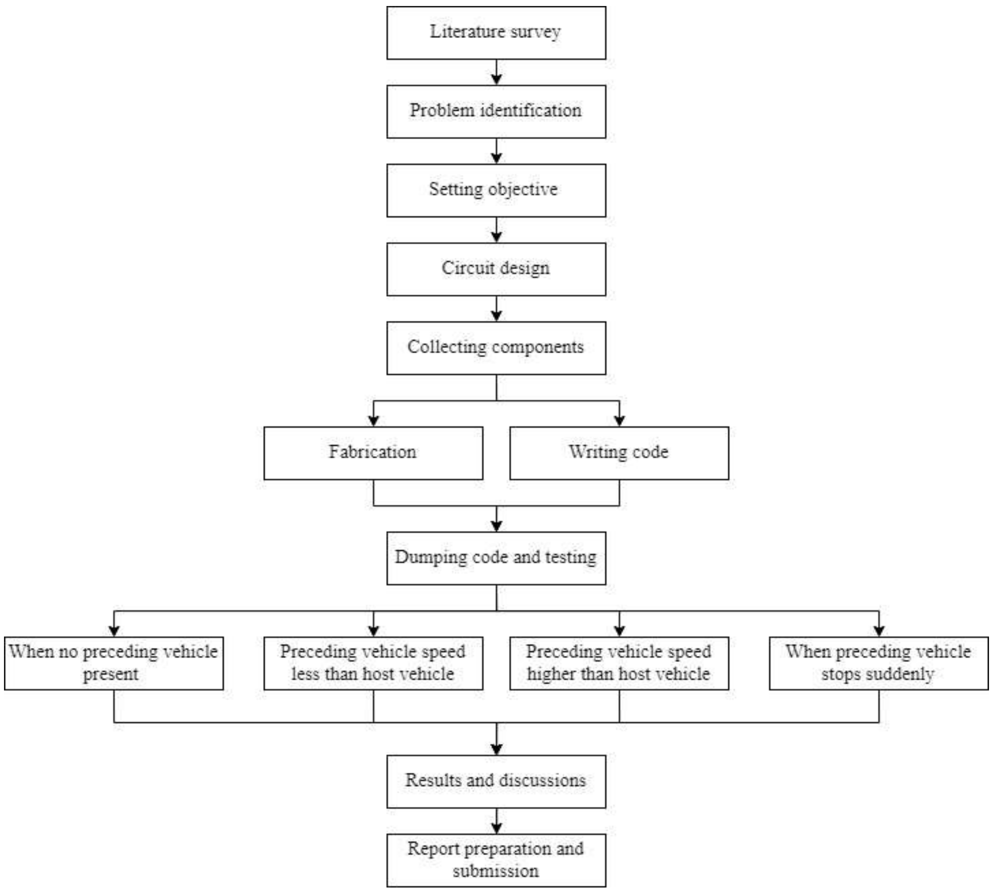



Figure 3.1: Flow of work

## 4. The list of components utilized for developing the system and it's description is as follows:

### *4.1 Arduino Uno R3:*

Arduino Uno R3 is one kind of ATmega328P based microcontroller board. It includes the whole thing required to hold up the microcontroller; just attach it to a PC with the help of a USB cable, and give the supply using AC-DC adapter or a battery to get started. The term Uno means "one" in the language of "Italian" and was selected for marking the release of Arduino's IDE 1.0 software. The R3 Arduino Uno is the 3rd as well as most recent modification of the Arduino Uno. Arduino board and IDE software are the reference versions of Arduino and currently progressed to new releases. The Uno-board is the primary in a sequence of USB-Arduino boards, & the reference model designed for the Arduino platform.

Table 4.1: Specifications of Arduino

| Parameter | specification |
|---|---|
| Operating voltage | 5V |
| i/p voltage | 6V to 20V |
| Direct current | 20 mA |

The above table 4.1 shows the specifications of Arduino. It comprises 14-digit I/O pins. From these pins, 6-pins can be utilized like PWM outputs. This board includes 14 digital input/output pins, Analog inputs-6, a USB connection, quartz crystal-16 MHz, a power jack, a USB connection, resonator-16Mhz, a power jack, an ICSP header an RST button.

Figure 4.1: Arduino UNO R3

The above figure 4.1 shows the Arduino functions. The programming of an Arduino Uno R3 can be done using IDE software. The microcontroller on the board will come with pre-burned by a boot loader that permits to upload fresh code without using an exterior hardware programmer. The communication of this can be done using a protocol like STK500. We can also upload the program in the microcontroller by avoiding the boot loader using the header like the In-Circuit Serial Programming.

The power supply of the Arduino can be done with the help of an exterior power supply otherwise USB connection. The exterior power supply (6 to 20 volts) includes a battery or an AC to DC adapter. The connection of an adapter can be done by plugging a center-positive plug (2.1mm) into the power jack on the board. The battery terminals can be placed in the pins of $V_{in}$ as well as GND. The power pins of an Arduino board include the following.

- $V_{in}$: The input voltage or $V_{in}$ to the Arduino while it is using an exterior power supply opposite to volts from the connection of USB or else RPS (regulated power supply). By using this pin, one can supply the voltage.
- 5Volts: The RPS can be used to give the power supply to the microcontroller as well as components which are used on the Arduino board. This can approach from the input voltage through a regulator.
- 3V3: A 3.3 supply voltage can be generated with the on-board regulator, and the highest draw current will be 50 mA.
- GND: GND (ground) pins.

The communication protocols of an Arduino Uno include SPI, I2C, and UART serial communication and We know that an arguing Uno R3 includes 14-digital pins which can be used as an input otherwise output by using the functions like pin Mode (), digital Read(), and digital Write(). These pins can operate with 5V, and every digital pin can give or receive 20mA, & includes a 20k to 50k ohm pull up resistor. The maximum current on any pin is 40mA which cannot surpass for avoiding the microcontroller from the damage. Additionally, some of the pins of an Arduino include specific functions.

