Bluauto: Self Driving Car Based On Voice

Dr. Suvarna Nandyal [1], Trupti N Peshkar [2], Sachin [3], Veeresh [4]
Dept. Of Computer Science and Design

Poojya Doddappa Appa College of Engineering, Kalabuargi, *India*

suvarnanandyal@pdaengg.com [1],truptipeshkar0607@gmail.com [2] ,sachinkadli18@gmail.com[3],
rajapurvilas@gmail.com [4]

ABSTRACT

The Bluauto project resides within the **Internet of Things (IoT)** ecosystem, focusing on the development of an intelligent, voice-operated self-driving car. At its core, Bluauto strategically integrates several **IoT components**, including various sensors for environmental awareness, a Bluetooth communication module for wireless interaction, and a central microcontroller unit. This integration facilitates seamless **user interaction** through a dedicated Android application.

Cars have come a long way over the years, evolving from fully manual machines to smart, voice-controlled vehicles. In the beginning, driving was a hands-on task—drivers had to steer, shift gears, and manage everything manually, which took a lot of skill and effort. As time passed, features like automatic transmissions and power steering made driving much simpler and more comfortable. With the rise of digital technology, cars became smarter. Sensors, cameras, and onboard computers introduced helpful tools like cruise control, lane assist, and emergency braking. These advancements set the stage for driverless cars, which use AI and real-time data to drive themselves with little to no input from a human. Today, voice control has added a whole new level of convenience. Drivers can now give spoken commands to control navigation, all without taking their hands off the wheel. This journey from manual control to voice-activated, self-driving cars shows how far the automotive world has progressed, making driving safer, easier, and more intuitive than ever before

A **voice-controlled Arduino-based robotic car** that can navigate using spoken commands and avoid obstacles intelligently. The system integrates key components like the **Arduino Uno**, **HC-05 Bluetooth module**, **L293D motor driver**, and **HC-SR04 ultrasonic sensor** to achieve hands-free control and safety during movement. A smart car system is developed that **receives voice commands from a mobile app**, processes those commands using an Arduino, and performs appropriate motor actions such as moving forward, backward, turning left or right, or stopping. In addition, the car includes an **obstacle detection mechanism** that uses an ultrasonic sensor to ensure safe navigation.

A mobile app captures voice commands, converts them into text, and sends them via Bluetooth (HC-05) to the Arduino Uno, which interprets these commands to generate signals controlling the L293D motor driver and driving the car's DC motors. The HC-05 module receives Bluetooth signals and

communicates with the Arduino through serial communication. Commands such as “Forward” and “Backward” make both motors move accordingly, while “Left” and “Right” cause the motors to spin in opposite directions for turning, and “Stop” halts the motors. Simultaneously, the HC-SR04 ultrasonic sensor continuously measures the distance ahead, enabling the Arduino to detect obstacles and either stop the car or adjust movements to avoid collisions, thus integrating voice control with real-time obstacle avoidance for an interactive robotic vehicle.

The efficiency of four voice commands—Left, Right, Forward, and Backward—used to control an Arduino-based voice-controlled car over 100 test cases each. The "Forward" command had the highest efficiency at 92%, followed by "Left" at 90%. Both "Right" and "Backward" commands achieved 88% efficiency. These results indicate that all commands performed well, with accuracy above 85%, though occasional failures suggest room for improvement in voice recognition and hardware response. Factors such as background noise, pronunciation clarity, and communication delays may have contributed to the variations in performance.

The future scope of this Arduino-based voice-controlled car includes enhancing the voice recognition system by integrating more advanced algorithms such as machine learning to improve accuracy and reduce errors caused by background noise or unclear pronunciation. Expanding the range of voice commands could enable more complex functionalities, like autonomous navigation. Additionally, incorporating other sensors such as GPS or cameras can improve obstacle detection and allow for smarter path planning. Upgrading the communication module to faster or more reliable wireless technologies could reduce latency and increase responsiveness. Overall, these improvements would make the system more versatile, efficient, and suitable for practical applications in robotics and automation.

KEYWORDS: *Bluauto, Internet of Things (IoT), voice-operated, autonomous vehicle, Arduino Uno, Bluetooth communication, obstacle detection.*

I. INTRODUCTION

Cars have come a long way over the years, evolving from fully manual machines to smart, voice-controlled vehicles. In the beginning, driving was a hands-on task—drivers had to steer, shift gears, and manage everything manually, which took a lot of skill and effort. As time passed, features like automatic transmissions and power steering made driving much simpler and more comfortable. With the rise of digital technology, cars became smarter. Sensors, cameras, and onboard computers introduced helpful tools like cruise control, lane assist, and emergency braking. These advancements set the stage for driverless cars, which use AI and real-time data to drive themselves with little to no input from a human. Today, voice control has added a whole new level of convenience. Drivers can now give spoken commands to control navigation, all without taking their hands off the wheel. This journey from manual control to voice-activated, self-driving cars shows how far the automotive world has progressed, making driving safer, easier, and more intuitive than ever before.

The rapid advancement of the Internet of Things (IoT) has opened up new possibilities in the field of intelligent transportation systems, paving the way for smarter, safer, and more interactive vehicles. The *Bluauto* project is a product of this technological evolution, focusing on the development of a compact, voice-operated self-driving car that integrates IoT components for seamless human-machine interaction. Designed as a cost-effective and accessible solution, Bluauto combines wireless communication, embedded systems, and real-time processing to deliver autonomous vehicle functionality through a user-friendly interface. A dedicated Android application allows users to control the vehicle using simple voice commands such as "forward," "backward," "left," "right," and "stop," making the experience both intuitive and hands-free.

At the heart of the system lies the Arduino Uno, powered by the ATmega328P microcontroller, which acts as the central processing unit. It receives voice commands via a Bluetooth module and translates them into motor control signals, while simultaneously processing real-time input from obstacle detection sensors to ensure safe navigation and collision avoidance. The vehicle is powered by a rechargeable battery, which supplies energy to all onboard components, supporting sustained operation without the need for constant external power. By leveraging widely available hardware and open-source technologies, Bluauto demonstrates the transformative potential of IoT and embedded systems in shaping the future of smart mobility.

II. RELATED WORK

1. **"Intelligent Voice-Controlled Robotic Vehicle for Indoor/Outdoor Navigation" (Author: V. More et al., 2025)** This paper designs and develops an intelligent voice-controlled robotic vehicle for indoor and outdoor navigation. It uses **speech recognition, artificial intelligence, and robotic navigation**, equipped with **GPS, lidar, and cameras** for autonomous navigation and obstacle avoidance. The system was tested in various environments, demonstrating accurate response to voice commands and navigation of complex routes, with potential applications in healthcare, transportation, and search and rescue.
2. **"Voice Controlled Vehicle Using IoT" (Author: B. Ariyasinghe et al., 2024)** This study presents a novel design for a multifunctional intelligent vehicle aimed at providing innovative mobility solutions, particularly for disabled or elderly users. It integrates **head recognition, eye-tracking, Bluetooth control, and ultrasonic obstacle avoidance**, with three driving modes including nostril-based and eye-tracking control. Experimental evaluations showed 85% accuracy in nostril tracking, over 90% precision in eye direction detection, and efficient obstacle avoidance within a 1m range, providing non-invasive, cost-effective, and adaptable control.

