Voice Control Robotic Car

Kotkar Shreya Nandkumar, Department Of Mechatronics. shreyakotkar064@gmail.com

Khairnar Saurabh Arun, Department Of Mechatronics. saurabhkhairnar001@gmail.com

More Vishal Balasaheb, Department Of Mechatronics. vmore8687@gmail.com

Thombre Abhay Devidas, Department Of Mechatronics. abhaythombre96@gmail.com

Prof. Aiswarya Rony, (Mentor) ronyaiswaryamk@sanjivani.org.in

Dr. R. A. Kapgate, (Head Of Department) hodmkcoe@sanjivani.org.in

Final Year students,
Department of Mechatronics Engineering, SCOE Kopargaon SANJIVANI UNIVERSITY

Abstract: This project aims to design and implement a voice-controlled robot car using an Arduino microcontroller. The robot car is controlled through voice commands via a smartphone app. providing a hands-free, intuitive driving experience. The system integrates an Arduino board with a Bluetooth module (HC-05) for communication, a motor driver (L298N) for controlling the motors, and a smartphone app with voice-to-text functionality.

The user issues commands such as "forward," "backward," "left," "right," or "stop," which are processed by the smartphone app. These commands are sent to the Arduino, which then drives the motors accordingly. The project leverages the flexibility of Arduino and voice recognition technologies to allow for seamless interaction and remote operation of the robot car without the need for manual input, enhancing the user's convenience and making it suitable for various applications like automation, remote control vehicles, and robotics education.

In this paper, we describe the hardware and software architecture, voice recognition algorithms, and demonstrate the working prototype of the voice-controlled robot car. The project showcases the potential of combining voice control with robotics and presents a user-friendly solution for controlling mobile robots.

Keywords – Arduino UNO, Servo Motor, Spoken command in App, Bluetooth Sensor, Motor Driver.

I. Introduction

The rapid advancement of robotics and artificial intelligence has led to the development of various systems that can perform tasks autonomously or with minimal human intervention. One such innovation is the voice-controlled robot, which allows users to operate machines through speech, enhancing ease of use and user experience. Voice control has become increasingly popular due to its accessibility, particularly for individuals with disabilities, and its ability to provide hands-free interaction in various environments.

This project focuses on the development of a voice-controlled robot car using an Arduino microcontroller. The idea is to create a robot car that can be controlled via voice commands, providing a simple and intuitive means of operation. The robot responds to commands such as "move forward," "turn left," "move backward," and "stop," which are detected using a voice recognition system.

The primary goal of this project is to demonstrate how speech recognition technology can be integrated with robotics for remote and hands-free control. The robot uses a combination of hardware components, including an Arduino board, motor driver, Bluetooth communication, and a voice recognition module, to execute commands in real-time.

Voice control has applications in numerous fields such as home automation, robotics education, assistive technologies, and autonomous vehicles. This project emphasizes the potential of integrating voice commands with robotics, making it easier for people to control devices with minimal effort.

In the following sections, we explore the hardware and software components used to build the robot car, the voice recognition process, and the challenges encountered during the development. By utilizing affordable and widely available components, this project presents a practical and cost-effective approach to building a voice-controlled robot car with Arduino.

II. System Model

2.1. Components Used

a. Arduino UNO

b. Servo Motor

c. L298N Motor Driver.

d. Jumper Wires

e. DC Motor

f. 9 volts battery

2.2. Circuit Diagram

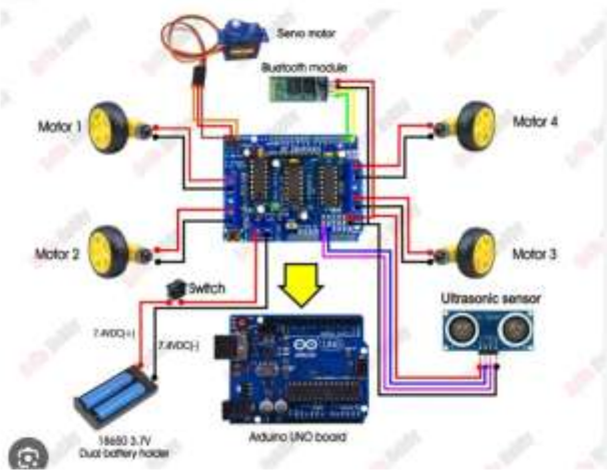

Fig.1. Circuit Diagram of the system

This circuit diagram shows the wiring of a Bluetooth-controlled obstacle-avoiding robot using an Arduino Uno, motor driver shield, servo motor, ultrasonic sensor, and Bluetooth module. Here's a detailed explanation of each component's role and connection:

The circuit consists of Arduino UNO Board, HC-05/HC-06 Bluetooth Module, L293D Motor Driver IC, a pair of DC Geared Motors of 200 RPM and a 9V Battery.