### *4.2 HC-SR04 Ultrasonic Sensor:*

The HC-SR04 is a type of ultrasonic sensor, which uses sonar to find out the distance of the object from the sensor. It provides an outstanding range of non-contact detection with high accuracy & stable readings. It includes two modules like ultrasonic transmitter & receiver. This sensor is used in a variety of applications like measurement of direction and speed, burglar alarms, medical, sonar, humidifiers, wireless charging, non-destructive testing, and ultrasonography. The below table 4.2 shows the specifications of ultrasonic sensor.

Table 4.2: Specifications of ultrasonic sensor

| Parameter | specification |
|---|---|
| voltage | 5V |
| frequency range | 40Hz |
| Effectual Angle | <15° |
| distance range | 2cm to 800 cm |
| Direct current | 15mA |

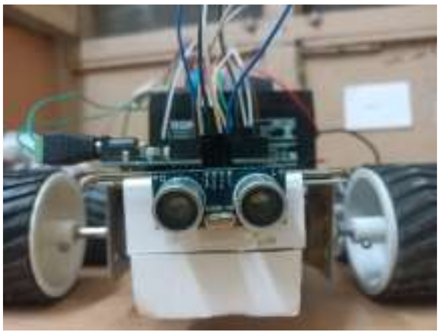


Figure 4.2: HC-SR04 ultrasonic sensor

The above figure 4.2 shows the ultrasonic sensor functions. The HC-SR04 Ultrasonic sensor comes with four pins namely Vcc pin, Trigger pin, Echo pin, & Ground pin. This sensor is used to measure the accurate distance between the target and the sensor. This sensor mostly works on the sound waves. When the power supply is given to this module, it generates the sound waves to travel throughout the air to hit the necessary object. These waves strike and come back from the object, then collects by the receiver module. Here both the distance as well as time has taken is directly proportional because the time taken for more distance is high. If the trigger pin is kept high for 10 µs, then the ultrasonic waves will be generated which will travel at the sound speed. So it creates eight cycles of sonic burst that will be gathered within the Echo pin. This ultrasonic sensor is interfaced with Arduino to gauge the necessary distance between sensor & object. The distance can be calculated using the following formula.

$S = \frac{VT}{2}$

Where the ‘S’ is the required distance

‘V’ is the sound’s speed

‘t’ is the time taken for sound waves to return after striking the object.

The actual distance can be calculated by dividing its value with 2 as the time will be twice once the waves travel and get back from the sensor. The measurement of accurate distance can be achieved by interfacing the HC-SR04 sensor with diverse types of Arduino boards. At first, give the power supply to the sensor to turn on and connect the GND pin of this sensor to the GND pin of the Arduino board. And the sensor module can be powered up with the voltage supply of the Arduino board when the current which is drawn through the sensor is below 15mA. So the Arduino current ratings will not affect the sensor. Once the primary arrangement is set up then connect both the pins of sensors like Trig & Echo to the Arduino board’s input/output pins. As we discussed earlier, the Trig pin in the sensor must be kept 10us in the beginning to start the measurement method. So, this sensor module will generate sound waves by the 40,000 Hz frequency around for every second from the source. When the sound waves return, the Echo pin will activate until these waves are obtained by the receiver. The time will be measured with the help of an Arduino board.

### *4.3 Hc-05 Bluetooth Module:*

The HC-05 is a popular Bluetooth module which can add two-way (full-duplex) wireless functionality to your projects. The below table 4.3 shows the specifications of Bluetooth module.

Table 4.3: Specification of Bluetooth module

| Parameter | specification |
|---|---|
| voltage | 4V to 6V |
| <100m | <100m |
| Direct current | 30mA |

The below figure 4.3 shows the Bluetooth working. The HC-05 is a popular module which can add two-way (full-duplex) wireless functionality to your projects. You can use this module to communicate between two microcontrollers like Arduino or communicate with any device with Bluetooth functionality like a Phone or Laptop. There are many android applications that are already available which makes this process a lot easier. The module communicates with the help of USART at 9600 baud rate hence it is easy to interface with any microcontroller that supports USART. We can also configure the default values of the module by using the command mode. So if you looking for a Wireless module that could transfer data from your computer or mobile phone to microcontroller or vice versa then this module might be the right choice for you. However, do not expect this module to transfer multimedia like photos or songs; you might have to investigate the CSR8645 module for that.

### *4.4 L298N Motor Driver Module:*

It is a dual full-bridge driver IC with high current & voltage, designed to allow typical TTL logic levels to control different inductive loads like DC motors, solenoids, relays, stepper motors, etc. The motor driver is a small current amplifier that uses a low current signal to provide a high current signal for driving an electric motor. L298 IC includes four separate power amplifiers where two amplifiers can form H-bridge A and the other two types of amplifiers can form H-bridge B. Here, One H Bridge is used to switch the polarity to control the motor direction whereas pair of H bridges are used for controlling a bipolar stepper motor. Each bridge in this IC includes two current sense pins like CSA & CSB and enable pins like ENA & ENB. Here, current sense pins are connected to the ground terminal but we can also include a low-value resistor where its voltage reading is relative to the current. Similarly, enable pins can also be used to make all the outputs active simultaneously. All the enable and input pins in this IC work with 5V TTL logic to make the connection simple with several types of microcontrollers. The operating voltage supply is up to 46 V.