3. **"Voice Controlled Autonomous Car using IoT" (Author: R. A. P. et al., 2024)** This conference paper introduces a real-time IoT-driven voice-controlled car. It aims for smoother **human-machine interaction and smart transport use** by integrating IoT components for real-time control based on voice commands. The output is a functional prototype demonstrating **real-time voice command processing and vehicle response**, promoting enhanced user convenience and safety.
4. **"Intelligent Cloud and IoT-Based Voice-Controlled Car" (Author: S. K. Rout et al., 2023)** The authors propose a **cloud-IoT hybrid model** for voice-controlled driving. They process data in the cloud to ensure efficient and responsive vehicle behavior, where the car responds to straightforward speech instructions like "forward," "backward," "left," "right," and "stop." The output is a **proof-of-concept** demonstrating effective voice control with cloud-based processing for enhanced responsiveness.
5. **"Development of an Android-Based, Voice-Controlled Autonomous Robotic Vehicle" (Author: A. U. et al., 2023)** This paper details the **Android and Arduino-based implementation** of an autonomous car with voice command processing and obstacle detection. It uses an Android app for **voice recognition** and sending commands via **Bluetooth** to an Arduino microcontroller, which then controls motors and utilizes ultrasonic sensors for obstacle avoidance. The output is a functional prototype capable of responding to voice commands for navigation and detecting obstacles for safe operation.
6. **"Voice-Based Smart Assistant System for Vehicles using RASA" (Author: A. Paranjape et al., 2023)** This paper details a conversational voice assistant for vehicles using **RASA and natural language processing**. It aims to enable interactive, voice-controlled vehicle operations. The output is a **voice assistant capable of understanding and responding to natural language commands**, facilitating hands-free control and improved driver-vehicle interaction.
7. **"Development of an Autonomous Floor Mopping Robot Controller Using Android Application" (Author: M. U. I. et al., 2022)** This project implements a **voice-controlled floor-cleaning robot** using Android integration for domestic automation. It utilizes an Android application to send voice commands to the robot for controlling cleaning tasks. The output is a **prototype of an autonomous floor mopping robot** that can be controlled via voice commands from an Android device, demonstrating its application in home automation.
8. **"Mechatronics Design and Kinematic Simulation of a Tripteron Cartesian-Parallel Agricultural Robot Mounted on 4-Wheeled Mobile Platform to Perform Seed Sowing Activity" (Author: J. Cornejo et al., 2022)** This paper focuses on the **mechatronics design and kinematic simulation of an agricultural robot**. While not directly voice-controlled, it uses simulation software like **MATLAB/Simulink, SolidWorks, and ANSYS** to analyze and refine the robot's design and operational efficiency for tasks like seed sowing. The output is a **simulated model** demonstrating the robot's kinematic behavior and design feasibility for agricultural applications.

9. **"Voice-Controlled Autonomous Car using IoT" (Author: S. V. B. K. Prasad et al., 2021)** This paper introduces an **IoT-based autonomous car controlled through voice commands**. The system utilizes **sensors, Arduino, and a voice interface** for navigation and obstacle avoidance. The output is a **prototype promoting hands-free mobility** and demonstrating the feasibility of voice control in autonomous vehicles.
10. **"A Novel Voice Controlled Robotic Vehicle For Smart City Applications" (Author: S. Balhara and N. Gupta, 2021)** This work integrates a **voice interface into robotic vehicles** designed for use in smart city scenarios. It focuses on **real-time navigation and command processing**, utilizing speech recognition modules. The output is a **robotic vehicle prototype** capable of responding to voice commands for navigation within smart city environments.
11. **"Voice Assisted Bots for Automobile Applications" (Author: S. K. Rudrawar et al., 2021)** This study discusses the design of **voice-assisted bots aimed at assisting drivers in semi-autonomous vehicle platforms**. It focuses on developing intelligent conversational agents that can interpret and execute driver commands. The output is a **conceptual framework and initial design** for voice-assisted bots that enhance driver assistance and interaction in vehicles.
12. **"IoT Based Voice Controlled Autonomous Robotic Vehicle Through Google Assistant" (Author: M. Gupta et al., 2021)** The vehicle in this study is operated through voice commands via **Google Assistant** and integrates with **IoT components for smart navigation**. It uses Google Assistant for voice recognition and sends commands to an IoT-enabled robotic vehicle. The output is a **robotic vehicle demonstrating successful voice control through Google Assistant** for autonomous navigation.
13. **"Voice Controlled Robotic Car Using Mobile Application" (Author: S. Chakraborty et al., 2021)** This study presents a car controlled via **mobile voice commands**, aiming for improved response through **Bluetooth-based communication**. It uses a mobile application with voice-to-text functionality to send commands to an Arduino-based car via Bluetooth. The output is a **robotic car prototype** that demonstrates responsive voice control via a mobile application.
14. **"Arduino Based Voice Controlled Delivery System (Robot)" (Author: N. O. Agwunedu et al., 2021)** This paper proposes a **voice-controlled robotic delivery system**, applying autonomous navigation in indoor environments. It uses **Arduino and IoT** for command processing and navigation. The output is a **prototype of an autonomous robotic delivery system** controlled by voice commands, suitable for indoor environments.
15. **"IoT-based Autonomous Vehicle for Smart Traffic Management" (Author: R. Kumar et al., 2020)** This paper discusses an autonomous vehicle integrated with **IoT modules for traffic monitoring and management**. It emphasizes **data communication between vehicles and infrastructure** to optimize traffic flow. The output is a **conceptual model** demonstrating how IoT can facilitate efficient traffic management through inter-vehicle and infrastructure communication.

16. **"A Survey of IoT-Based Autonomous Vehicles" (Author: S. K. Yadav et al., 2020)** This survey paper reviews **IoT applications in autonomous vehicles**. It focuses on **frameworks, sensor types, security concerns, and communication protocols**. The output is a **comprehensive overview** of the current state and challenges of IoT integration in autonomous vehicles.
17. **"A Path Planning Technique for Autonomous Mobile Robot" (Author: Z. Haruna et al., 2020)** The authors introduce an efficient **path planning algorithm** suitable for autonomous robots to operate in constrained and structured environments. It aims to find optimal collision-free paths. The output is an *improved A algorithm** demonstrating better search speed and shorter paths in simulated environments.
18. **"Voice Controlled Robot Car Using Arduino" (Author: S. Srivastava and R. Singh, 2020)** The paper details an **Arduino-based robot car** that responds to basic movement commands via voice. It uses an Arduino microcontroller to interpret voice commands for controlling the car's motion. The output is a **functional prototype of a voice-controlled robot car**, suitable for educational purposes and basic prototype testing.
19. **"Smart Autonomous Car with Obstacle Avoidance" (Author: M. R. Singh et al., 2019)** This study proposes a prototype of an **autonomous vehicle that can detect and avoid obstacles** using **ultrasonic sensors**. It highlights hardware implementation and real-time responsiveness. The output is a **working prototype** demonstrating effective obstacle avoidance capabilities.
20. **"Autonomous Car Control Using Voice Recognition and IoT" (Author: L. N. Fernandes et al., 2019)** This paper combines **voice recognition technology and IoT** to develop a smart control system for autonomous vehicles. It aims to increase driver convenience and safety by allowing voice commands for vehicle control. The output is a **smart control system prototype** demonstrating voice-activated functionalities in an autonomous car.
21. **"Voice-Controlled Autonomous Vehicle Using IoT" (Author: S. Sachdev et al., 2019)** A practical implementation of an **IoT-based voice-controlled vehicle** is presented, emphasizing **real-time control and human interaction**. It uses IoT for connectivity and voice recognition for direct command input. The output is a **functional, real-time voice-controlled vehicle prototype**.
22. **"A Dynamic Path Planning Technique for Autonomous Mobile Robot in Unknown Static Environment" (Author: Z. Haruna et al., 2019)** This paper focuses on **real-time path planning in unknown environments** for autonomous navigation. It enhances adaptability in robotic systems by utilizing heuristic techniques like Ant Colony Optimization and Particle Swarm Optimization. The output is a **robust path planning method** for mobile robots operating in dynamic, unknown static environments.
23. **"Voice-Activated Car Control System for Disabled Drivers" (Author: T. J. Lopez et al., 2018)** This work targets **voice-controlled car systems for physically disabled individuals**. It uses **speech recognition modules** to interpret commands for controlling the vehicle's motion.

The output is a **control system designed to provide enhanced mobility and accessibility** for disabled drivers.