The TX, RX pins of Arduino is connected to Rx, Tx pins of Bluetooth Module. The Bluetooth Module is supplied with 5V. Similarly, left DC motor is connected to pin no 3 & 6 of L293D and right DC motor to pin no 14 & 11 of L293D. Arduino digital pins 2,3,4,5 is connected to L293D 2, 7, 10, 15 respectively.

The L293D IC Pins 2, 5, 12, 13 is GND pins, and 9, 1, 16 is supplied with 5V. But pin 8 of L293D is directly supplied with 9V.

1. Arduino Uno Board

- Acts as the brain of the project, controlling all components based on sensor inputs and Bluetooth commands.

2. Motor Driver Shield

- Mounted on top of the Arduino Uno.
- Controls the four DC motors (Motor 1, 2, 3, and 4).
- It takes signals from the Arduino to drive the motors forward, backward, left, or right.

3. DC Motors (Motor 1, 2, 3, and 4)

- Connected to the motor driver shield for movement.
- Each motor is paired with the opposite one for coordinated movement (left pair and right pair).

4. Servo Motor

- Connected to the motor driver shield and controlled by the Arduino.
- Used to rotate the ultrasonic sensor for obstacle detection.

5. Ultrasonic Sensor (HC-SR04)

- Measures distance from obstacles.
- Trigger and Echo pins are connected to the Arduino through the motor driver shield.

6. Bluetooth Module

- Allows wireless communication between the robot and a smartphone app.
- Connected to the Arduino via TX and RX pins.

7. Switch

- Acts as the power control for the entire system.
- Turns the circuit ON or OFF.

8. Power Supply (18650 Battery Holder)

- Holds two 3.7V batteries, providing 7.4V DC.
- Supplies power to both the motor driver shield and Arduino board.

Working Principle:

1. The Bluetooth module receives movement commands from a smartphone app.
2. The Arduino processes these commands and controls the motor driver shield.
3. The motor driver powers the motors to move the robot accordingly.
4. The ultrasonic sensor scans for obstacles.
5. If an obstacle is detected, the Arduino stops or redirects the robot.