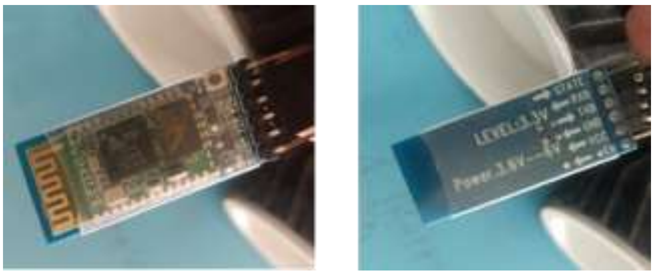


Figure 4.3: hc-05 Bluetooth module

Table 4.4: Specifications of motor driver

| Parameter | Specification |
|---|---|
| Voltage | +5V to +46V |
| Power dissipation | 25W |
| Storage temperature | -40°C – 150°C |
| Operating temperature | -23°C – 130°C |
| Direct current | 4A |

The above table 4.4 shows the specifications of motor driver. To understand the working of the L298 motor driver IC, consider the following simple circuit configuration. In this circuit, one of the H Bridge of l298 IC is used. This circuit can be designed with two push buttons which are denoted Q1 & Q2. These pushbuttons are used to control the inputs of bridge-A. Here, these logic inputs are provided through a microprocessor or a microcontroller within application circuits. The four diodes used in the circuit are flyback diodes that protect the IC from voltage spikes. Here, the function of bridge-A depends on the enable pin. Once the enable pin is pulled high using a resistor then Bridge-A will enable and start working. Similarly, when it is pulled to GND, it will be disabled so stops working.

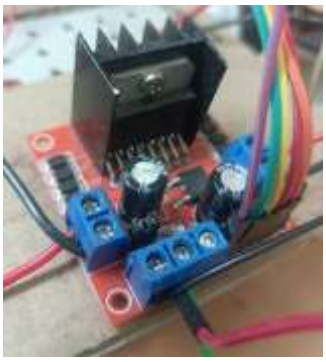


Figure 4.4: L298N motor driver

The above figure 4.4 shows the motor driver functions once the circuit is connected, we need to push the two buttons like Q1 & Q2 for changing the current flow among two output pins like OUT1 & OUT2. The logic control follows as

- When Push buttons Q1 is high and Q2 is low then it is forward current.
- When Push buttons Q1 is low and Q2 is high then it is reverse current
- When both the pushbuttons are equal like Q1=Q2 then it is a quick motor stop

If the Q1 push button is pressed then the current starts flowing from Output to Output2 the motor's rotating direction will be clockwise. Similarly, If the Q2 push button is pressed, the current starts flowing from Output2 to Output1 then the motor direction will be in the anti-clockwise direction. If both pushbuttons like Q1 & Q2 are simultaneously pressed or released then the motor will be stopped immediately. So, in this way, the motor speed can be controlled through an L298 motor driver IC.

### *4.5 200 Rpm Dc Motor:*

The DC Series Motor functions similarly to other motors by converting electrical energy into mechanical energy through the principle of electromagnetism. It operates when a magnetic field is formed and a current-carrying conductor interacts with an exterior magnetic field, resulting in a rotating motion. The motor is divided into two types based on construction: self-excited and separately excited, with self-excited motors further categorized into DC series and DC shunt motors. Once started, the series motor gradually reaches its maximum speed and torque. Components of the motor include the rotor (armature), commutator, stator, axle, field windings, and brushes. The stator is the fixed component consisting of two or more electromagnet pole parts, while the rotor comprises the armature and windings connected to the commutator. The armature is powered by connecting the power source to its windings through brushes attached to the commutator. The rotor includes a central axle for rotation, and the field winding must be able to hold a high current to produce the desired torque. Solid gauge wire or copper bars can be used to fabricate the winding for efficient heat dissipation due to the large current flow.

Figure4.5: 200 RPM DC MOTOR

The speed control of DC motors can be attained by using the two following methods

- Flux control Method
- Armature-resistance Control Method.

The above figure 4.5 shows the motor working. The armature-resistance control method is the most commonly used method for controlling the speed of the DC series motor. This is because it allows for changing the flux generated by the motor. The difference in flux can be achieved through three methods, namely, field diverters, armature diverters, and tapped field control. In the armature resistance control method, a variable resistance is connected in series with the supply, which reduces the voltage across the armature and slows down the speed. By adjusting the value of the variable resistance, any speed below the regular speed can be achieved. This method is widely used to regulate the speed of the DC series motor.

In general, for this motor, there are 3-characteristic curves are considered significant like Torque Vs. armature current, Speed Vs. armature current, & Speed Vs. torque. These three characteristics are determined by using the following two relations.

$T_a \propto \phi . I_a$

$N \propto \frac{E}{\emptyset}$

The above two equations can be calculated at the equations of emf as well as torque. For this motor, the back emf's magnitude can be given with the similar DC generator e.m.f equation like $E_b = \frac{P\phi\, NZ}{60A}$. For a mechanism, A, P, and Z are stable, thus, $N \propto \frac{E}{\emptyset}$.

The DC series motor torque equation is:

Torque= Flux* Armature current

$T = I_f * I_a$

Here $I_f = I_a$, then the equation will become

$T = I^2$

'$E_b$' is the motor's induced EMF called Back EMF

'A' is the no. of parallel lanes throughout the armature among the reverse polarity brushes

'P' is the no. of poles

'N' is the speed

'Z' is the whole number of conductors within the armature

'ϕ' is a helpful flux for each pole

'L' Active length of the conductor

'I' Current passing through the conductor

The DC series motor torque (T) can be proportional to the $I_a$^2 (square of the armature current). In load test on DC series motor, the motor should be activated on load condition because if the motor can be activated on no load, then it will achieve an extremely high speed.

## 5. Circuit design and coding:

The circuit is designed using Proteus 8 software. The Proteus Design Suite is a proprietary software tool suite used primarily for electronic design automation. The software is used by electronic design engineers and technicians to create schematics and electronic prints for manufacturing printed circuit boards. First, all required electronic components libraries must be downloaded and kept in program files and then dragon and drop each component on to schematic work space in Proteus software and connections are done. The below figure 5.1 shows the circuit design of the project.