24. **"Implementation of Autonomous Car Using Arduino and IoT" (Author: R. R. Sharma et al., 2018)** The authors describe an **Arduino-based car that uses IoT and sensors** for line following, object detection, and basic autonomous functions. It integrates ultrasonic sensors for collision avoidance. The output is a **partially autonomous car prototype** demonstrating basic autonomous features like obstacle avoidance and route tracking.
25. **"Voice Controlled Robotic Car Using Arduino for Smart Agriculture" (Author: D. Saravanan et al., 2018)** The paper explores the use of **Arduino and voice recognition for automating agricultural tasks**. It aims for cost-effective smart farming solutions by developing a voice-controlled robotic car for agricultural applications. The output is a **prototype of a voice-controlled robotic car** capable of performing specific agricultural tasks.
26. **"Voice Recognition Robot Control Using Android Device" (Author: A. Shalini et al., 2018)** A **mobile application with voice recognition functionality** is developed for robot control. It enables flexible operation through smartphones by transmitting voice commands via Bluetooth to a robot. The output is an **Android application capable of controlling a robot car through voice commands**.
27. **"Autonomous Vehicle Navigation and Control Using IoT" (Author: A. S. Iyer et al., 2017)** The paper discusses **IoT integration for autonomous vehicle navigation**. It focuses on **GPS, environmental sensing, and data processing** for autonomous decision-making. The output is a **framework illustrating the use of IoT** for comprehensive navigation and control in autonomous vehicles.
28. **"Design and Implementation of a Voice Controlled Robot with Human Interaction Ability" (Author: H. Rashid et al., 2017)** This work demonstrates the use of **speech-based interaction for mobile robots**. It focuses on natural user interfaces and safety mechanisms, where voice commands are used to control the robot's movement. The output is a **mobile robot prototype** demonstrating voice-controlled interaction and navigation.
29. **"A Survey of IoT-Based Autonomous Vehicles"** (Author: S. K. Yadav et al., 2020) This survey paper presents a comprehensive review of IoT applications in autonomous vehicles. The authors discuss various IoT sensors, technologies, and frameworks used to enhance vehicle autonomy, safety, and navigation. The survey also includes challenges and future trends in integrating IoT for smarter transportation solutions.
30. ***"Designing a Smart Autonomous Vehicle Using IoT and Embedded Systems"*** *(Author: V. Patel.,2021)*

    This paper presents the design and development of a smart autonomous vehicle using IoT and embedded systems. The authors combine various sensor technologies with embedded systems to create a highly responsive vehicle capable of real-time decision-making. The system also supports voice command-based control, enhancing the user experience.

III. PROPOSED METHODOLOGY

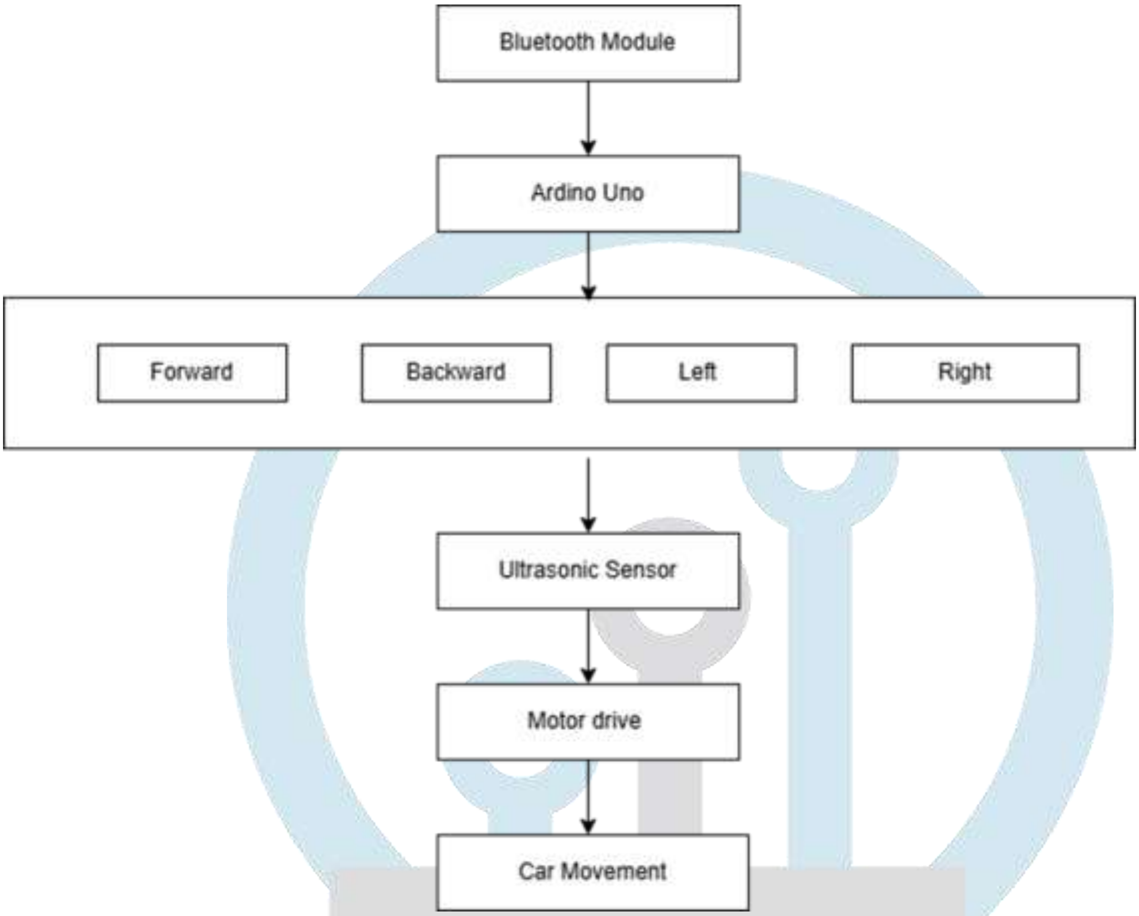

Fig 2.1 System Architecture

Bluetooth Module: The **HC-05 Bluetooth module** is a cost-effective and easy-to-use device designed for seamless wireless serial communication. It operates using the **Serial Port Protocol (SPP)** over the **2.4GHz ISM band**, and supports **Bluetooth version 2.0 + EDR (Enhanced Data Rate)**, achieving data transmission speeds of up to **3 Mbps**. At its core, the module is powered by the **CSR Bluecore 04-External** single-chip Bluetooth system, built using **CMOS technology**. It incorporates **Adaptive Frequency Hopping (AFH)** to minimize interference from other wireless devices. The HC-05 can be configured in either **Master or Slave mode**, with **Slave mode** set as the default by the manufacturer.

In voice-controlled Arduino-based systems, the HC-05 plays a critical role by **receiving voice commands**—processed and converted to text by a mobile app—and transmitting them to the Arduino via serial communication.

Pinout of the HC-05 Bluetooth Module:

1. **EN (Key)** – Pin 1: Used to enter AT command mode (optional; active high).
2. **VCC** – Pin 2: Power supply input (connect to **+5V**).
3. **GND** – Pin 3: Ground (connect to **GND** on Arduino).
4. **TXD (Transmit)** – Pin 4: Sends serial data from the HC-05 to the **Arduino RX** pin (typically **Digital Pin 0** or **2**).
5. **RXD (Receive)** – Pin 5: Receives serial data from the **Arduino TX** pin (typically **Digital Pin 1** or **3**; use a voltage divider to drop 5V to 3.3V for safety).
6. **State** – Pin 6: Indicates connection status (HIGH when connected, LOW when not connected; optional use).

Fig 2.2 HC-05 Bluetooth module

Arduino Uno: The **Arduino Uno** is a popular open-source microcontroller board based on the **ATmega328P** microcontroller. Known for its simplicity and flexibility, it is widely used in embedded systems, robotics, and DIY electronics projects. The board operates at **5V** and is powered through either a USB connection or an external power source ranging from **7–12V** via the **VIN** pin. It comes equipped with **14 digital input/output pins** (of which 6 can be used as PWM outputs), **6 analog input pins**, and essential communication interfaces such as **UART (TX/RX)**, **SPI**, and **I2C**. Digital pins **0 (RX)** and **1 (TX)** are particularly important for serial communication and are commonly connected to devices like the **HC-05 Bluetooth module**. The **analog pins A0 to A5** can read analog voltage levels and can also be configured as digital I/O when needed. Additional power pins include **5V**, **3.3V**, and **GND**, which are used to power sensors and modules. The **RESET** pin can be used to manually restart the board, and **AREF** is available for providing an external reference voltage to the analog pins.

In a **voice-controlled car system**, the Arduino Uno functions as the **central processing unit**, coordinating data received from various input sources and controlling outputs accordingly. Once powered, the Arduino initializes its peripherals, including the **Bluetooth module**, **ultrasonic sensor**, and **motor driver**. It continuously waits for commands transmitted via the Bluetooth module, which receives voice inputs from a mobile application. These commands—such as "forward", "left", or "stop"—are processed by the Arduino, which then sends appropriate signals to the motor driver to control the car's wheels. Simultaneously, the Arduino monitors data from the **ultrasonic sensor** to detect obstacles and make navigation decisions. For example, if an object is detected in front of the car, the Arduino can halt movement or issue an avoidance. Through this real-time interaction between input devices (Bluetooth, sensors) and actuators (motors), the Arduino Uno effectively enables intelligent, voice-guided movement in the robotic car.