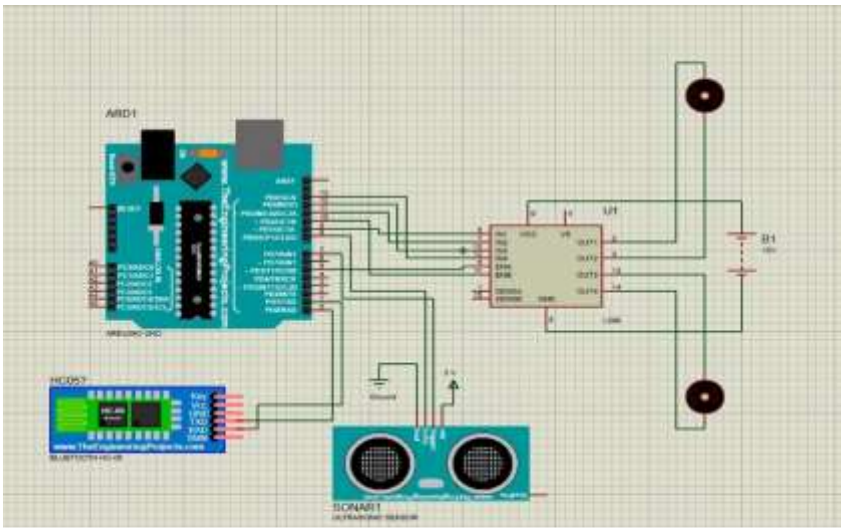



Figure 5.1: Circuit design

### *5.1 Circuit Connections:*

**Ultrasonic sensor to Arduino UNO connections:**

Trigger, Echo pins of ultrasonic sensor are connected to digital pins of Arduino UNO, Vcc pin is connected to 5V power output pin and GND pin is connected to ground pin of Arduino UNO as shown in circuit design.

**Bluetooth module to Arduino UNO connections:**

Bluetooth module has six pins but only four pins used for connection purpose. RXD and TXD pins of Bluetooth module are connected to TXD and RXD pins of Arduino UNO respectively and Vcc pin in connected 5V power output pin and GND pin is connected to ground pin of Arduino UNO as shown in circuit design.

**Motor driver to Arduino UNO connections:**

Four input pins (IN1, IN2, IN3, IN4) of motor driver is connected to any four digital pins of Arduino UNO and two enable pins are connected to two PWM pins of Arduino UNO. And common ground is connected between motor driver and Arduino board as shown in circuit design.

**DC motors to motor driver connections:**

Motor driver has four output ports the two terminals of two DC motors are connected to these four output ports. Red, black terminals of first DC motor is connected to out1 and out2 port respectively and red, black terminals of second DC motor is connected to out3 and out4 port respectively as shown in circuit design.

**Battery to motor driver connections:**

12V battery is connected to motor driver power supply ports (5V, GND, 12V). Red terminal of battery was connected to 12V port and black terminal is connected to GND port of motor driver as shown in circuit design.

**Arduino IDE software and code:**

Arduino IDE software is used for writing the code and then it is dumped into Arduino. The Arduino IDE is an open-source software, which is used to write and upload code to the Arduino boards. The IDE application is suitable for different operating systems such as Windows, Mac OS X, and Linux. It supports the programming languages C and C++. Here, IDE stands for Integrated Development Environment. The program or code written in the Arduino IDE is often called as sketching. We need to connect the Genuino and Arduino board with the IDE to upload the sketch written in the Arduino IDE software. The sketch is saved with the extension '.ino.' The Upload button compiles and runs our code written on the screen. It further uploads the code to the connected board. Before uploading the sketch, we need to make sure that the correct board and ports are selected. If any errors in the code it will display the error message and action should be taken according to it. In this project, connection between Bluetooth module and Arduino should be removed before uploading the code as Arduino may think that it is receiving data from Bluetooth module and error is displayed while uploading the code. The code of this project was shown in the figure 5.2. This is an Arduino code for a car that uses an ultrasonic sensor to measure distance and a motor driver to control the speed of the car. The car is designed to maintain a safe distance from the car in front of it and slow down gradually as it approaches the preceding vehicle. Here's what each part of the code does:

- The first section of the code defines the pins that will be used for the ultrasonic sensor and the L298N motor driver. It also sets an initial maximum speed for the motors and defines the distance at which the car should start slowing down and the safe distance to maintain from the preceding vehicle.
- In the **setup()** function, the code initializes the pins for the ultrasonic sensor and motor driver, sets the motor speed to zero, and initializes the Bluetooth module for communication.
- In the **loop()** function, the code reads the distance from the ultrasonic sensor and checks for incoming data from the Bluetooth module. If there is incoming data, it sets the maximum speed of the car accordingly. Then, it calls the **speedOfCar()** function to calculate the appropriate speed for the motors and set the motor speed based on the calculated speed.
- The **speedOfCar()** function takes the maximum speed, distance and duration as inputs and calculates the appropriate speed for the motors based on the distance from the preceding vehicle. If the distance is less than or equal to the slowing distance, the car slows down gradually as it approaches the preceding vehicle. If the distance is less than or equal to the safe distance, the car stops completely. If there is no preceding vehicle, the car maintains the predefined maximum speed. Finally, the function sets the motor speed based on the calculated speed.

sketch_mar09a | Arduino 1.8.19

File Edit Sketch Tools Help

sketch_mar09a

```
// Define the pins for the ultrasonic sensor
const int trigPin = 2;
const int echoPin = 3;

// Define the pins for the L298N motor driver
const int enA = 9;
const int in1 = 5;
const int in2 = 6;
const int enB = 10;
const int in3 = 11;
const int in4 = 12;

// Define the initial maximum speed of the motors (between 0 and 255)
int constant = 0;

// Define the distance at which the car should start slowing down
const int slowingDistance = 20; // in centimeters

// Define the safe distance to maintain from the preceding vehicle
const int safeDistance = 10; // in centimeters

void setup() {
  // Initialize the ultrasonic sensor pins
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);

  // Initialize the L298N motor driver pins
  pinMode(enA, OUTPUT);
```

sketch_mar09a | Arduino 1.8.19

File Edit Sketch Tools Help

sketch_mar09a

```
  // Initialize the L298N motor driver pins
  pinMode(enA, OUTPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(enB, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);

  // Set the motor speed to zero
  analogWrite(enA, 0);
  analogWrite(enB, 0);

  // Initialize the Bluetooth module
  Serial.begin(9600);
}

void loop() {
  // Read the distance from the ultrasonic sensor
  long duration, distance;
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = (duration / 2) / 29.1;
```

sketch_mar09a | Arduino 1.8.19

File Edit Sketch Tools Help

sketch_mar09a

```
  // Check if there is any data available from the Bluetooth module
  if (Serial.available() > 0&&Serial.parseInt()>0) {
    // Read the incoming data and set the maximum speed accordingly
    constant = Serial.parseInt();
  }
speedOfCar(constant,duration, distance);
}

void speedOfCar(int maxSpeed,int duration,int distance){
  // Determine the speed to set the motors to based on the distance from the preceding vehicle
  int speedA, speedB;
  if (distance <= slowingDistance && distance > safeDistance) {
    // Slow down gradually as the car approaches the preceding vehicle
    speedA = map(distance, slowingDistance, safeDistance, maxSpeed, 0);
    speedB = map(distance, slowingDistance, safeDistance, maxSpeed, 0);
  } else if (distance <= safeDistance) {
    speedA = 0;
    speedB = 0;
  } else {
    // Maintain the predefined speed if there is no preceding vehicle
    speedA = maxSpeed;
    speedB = maxSpeed;
  }

  // Set the motor speed based on the calculated speed
  analogWrite(enA, speedA);
  digitalWrite(in1, HIGH);
```

sketch_mar09a | Arduino 1.8.19

File Edit Sketch Tools Help

sketch_mar09a

```
    constant = Serial.parseInt();
  }
speedOfCar(constant,duration, distance);
}

void speedOfCar(int maxSpeed,int duration,int distance){
  // Determine the speed to set the motors to based on the distance from the preceding vehicle
  int speedA, speedB;
  if (distance <= slowingDistance && distance > safeDistance) {
    // Slow down gradually as the car approaches the preceding vehicle
    speedA = map(distance, slowingDistance, safeDistance, maxSpeed, 0);
    speedB = map(distance, slowingDistance, safeDistance, maxSpeed, 0);
  } else if (distance <= safeDistance) {
    speedA = 0;
    speedB = 0;
  } else {
    // Maintain the predefined speed if there is no preceding vehicle
    speedA = maxSpeed;
    speedB = maxSpeed;
  }

  // Set the motor speed based on the calculated speed
  analogWrite(enA, speedA);
  digitalWrite(in1, HIGH);
  digitalWrite(in2, LOW);
  analogWrite(enB, speedB);
  digitalWrite(in3, HIGH);
  digitalWrite(in4, LOW);}
```

Figure 5.2: Code

## 6. Experimental procedure:

### *6.1 Experimental setup:*

First two DC motors are connected to chasses alternatively as shown in figure 6.1, ultrasonic sensor is attached on the front end of the vehicle as shown in figure 6.2, Arduino and motor driver is attached on the top of the chasses as shown in figure 6.3, 12V battery is placed between Arduino and motor driver and Bluetooth module is attached on the left-hand side of the Arduino when it is seen from rear end. Now connection is made between Arduino and remaining components by following circuit design.

Figure 6.1: Chassis with two DC motors

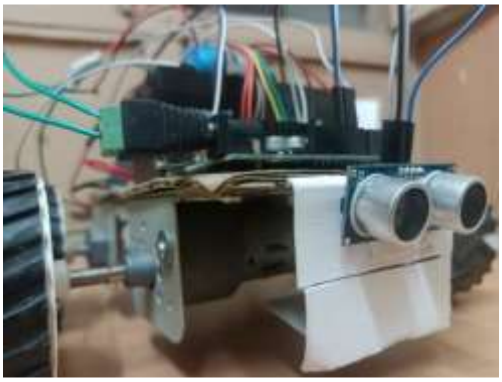


Figure 6.2: Ultrasonic sensor attached at front end of vehicle

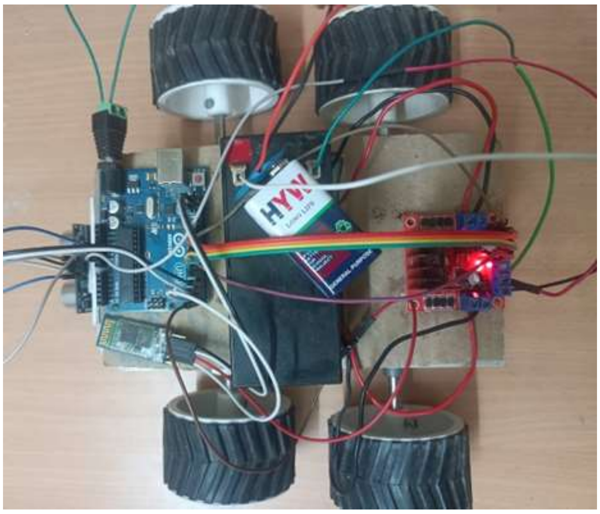


Figure 6.3: Top view of whole assembly

**Ultrasonic sensor to Arduino connections:** 5V power input pin (VCC) and ground (GND) pin of ultrasonic sensor is connected to 5V output pin and ground pin of Arduino board and Echo, Trig pins are connected to second and third digital pins of Arduino.

**Bluetooth module to Arduino connections:** The VCC and GND pins of Bluetooth module are connected to 5V power output and GND pins of Arduino and RXD and TXD pins are connected to TXD and RXD pins of Arduino respectively (these pins are removed at the time of dumping the code).

**Motor driver to Arduino connections:** The ENA and ENB enable pins are connected to 9 and 10 PWM pins of Arduino and four input pins (IN1, IN2, IN3, IN4) of motor driver are connected to 5th, 6th, 11th, 12th pins of Arduino, respectively.