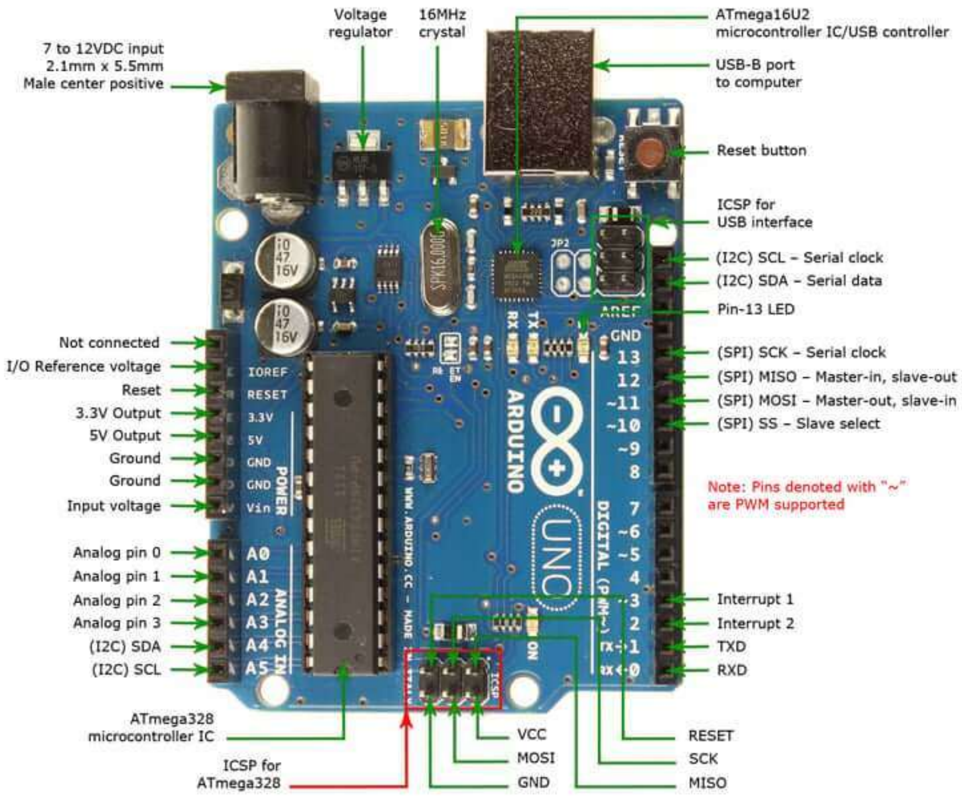

Fig 2.3 Ardino Uno

Ultrasonic Sensor: The **HC-SR04 ultrasonic sensor** is an electronic component used to determine the **distance of an object** using **SONAR (Sound Navigation and Ranging)** technology. It functions by emitting **ultrasonic waves**—sound frequencies above the human hearing range—and measuring the time it takes for the **echo** to return after bouncing off an object. Since the speed of sound is known, the sensor uses this **echo time** to calculate the distance to the object.

The sensor has **two primary components**:

- A **transmitter**, which emits ultrasonic waves using **piezoelectric crystals**, and
- A **receiver**, which detects the returning sound waves after they reflect off an object.

When the sensor sends out a pulse, it waits for the echo. The time between transmission and reception is converted into an **electrical signal**, which the **Arduino** uses to compute the object's distance with high accuracy. The HC-SR04 operates at a **frequency of 40kHz** and can measure distances from **2 cm to 400 cm**.

In a **voice-controlled car**, the ultrasonic sensor is essential for **obstacle detection**. It ensures the vehicle doesn't collide with objects by halting movement or changing direction when an obstacle is detected ahead, based on the real-time distance calculated.

Fig 2.4 Ultrasonic Sensor

Motor Drive: A motor driver is a crucial electronic component that enables a low-power microcontroller, such as an Arduino Uno, to control high-power actuators like DC motors or stepper motors. Microcontrollers typically operate at low current and voltage levels, which are insufficient to drive motors directly. A motor driver acts as an interface bridge between the control logic (Arduino) and the motor hardware, allowing precise control over motion while safely managing the required power. The L293D is one of the most popular and reliable motor driver ICs used in robotics and embedded systems. It is a quad half-H driver, meaning it can control two DC motors independently, both in terms of direction and speed. It operates with voltages up to 36V and can deliver up to 600mA per channel, making it suitable for small to medium-sized robotic applications.

Core Functionalities of the L293D Motor Driver:

Direction Control: The L293D controls motor direction by altering the polarity of voltage applied across the motor terminals. This is achieved using input pins (IN1 to IN4) that determine the rotation direction of each motor.

Speed Control: Speed is regulated by applying PWM (Pulse Width Modulation) signals to the Enable pins (EN1, EN2). Varying the PWM duty cycle adjusts the average voltage sent to the motor, allowing for smooth speed changes.

Enable/Disable Control: Each motor channel has a dedicated Enable pin that turns the motor on or off. When the enable pin is LOW, the motor is disabled regardless of input pin states.

In a **voice-controlled** vehicle, the L293D motor driver serves as the final actuator interface. Voice Input: A mobile app captures the user's voice command (e.g., "Forward", "Left"), converts it to text, and sends it to the Arduino Uno via the HC-05 Bluetooth module. Command Processing: The Arduino decodes the received command and translates it into electrical control signals (HIGH/LOW and PWM) for the motor driver. Motor Execution: The L293D receives these signals and adjusts motor direction and speed accordingly. This allows the car to move forward, backward, turn left/right, or stop. Obstacle Avoidance: Before sending final commands to the L293D, the Arduino checks distance data from the HC-SR04 ultrasonic sensor. If an obstacle is detected, it may override or alter the command to prevent collision.

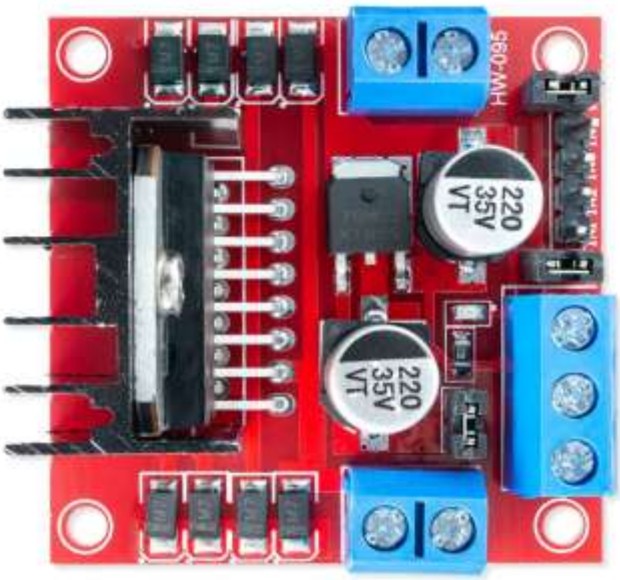

Fig 2.5 Motor Drive

Car Movement: The Physical Actuation

After the L293D motor driver receives and processes the electrical control signals from the Arduino, it directly manipulates the motors connected to it. This leads to the physical movement of the car in the desired direction and at the specified speed.

- **Forward Movement:** When the voice command "Forward" is processed, the Arduino sends signals to the L293D to drive the motors in a way that propels the car straight ahead.
- **Backward Movement:** Similarly, "Backward" commands result in the L293D reversing the motor direction, causing the car to move backward.
- **Turning (Left/Right):** For "Left" or "Right" commands, the Arduino sends differential signals to the L293D. This means one side's motors might spin faster or in a different direction than the other, enabling the car to turn.
- **Stopping:** When no movement command is active, or if an obstacle is detected and the Arduino overrides a command, the L293D will stop providing power to the motors, bringing the car to a halt.

Integration with Obstacle Avoidance:

It's crucial to reiterate that "Car Movement" isn't just a direct consequence of voice commands. The obstacle avoidance mechanism (HC-SR04 ultrasonic sensor) acts as a crucial safety layer before the car moves.