**Motor driver to DC motors connections:** The output 1, output2 ports of motor driver are connected to red and black terminal of one DC motor respectively and output 3, out output 4 ports of motor driver are connected to red and black terminal of second DC motor, respectively.

**Power supply connections:** 12V power supply is given to motor driver from battery and 5V power is given to Arduino using 9V HW battery

***6.2 Testing procedure:***

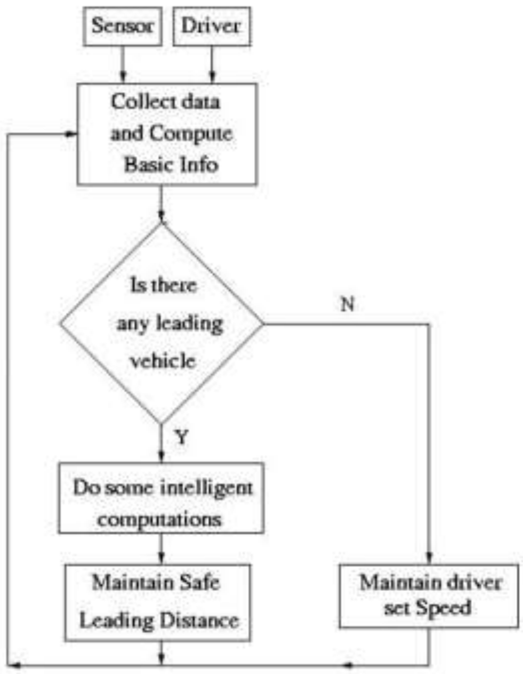



Figure 6.4: Testing procedure

After assembling all components on the chassis Bluetooth module is used to start the vehicle with predefined speed and this speed can be changed anytime according to drivers comfort. The vehicle will movie according to set speed until new speed value is read from Bluetooth module. Now a new vehicle is introduced in front of the host vehicle and it is operated manually, according to this vehicle host vehicle will adjust its speed above figure 6.4 shows the testing procedure of the project.

**Setting predefined speed:**

First Bluetooth module is used to set predefined speed of vehicle. Bluetooth module is connected to Arduino board and the module is connected wirelessly to smart phone or laptop. And by using serial Bluetooth terminal app input is given to Arduino then vehicle starts running according to predefined speed.

**Distance measurement:**

Secondly ultrasonic sensor is used to calculate the distance between host vehicle and preceding vehicle at regular intervals of time. First, the sound travels away from the sensor, and then it bounces off a surface and returns. To measure the distance the sound has travelled we use the formula:

Distance $=\frac{time \times speed\ of\ sound}{2}$

The “2” is in the formula because the sound must travel back and forth. And this calculated distance sent to Arduino board through jumper wires.

**Constant speed and safe distance w.r.t preceding vehicle principle:**

After calculating distance using ultrasonic sensor. According to this distance the speed of motors is adjusted, means it tries to maintain predefined distance between host and preceding vehicle which automatically turns to maintain constant speed with respective to preceding vehicle. Arduino UNO and L298N motor driver is used to do all these functions.

Arduino board is operated by dumping code into it which is done by using Arduino IDE software. And according to this code we can achieve our required operations.

Motor has two enable pins and four input pins. Enable pins are connected to PWM pins and other four input pins are connected to any digital pins of Arduino board. Here we have used enable pins to control speed of two dc motors.

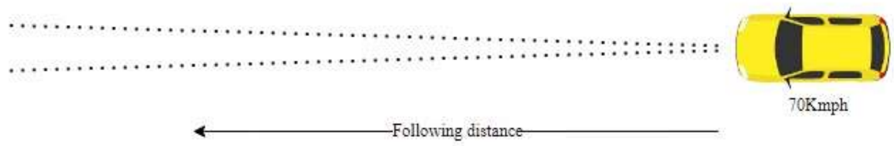
By this approach both safe distance and constant speed according to preceding vehicle is achieved.

***6.3 Testing conditions:***

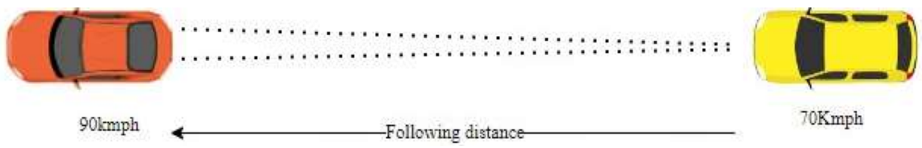
Four different conditions has been tested on the prototype by varying two parameter between host and preceding vehicle and results are taken based on distance parameter in all four different cases.

**Four different testing conditions are:**

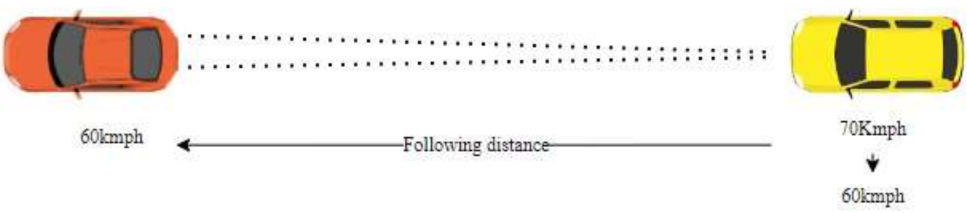
1. When no preceding vehicle present.



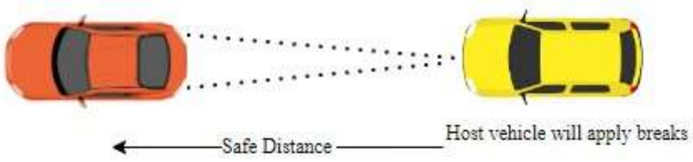

2. When preceding vehicle present less than or equal to following distance and if host vehicle speed is less than preceding vehicle.




3. When preceding vehicle present less than or equal to following distance and if host vehicle speed is greater than preceding vehicle.




4. When preceding vehicle present lest than safe distance.



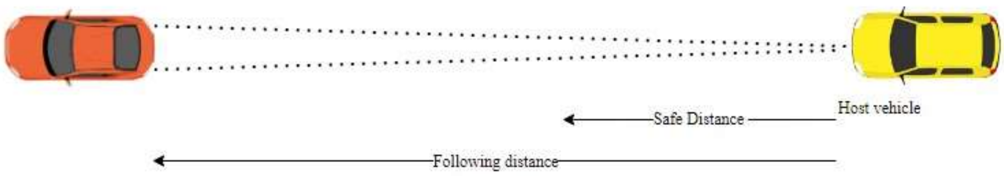

These conditions were created by taking another preceding vehicle and operating it manually to obtain required conditions.

**Two parameters are:**

1. Safe distance.
2. Following distance.

**Safe distance** means a minimum distance is set by user according to his outside conditions and his comfort, where the vehicle will apply breaks when any object is detected in front of the host vehicle less than this distance as shown in figure.

**Following distance** means a distance is set by user where the vehicle will start slowing down when any vehicle is detected in this range and started to main constant speed according to preceding vehicle speed as shown in figure.




These two parameters were varied in these four different conditions and results are taken experimentally. And checked the response of vehicle at different safe distance and following distances.

## 7. Results and discussion:

This study is concentrated on stop and go cases where adaptive cruise control system gets turned off. In this study distance is calculated using ultrasonic sensor. By using this device we can calculate distance from 2cm to 400cm and sensing angle is 15 degrees each on both sides from centre of line and overall angular range is 30 degrees. It cannot measure distance which are out of distance range or sensing range, these are some major drawbacks when we consider it for real applications. But it can give accurate results when we consider for shorter distances. For this we have prepared a prototype and tested in four different cases as shown in Table 7.1 to demonstrate the adaptive cruise control system in stop and go and normal scenarios.

Table7.1: Different cases of adaptive cruise control system.

| S.NO. | DIFFERENT CASES | ACTION |
|---|---|---|
| 1. | When no vehicle present | Host Vehicle maintains predefined speed |
| 2. | When preceding vehicle present less than 20 cm away (if host vehicle speed is less than preceding vehicle speed) | Host Vehicle maintains predefined speed |
| 3. | When preceding vehicle present less than 20 cm away (if host vehicle speed is greater than preceding vehicle speed) | Host Vehicle maintains preceding vehicle speed |
| 4. | When preceding vehicle present less than 10 cm away | Host vehicle will stop |
| 5. | When preceding vehicle starts moving after rest position | Host vehicle will continue with predefined speed set by driver |

We have created our prototype in such a way that ACC feature will not be turned off after applications of breaks also, this is the area where our project concentrated on.

First we have tested our prototype when there are no preceding vehicles present. In this case, host vehicle will maintain predefined speed which is set by us using Bluetooth module (we can set lower speed values also). Input predefined speed is taken from serial monitor which is connected to prototype using Bluetooth module and predefined speed can be changed any time according to drivers comfort.

Secondly, when there are preceding vehicles present 20cm away from host vehicle this is following distance and it can be changed according to traffic conditions. In this case host vehicle will maintain its predefined vehicle speed if its speed is less than preceding vehicle speed. When there are preceding vehicles present 20cm away from host vehicle and if its speed is greater than preceding vehicle speed then the host vehicle will adjust its speed according to preceding vehicle speed and vehicle will stop when the distance between vehicle is less than 10 cm this safe distance and it can be adjusted according to braking conditions. And the again the host vehicle will continue with the set speed when there is no vehicle present or present 20cm away from it. These all cases are tested using our prototype as shown in Table 7.2.

Table7.2: Comparison between experimental and theoretical results.

| S. No. | Preceding vehicle distance | Host vehicle Action taken(experimental results) | Host vehicle action needed (theoretically results) |
|---|---|---|---|
| 1. | >20cm | Vehicle continue with set speed | Vehicle continue with set speed |
| 2. | =20cm | Vehicle slow downs | Vehicle slow downs |
| 3. | <20cm | Vehicle adjust its speed to preceding vehicle speed | Vehicle adjust its speed to preceding vehicle speed |
| 4. | <10cm | Vehicle apply breaks | Vehicle apply breaks |

Here 20cm is following distance and 10cm is safe distance and these can be changed according to drivers comfort and outside conditions. The present prototype is tested at different safe and following distance experimentally.

(a) (b) (c)

Figure 7.1: Accuracy testing of ultrasonic sensor

These results shows that ultrasonic sensor sensing ability is getting less as we increasing sensing distance as shown in figure 7.1. In figure 7.1 (a) the following distance is 20cm here the car will follow accurately. When the following distance is 100cm it can be observed in figure 7.1 (b), (c) that car is crossing the line and it is not following the exact following distance. So, in real life applications some advance sensors are used like radar, laser and camera sensors are used for more accurate results.

In this study prototype is also checked experimentally to find accuracy of sensing component (ultrasonic sensor) when any vehicle is appeared suddenly at 90 degrees with respective to sound waves emitted by ultrasonic sensor and if it is less than safe distance then breaks are applied and host vehicle will stop as shown in figure 7.2.