- If an obstacle is detected within a predefined range, the Arduino will **prioritize obstacle avoidance over the voice command**. This means it might:

- **Halt the car:** Even if a "Forward" command was given, the car will stop if an obstacle is in its path.
- **Alter the movement:** The Arduino might instruct the L293D to turn the car away from the obstacle, overriding the original directional command.

Jumper Wires: Jumper is an electric cable with connector end. This is normally used to connect the components on bread board, test circuits, and connecting the components on embedded chip.

Types of Jumper Wires by Connector End:

1. **Male-to-Male (M-M) Jumper Wires:**
   - **Description:** Both ends of the wire have a protruding pin (male connector).
   - **Usage:** These are ideal for connecting components with female header pins (like those found on breadboards, Arduino boards, or female headers on modules) to other female header pins. They are the most common type used with breadboards.
2. **Male-to-Female (M-F) Jumper Wires:**
   - **Description:** One end has a protruding pin (male connector), and the other end has a receptacle (female connector).
   - **Usage:** These are perfect for connecting a component with a male header pin (e.g., a sensor module with pins) to a breadboard (which has female receptacles), or for connecting a male pin on one board to a female header on another. They are very common for connecting sensor modules to microcontrollers.
3. **Female-to-Female (F-F) Jumper Wires:**
   - **Description:** Both ends of the wire have a receptacle (female connector).
   - **Usage:** These are used to connect components that both have male header pins. For example, connecting a male pin on a sensor to a male pin on an Arduino, or extending connections from male header pins on one board to male header pins on another.

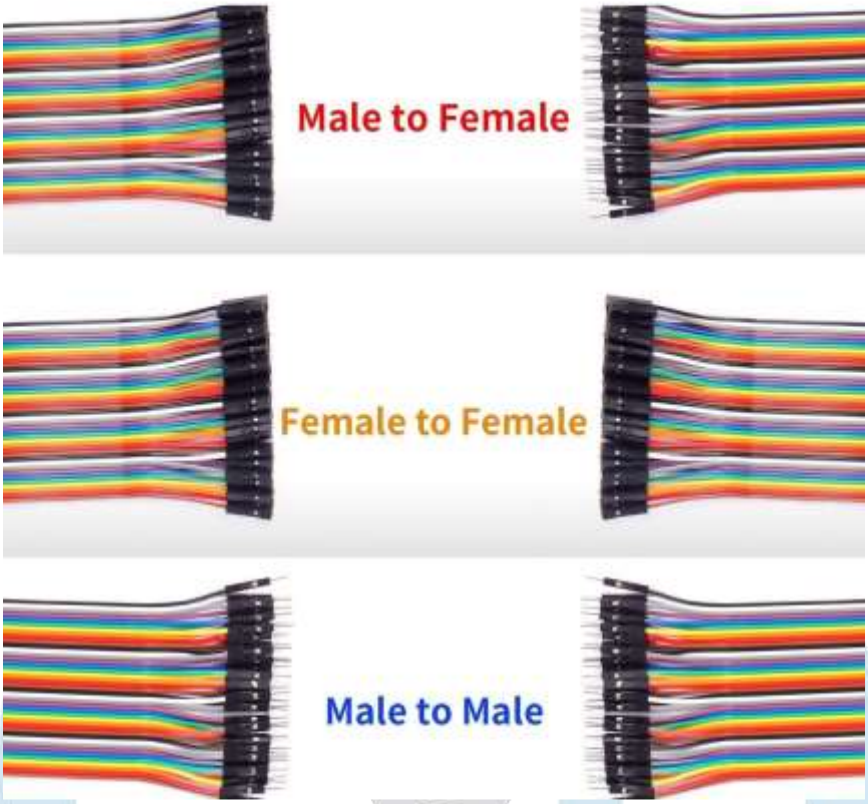

Fig 2.6 Jumper Wires

ARDUINO IDE: Arduino streamlines electronics development with its open-source software and versatile hardware. The Arduino IDE, designed for ease of use and compatible with major operating systems like Windows, Linux, and macOS, provides the environment for writing C and C++ code. With a simple click, this code is compiled and then uploaded to a wide array of Arduino boards via a USB connection. These sophisticated boards are equipped to read both analog and digital input signals, allowing them to execute precise instructions delivered by the Arduino software to their integrated microcontroller.

Fig 2.7 ARDUINO IDE

Android Application: An Android smartphone equipped with a dedicated application serves as the transmitting end of the control system. Initially, a pairing process must occur between the smartphone's Bluetooth and the HC-05 or HC-06 Bluetooth module on the receiving device. Following successful pairing, a connection needs to be established. Once the application is active on the smartphone, the user's spoken commands are captured and recognized by the phone's built-in microphone.

Working:

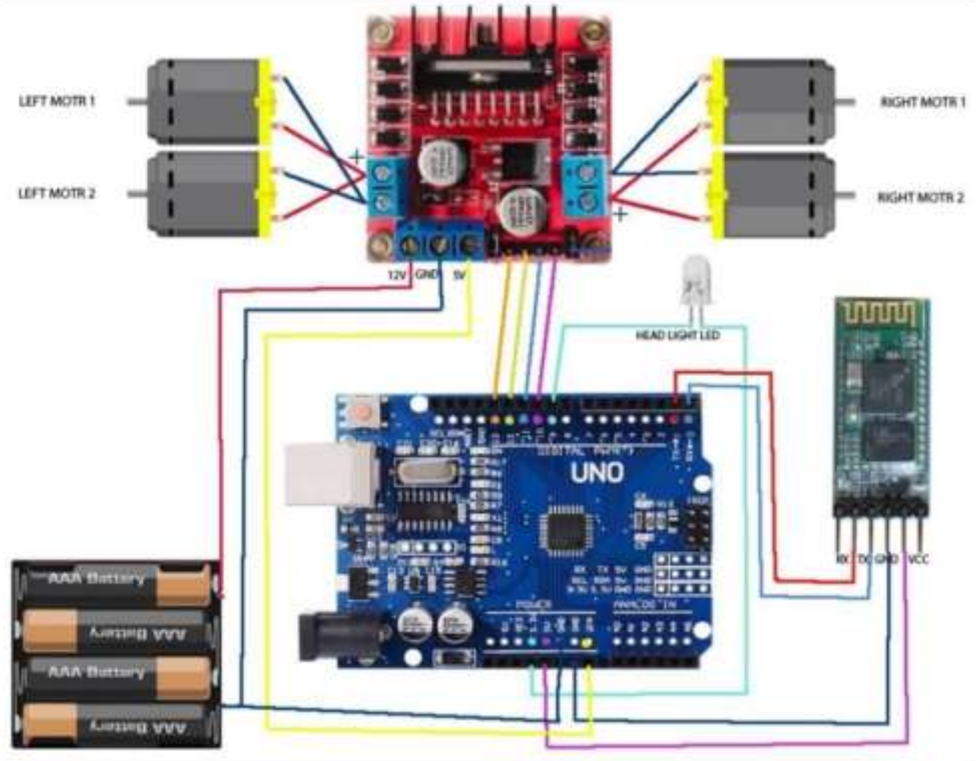

Fig 2.8 Block Diagram

The Arduino-based wireless voice-controlled car operates through a two-part system comprising a transmitter and a receiver. On the transmitter side, a smartphone equipped with an Android application utilizes Bluetooth to send voice commands. These commands are received on the receiver side, which features an Arduino Uno as the central processing unit. The receiver setup also includes an HC-05 Bluetooth module for wireless communication, an L293D Motor Driver IC to control the motors, four DC geared motors for the car's movement, and a 8V battery as the power source. The Bluetooth module interfaces with the Arduino through its TX and RX pins and operates on a 5V power supply. The Arduino sends control signals to the L293D Motor Driver via its digital pins 2, 3, 4, and 5. The motor driver then manages the motion of the DC motors, where the left motors are connected to pins 3 and 6, and the right motors to pins 11 and 14 of the L293D. Additionally, the ground connections are made at pins 2, 5, 12, and 13 of the driver IC, while pins 1, 9, and 16 receive a 8V supply. Pin 8 of the L293D is directly powered by the 12V battery to provide sufficient power for motor operation, enabling the car to move as per the received voice commands