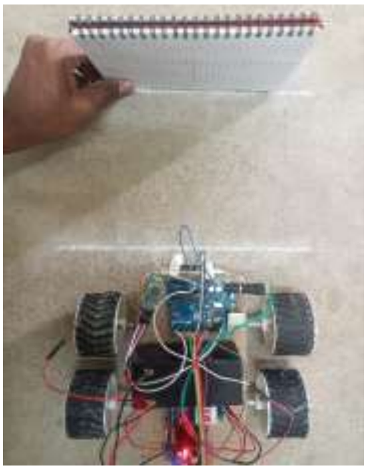


Figure 7.2 Obstacle appeared at 90 degrees

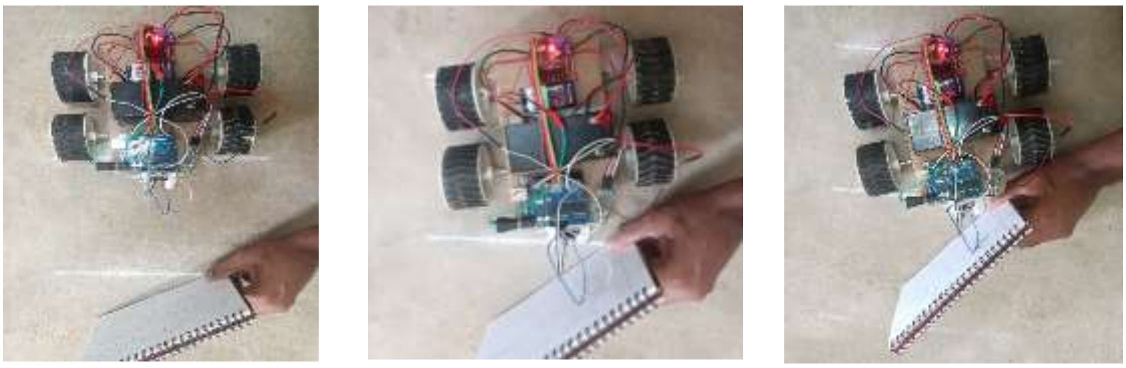


Figure 7.3: Obstacle appeared other than 90 degrees

If vehicle is appeared at other than 90 degrees then breaks are not applied, host vehicle will continue to move as shown in figures 7.3.

Above figures shows that as we increase the angle the accuracy of distance detection become less.

And this is major drawback of ultrasonic sensor as it cannot detect distance of preceding object if the angle between incident sound waves and preceding object surface is other than 90 degrees with respective to emitting sound waves.

## Conclusion:

In this project, we presented a prototype adaptive cruise control system that uses an Arduino UNO microcontroller and an ultrasonic sensor to detect obstacles and adjust the speed of a vehicle accordingly. The main focus was on stop-and-go traffic scenarios where the system can maintain a safe following distance even when the vehicle in front comes to a sudden stop. The results demonstrate that the prototype is effective at detecting obstacles and adjusting the speed of the vehicle accordingly. It can be found that the system was able to maintain a safe following distance in a variety of driving conditions, including heavy traffic and sudden stops. Overall, this work represents a crucial step forward in the development of adaptive cruise control systems that can handle a wide range of driving scenarios. While there is still work to be done to refine the system and optimize its performance, these results provide solid foundation for future research in this area.

**Abbreviations:**

| | |
|---|---|
| ACC | -Adaptive Cruise Control |
| IDE | - Integrated Development Environment |
| USB | - Universal Serial Bus |
| PWM | - Pulse Width Modulation |
| LED | - Light Emitting Diode |
| LIDAR | -Light Detection and Ranging |
| CA | -Collision Avoidance |
| ADAS | - Advanced Driver Assistance System |
| GPS | - Global Positioning System |
| TOF | -Time of Flight |
| IC | - Integrated Circuit |
| MCU | -Microcontroller Unit |
| DC | - Direct Current |
| GND | -Ground |
| RPS | -Regulated Power Supply |
| TXD | -Transmitting Data |
| RXD | - Receiving Data |
| ENA and ENB | - Enable Pins |

**References:**


1. Lang chen, Jianbo zhang, Yao wang, "Wireless car control system based on ARDUINO UNO R3." IEEE volume 2(2018):1783-1787.
2. "Kalman filter-based tracking of moving objects using linear ultrasonic sensor array for road vehicles." *Mechanical Systems and Signal Processing* 98 (2018): 173-189.
3. Yi, Kyongsu, et al. "Design of a full-range ACC with collision avoidacne/mitigation braking." *IFAC Proceedings Volumes* 40.10 (2007): 17-134.
4. Varotto, Silvia F., et al. "Do adaptive cruise control and lane keeping systems make the longitudinal vehicle control safer? Insights into speeding and time gaps shorter than one second from a naturalistic driving study with SAE Level 2 automation." *Transportation research part C: emerging technologies* 141 (2022): 103756.
5. Schrödel, Frank, Patrick Herrmann, and Norman Schwarz. "An improved multi-object adaptive cruise control approach." *IFAC-Papers Online* 52.8 (2019): 176-181.
6. Lin, Tsang-Wei, Sheue-Ling Hwang, and Paul A. Green. "Effects of time-gap settings of adaptive cruise control (ACC) on driving performance and subjective acceptance in a bus driving simulator." *Safety science* 47.5 (2009): 620-625.
7. R.Stiawan, A. Kusumadjati, N.S. Aminah, M. Djamal, & S.Viridi."An ultrasonic sensor system for vehicle detection application." *Journal of Physics: Conference Series*. Vol. 104. No. 1. IOP Publishing, 2019.

8. Carullo, Alessio, and Marco Parvis. "An ultrasonic sensor for distance measurement in automotive applications." *IEEE Sensors journal* 1.2 (2001): 143.

9. Subodh B. Damale, Swati J Jangade, Swati C Wanjari, Abhishekh P. Goswami, “Four Quadrant DC motor speed control using Arduino.”IJSRD volume 9(2021): 410-41.

10. Kesting, Arne, et al. "Adaptive cruise control design for active congestion avoidance." *Transportation Research Part C: Emerging Technologies* 16.6 (2008): 668-683.

11. Bifulco, Gennaro Nicola, et al. "Development and testing of a fully adaptive cruise control system." *Transportation Research Part C: Emerging Technologies* 29 (2013): 156-170.

12. Hoomod, Haider Kadhim, and Sadeem Marouf M. Al-Chalabi. "Objects detection and angles effectiveness by ultrasonic sensors HC-SR04." IJSR 6 (2017): 6.