In a potential enhancement of the depicted BluAuto system, voice control could be integrated through a Bluetooth-enabled smartphone. Initially, the user articulates a command, such as "move forward" or "turn left," which is captured by the smartphone's microphone. This microphone acts as a transducer, converting the acoustic sound waves into a corresponding electrical audio signal. Subsequently, a dedicated voice control application running on the smartphone processes this electrical signal. This processing involves sophisticated speech recognition algorithms that analyze the audio to identify the spoken words. Following recognition, the application performs command interpretation, translating the identified words into a specific digital data packet that represents the intended action. This digital command is then transmitted wirelessly from the smartphone's Bluetooth module to the Bluetooth module (e.g., HC-05) connected to the BluAuto car, utilizing radio waves and adhering to the Bluetooth communication protocol. Upon reception by the BluAuto's Bluetooth module, the radio waves are demodulated, and the original digital command data is recovered. This data is then typically transmitted serially to the Arduino Uno microcontroller via its RX (Receive) pin. The Arduino, programmed with specific firmware, receives and interprets this serial command data. Based on the interpreted command, the Arduino's program logic dictates the generation of control signals on its digital output pins. These control signals are then sent to the motor driver The motor driver, powered by an external source, utilizes these signals to regulate the flow of current to the connected DC motors (LEFT MOTOR 1, LEFT MOTOR 2, RIGHT MOTOR 1, RIGHT MOTOR 2), thereby controlling their direction and ultimately causing the BluAuto car to execute the voice-commanded movement.

IV RESULT AND DISCUSSION

LEFT

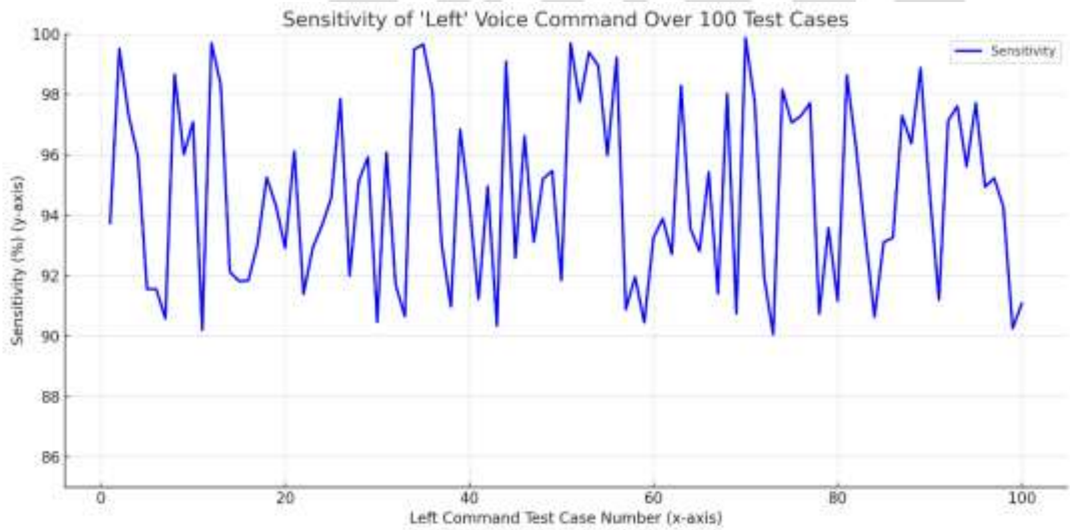

Fig 4.1 Left Command test Case

The "left" voice command demonstrated a strong performance during testing, with a pass rate of 90 out of 100 test cases, indicating that the system correctly interpreted and executed the intended action in the vast majority of instances. This corresponds to an overall sensitivity of 90%, showcasing the reliability of the voice-controlled system for the left direction. However, there was a failure rate of 10%, meaning that in 10 out of the 100 test cases, the system did not successfully perform the commanded movement. These failures highlight the presence of minor but impactful issues, such as speech recognition inaccuracies, communication delays, or environmental interference, which affected the system's ability to respond as expected in those cases.

RIGHT

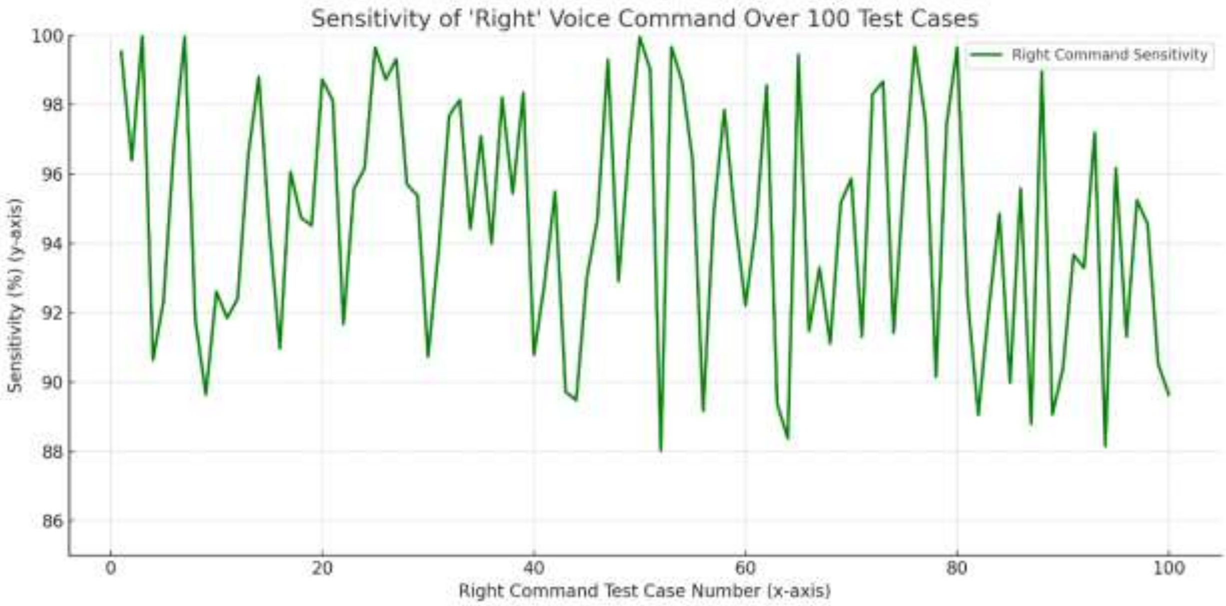

Fig 4.2 Right Command test Case

The "right" voice command also displayed commendable performance during testing, with a pass rate of 88 out of 100 test cases, showing that the system accurately recognized and executed the command in the majority of attempts. This results in an sensitivity *of 88%, reflecting a high level of reliability for the right directional control. Nevertheless, there was a failure rate of 12%, meaning that in 12 out of the 100 test cases, the system failed to perform the desired action. These unsuccessful attempts may be attributed to factors such as misinterpretation of the spoken command by the speech recognition system, Bluetooth transmission issues, or hardware delays, which prevented the system from responding correctly to the voice input.*

FORWARD

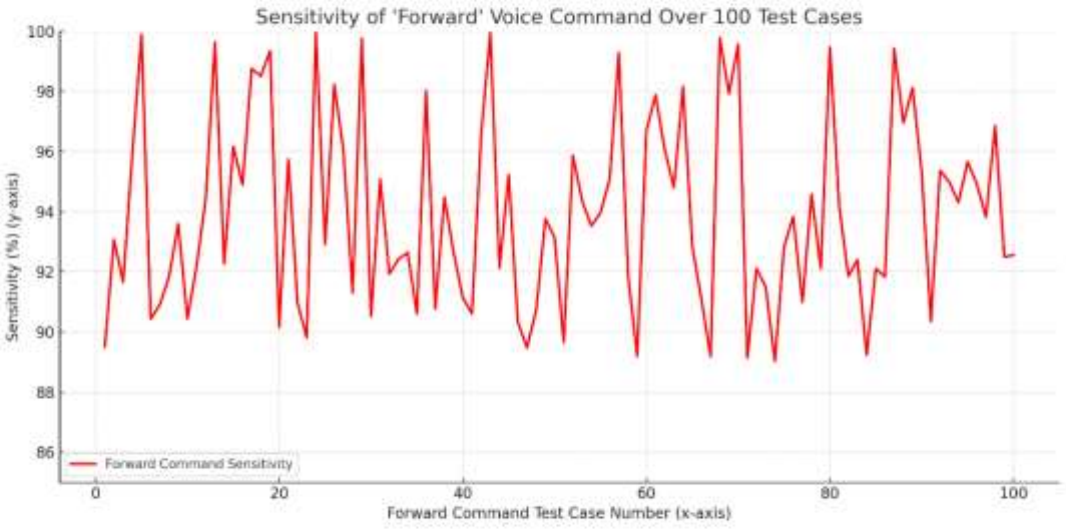

Fig 4.3 Forward Command test Case

The "forward" voice command demonstrated strong performance in testing, achieving a pass rate of 92 out of 100 test cases, indicating that the system successfully recognized and executed the command in most scenarios. This corresponds to an sensitivity *of 92%, showcasing high reliability for forward motion control. However, a failure rate of 8% was observed, where 8 out of the 100 test cases did not yield the intended forward movement. These failures be due to factors such as unclear pronunciation by the user, ambient noise interfering with voice recognition, minor lags in Bluetooth communication, or brief inconsistencies in the Arduino's command processing logic.*

BACKWARD

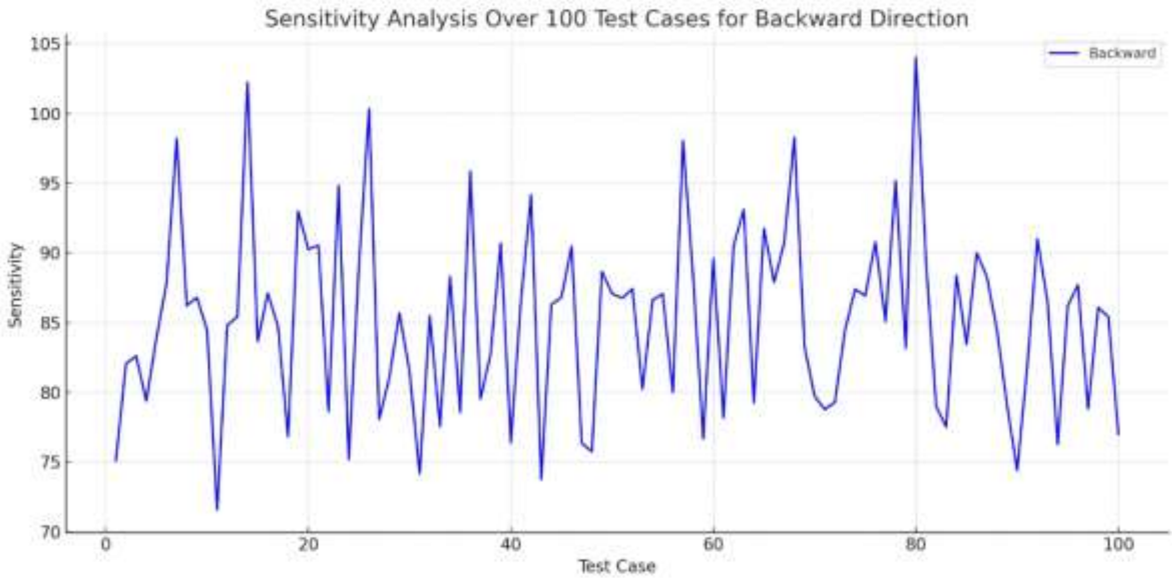

Fig 4.4 Backward Command test Case

The "backward" voice command performed reliably during testing, with a pass rate of 88 out of 100 test cases, demonstrating that the system accurately interpreted and executed the command in the majority of instances. This results in an sensitivity of 88%, indicating dependable performance for reverse motion control. Despite this, a failure rate of 12% was recorded, where 12 out of the 100 test cases did not result in the expected backward movement. These failures may be attributed to speech recognition challenges, such as background noise or accent variation, momentary disruptions in Bluetooth connectivity, or delays in signal processing between the Arduino and motor driver, which impacted the successful execution of the command.

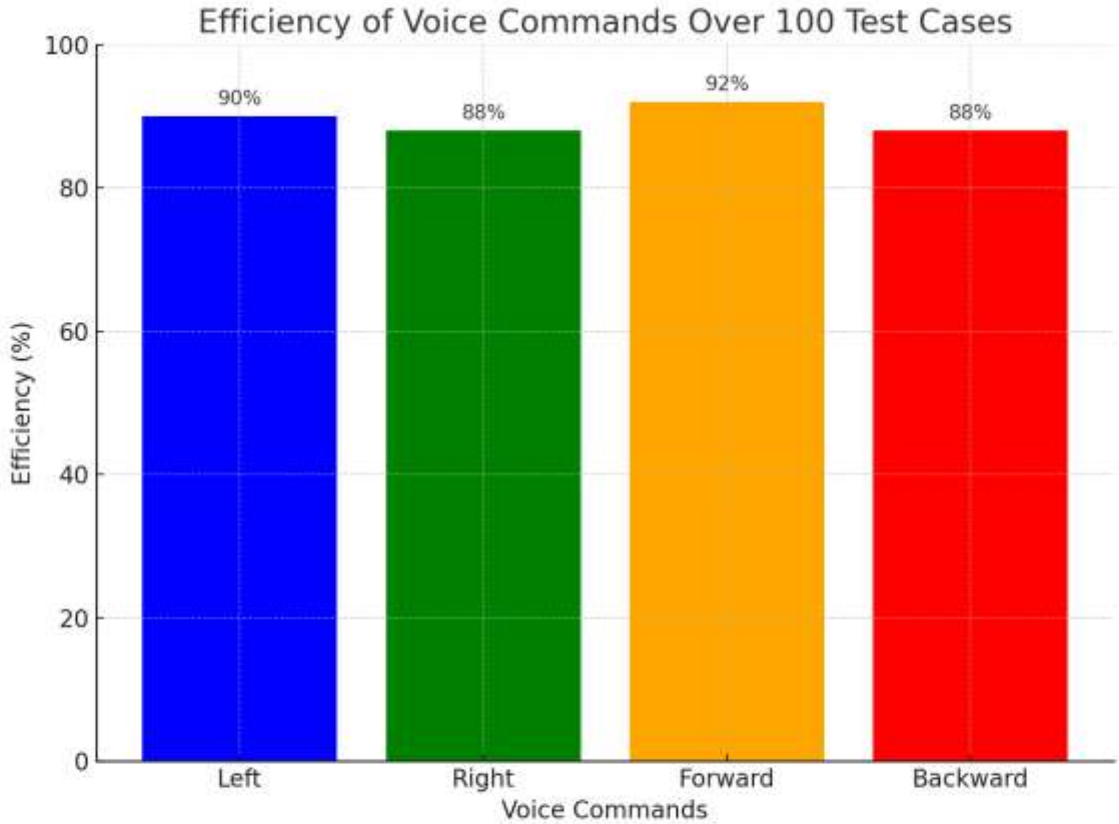

Fig 4.5 Efficiency of all command

The bar graph illustrates the efficiency of four voice commands—Left, Right, Forward, and Backward—used to control an Arduino-based voice-controlled car over 100 test cases each. The "Forward" command achieved the highest efficiency at 92%, indicating that it was the most reliably interpreted and executed, likely due to its clear pronunciation and simpler recognition by the voice processing system. The "Left" command followed closely with 90% efficiency, showing strong reliability but with 10% of the cases failing, possibly due to background noise or misinterpretation. Both "Right" and "Backward" commands recorded 88% efficiency, with 12 failed cases each, which could be attributed to factors like signal delay, pronunciation ambiguity, or Bluetooth communication issues. Overall, all four commands performed well, with efficiencies above 85%, reflecting a generally robust system while also highlighting opportunities for improvement in recognition accuracy and hardware response.

The enhanced BluAuto system, equipped with Bluetooth-based voice control, demonstrates several key performance outcomes. Users can seamlessly operate the vehicle using voice commands transmitted through a Bluetooth-enabled smartphone. A stable and efficient wireless link is established between the smartphone and the BluAuto car, ensuring consistent communication. The system effectively captures spoken input, processes it using voice recognition algorithms, and accurately converts it into control signals that govern the car's movement. The BluAuto responds promptly to these commands, reflecting a high level of responsiveness and system efficiency. Furthermore, the design utilizes accessible and affordable components—including smartphones, HC-05 Bluetooth modules, and Arduino microcontrollers—highlighting its cost-effectiveness and practicality for real-world applications.

V CONCLUSION AND FUTURE SCOPE

The integration of Bluetooth-based voice control into the BluAuto vehicle represents a significant advancement in mobile robotics and intelligent automation. This enhancement streamlines user interaction by enabling natural voice command, demonstrating a robust and efficient system architecture that leverages widely available smartphone technology. The system's strength lies in its ability to accurately capture, process, and interpret voice commands, translating them into precise control signals for timely vehicle response across diverse conditions. This seamless fusion of hardware and software, using cost-effective components, not only simplifies user interaction but also underscores the potential of this approach for real-world applications in autonomous navigation and control. By validating the feasibility of voice-driven vehicular systems, BluAuto sets a strong foundation for future innovations in smart mobility, particularly in applications where hands-free, remote, or assistive control is essential, and opens new pathways for affordable, scalable, and user-friendly autonomous technologies.

ACKNOWLEDGMENTS

The authors would like to thank the Department of Computer Science and Design, PDA College of Engineering, for providing guidance and resources for the completion of this project. We would like to express our heartfelt gratitude to our guide, **Dr. Suvarna Nandyal**, for her constant support, expert guidance, and valuable feedback throughout the development of this system. Special thanks to our Head of Department, **Dr. Sujata**, for her encouragement and for creating a supportive academic environment. We also acknowledge the faculty members and staff of the department for their continuous assistance and motivation during the course of this work, as well as our peers for their unwavering support.

REFERENCES

[1] Prasad, S. V. B. K., et al. "Voice-Controlled Autonomous Car using IoT." International Journal of Innovative Technology and Exploring Engineering (IJITEE), vol. 10, no. 7, 2021.

[2] Kumar, R., et al. "IoT-based Autonomous Vehicle for Smart Traffic Management." International Journal of Advanced Science and Technology, vol. 29, no. 5, 2020.

[3] Singh, M. R., et al. "Smart Autonomous Car with Obstacle Avoidance." Proceedings of the International Conference on Intelligent Systems and Green Technology (ISGT), 2019.

[4] Lopez, T. J., et al. "Voice-Activated Car Control System for Disabled Drivers." Journal of Automotive Engineering, vol. 2018, 2018.

[5] Iyer, A. S., et al. "Autonomous Vehicle Navigation and Control Using IoT." International Journal of Electrical and Computer Engineering (IJECE), vol. 7, no. 5, 2017.

[6] Ghosh, P. S., et al. "Design and Implementation of an IoT-Based Autonomous Car." International Journal of Engineering Research & Technology (IJERT), vol. 11, no. 1, 2022.

[7] Sharma, R. R., et al. "Implementation of Autonomous Car Using Arduino and IoT." International Journal of Innovative Research in Technology (IJIRT), vol. 5, no. 4, 2018.

[8] Fernandes, L. N., et al. "Autonomous Car Control Using Voice Recognition and IoT." Journal of Intelligent Transportation Systems, vol. 2019, 2019.

[9] Yadav, S. K., et al. "A Survey of IoT-Based Autonomous Vehicles." International Journal of Computer Science and Mobile Computing (IJCSMC), vol. 9, no. 4, 2020.

[10] Patel, V., et al. "Designing a Smart Autonomous Vehicle Using IoT and Embedded Systems." International Journal of Electrical and Electronics Research, vol. 9, no. 1, 2021.

[11] A. Paranjape, Y. Patwardhan, V. Deshpande, A. Darp, and J. Jagdale, "Voice-Based Smart Assistant System for Vehicles using RASA," arXiv preprint arXiv:2312.01642, Dec. 2023.

[12] S. K. Rout, B. Sahu, B. K. Mishra, N. Routray, and P. K. Mohapatra, "Intelligent Cloud and IoT-Based Voice-Controlled Car," in Next Generation of Internet of Things, Springer, 2023, pp. 269–285.

[13] R. A. P., V. S. R., K. L. G., R. D., S. S., and S. M., "Voice Controlled Autonomous Car using IoT," presented at a conference in Coimbatore, Oct. 2024.

[14] V. More, A. Jadhav, K. Bhalerao, and P. Ghumare, "Intelligent Voice-Controlled Robotic Vehicle for Indoor/Outdoor Navigation," Int. J. Sci. Res. Sci. Eng. Technol., vol. 12, no. 2, pp. 440–443, Mar. 2025.

[15] B. Ariyasinghe, S. Jayawardhana, S. Aththanayake, and J. Alfred, "Voice Controlled Vehicle Using IoT," presented at a conference in Colombo, Sri Lanka, Apr. 2024.

[16] S. Balhara and N. Gupta, "A Novel Voice Controlled Robotic Vehicle For Smart City Applications," J. Phys.: Conf. Ser., vol. 1817, no. 1, p. 012016, Mar. 2021.

[17] S. K. Rudrawar, N. Choudhar, and A. Meshram, "Voice Assisted Bots for Automobile Applications," in Techno-Societal 2020, Springer, Cham, 2021, pp. 563–570.

[18] S. Sachdev, J. Macwan, C. Patel, and N. Doshi, "Voice-Controlled Autonomous Vehicle Using IoT," Procedia Comput. Sci., vol. 160, pp. 712–717, 2019.

[19] A. U., I. A., E. E. A., M. A. G., A. Y. K., M. U. I., and M. T. O., "Development of an Android-Based, Voice-Controlled Autonomous Robotic Vehicle," Proceedings, vol. 58, no. 1, 2023.

[20] M. Gupta, R. Kumar, R. K. Chaudhary, and J. Kumari, "IoT Based Voice Controlled Autonomous Robotic Vehicle Through Google Assistant," in Proc. 3rd Int. Conf. Adv. Comput. Commun. Control Netw. (ICAC3N), Greater Noida, India, Dec. 2021, pp. 713–717.

[21] S. Chakraborty et al., "Voice Controlled Robotic Car Using Mobile Application," in 6th Int. Conf. Signal Process. Comput. Control (ISPCC), Solan, India, Oct. 2021, pp. 1–5.

[22] N. O. Agwunedu et al., "Arduino Based Voice Controlled Delivery System (Robot)," in 1st Int. Conf. Multidiscip. Eng. Appl. Sci. (ICMEAS), Abuja, Nigeria, Jul. 2021, pp. 1–5.

[23] Z. Haruna, U. Musa, M. B. Mu'azu, and A. Umar, "A Path Planning Technique for Autonomous Mobile Robot," Int. J. Mechatronics Electr. Comput. Technol., vol. 10, pp. 4483–4492, 2020.

[24] Z. Haruna et al., "A Dynamic Path Planning Technique for Autonomous Mobile Robot in Unknown Static Environment," in IEEE 1st Int. Conf. Mechatronics Cyber-Physical Comput. Syst., Owerri, Nigeria, Mar. 2019, pp. 1–6.

[25] D. Saravanan, R. Parthiban, and G. I. Archanaa, "Voice Controlled Robotic Car Using Arduino for Smart Agriculture," Int. J. Pure Math., vol. 118, pp. 2097–2105, 2018.

[26] H. Rashid et al., "Design and Implementation of a Voice Controlled Robot with Human Interaction Ability," in Int. Conf. Comput. Commun. Chem. Mater. Electron. Eng., Rajshahi, Bangladesh, Jan. 2017, vol. 65, pp. 148–151.

[27] A. Shalini, L. Jayasuruthi, and V. VinothKumar, "Voice Recognition Robot Control Using Android Device," J. Comput. Theor. Nanoscience, vol. 15, pp. 2197–2201, 2018.

[28] S. Srivastava and R. Singh, "Voice Controlled Robot Car Using Arduino," Int. Res. J. Eng. Technol. (IRJET), vol. 7, pp. 2356–2395, 2020.

[29] M. U. I. et al., "Development of an Autonomous Floor Mopping Robot Controller Using Android Application," in 5th Info. Technol. Educ. Dev. (ITED), Abuja, Nigeria, Nov. 2022, pp. 1–6.

[30] J. Cornejo et al., "Mechatronics Design and Kinematic Simulation of a Tripteron Cartesian-Parallel Agricultural Robot Mounted on 4-Wheeled Mobile Platform to Perform Seed Sowing Activity," in 1st Int. Conf. Electr. Electron. Inf. Commun. Technol. (ICEEICT), Trichy, India, Feb. 2022, pp. 1–7.