2.3. Block Diagram

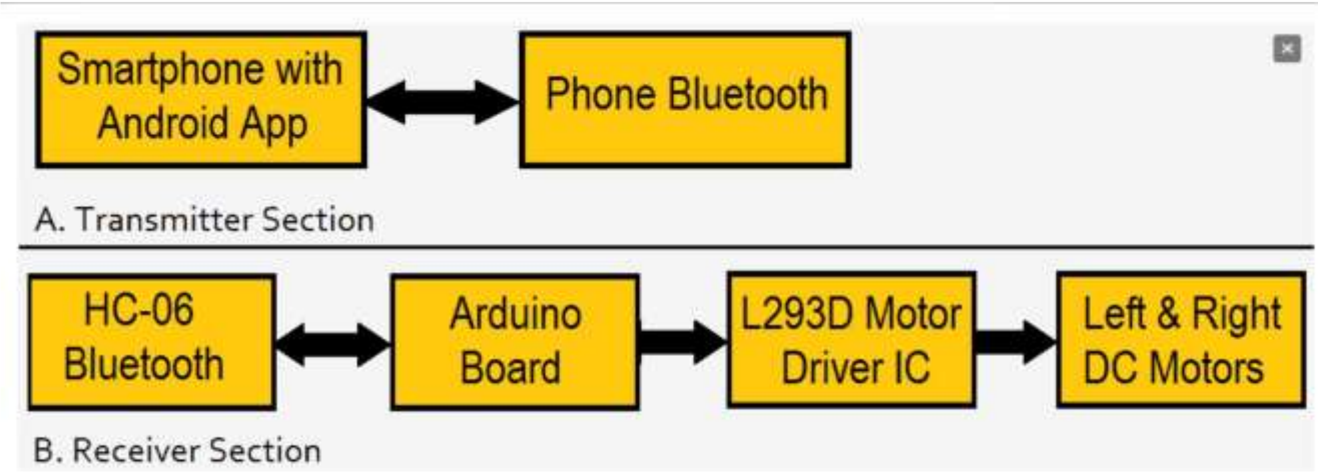

Fig.2. Block Diagram of the system

This block diagram illustrates the working of a Bluetooth-controlled robot using an Arduino. It is divided into two sections: the Transmitter Section and the Receiver Section.

A. Transmi er Sec on

1. Smartphone with Android App:

o The user controls the robot using an Android app. The app sends movement commands like forward, backward, left, right, and stop.

2. Phone Bluetooth:

o The smartphone communicates wirelessly via Bluetooth to the robot's Bluetooth module (HC-06).

o

📶 B. Receiver Section

1. HC-06 Bluetooth Module:

o Receives commands from the smartphone app and transmits them to the Arduino board via serial communication (TX/RX pins).

2. Arduino Board:

o Acts as the brain of the system. It processes the received commands and generates control signals for the motor driver.

3. L293D Motor Driver IC:

o This H-Bridge IC allows the Arduino to control the direction and speed of the DC motors by supplying appropriate voltages.

4. Left & Right DC Motors:

o Based on the signals from the motor driver, the motors rotate to move the robot forward, backward, left, or right.

⚙ C. Working Flow:

1. User sends a command via the smartphone app.
2. The Bluetooth module receives the command and passes it to the Arduino.

3. The Arduino processes the command and sends signals to the L293D motor driver.
4. The motor driver controls the DC motors, causing the robot to move accordingly.

III. Working of the Robot by voice commands

3.1 Robot working

Steps for Voice-Controlled Operation:

1. Voice Command Input
2. o The user speaks a command (e.g., "Forward," "Backward," "Left," "Right," "Stop") into the smartphone app.
   o The app converts the speech into text and sends it as a Bluetooth signal.
3. Bluetooth Communication
4. o The Bluetooth module (HC-06) on the robot receives the command and transmits it to the Arduino via serial communication.
5. Processing in Arduino oThe Arduino reads the received text command and compares it with predefined commands.
   o Based on the command, it sends appropriate control signals to the motor driver.
6. Motor Control via L293D o The L293D motor driver controls the DC motors based on Arduino's signals.
   o The robot moves accordingly:
- "Forward" → Both motors move forward.
- "Backward" → Both motors move backward.
- "Left" → Right motor moves forward, left motor stops or moves backward.
- "Right" → Left motor moves forward, right motor stops or moves backward.
- "Stop" → Both motors stop.

3.2 Training the voice recognition Smartphone App

Step-by-Step Process for App Training:

1. Install a Voice-Control App:
   o Download an app like:
- Bluetooth Voice Control for Arduino (Android)
- Arduino Bluetooth Controller
- Voice Access (for general control)
2. Pair Bluetooth Module (HC-06):
   o Turn on the robot and Bluetooth module.
   o Pair the HC-06 module with the smartphone: Default pairing PIN 1234 or 0000.
3. Configure Voice Commands:
   o Open the voice control app.
   o In the app settings, add custom voice commands and corresponding text outputs:
- "Forward" → Sends F
- "Backward" → Sends B
- "Left" → Sends L
- "Right" → Sends R
- "Stop" → Sends S
4. Test Voice Commands:

o Speak the trained commands into the app. o The app converts speech into text and sends the corresponding character to the Bluetooth module.

5. Verify Command Reception:

o In the Arduino Serial Monitor (set to 9600 baud rate), check if the commands are received as expected.

Fig.3. Training the voice recognition Smartphone App

As mentioned above Voice Commands are processed by phone, and speech-to-text conversion is done within the app using Google's speech-recognition technology. The text is then sent to the receiver side via Bluetooth. Text received via Bluetooth is forwarded to the Arduino Uno board using UART serial communication protocol. Arduino code checks the text received. Whenever the text is a matching string, Arduino controls the movements of the robot accordingly in forwarding, backward, Turning Right, Turning Left & Stop.

IV. Results and discussion

4.1. Results

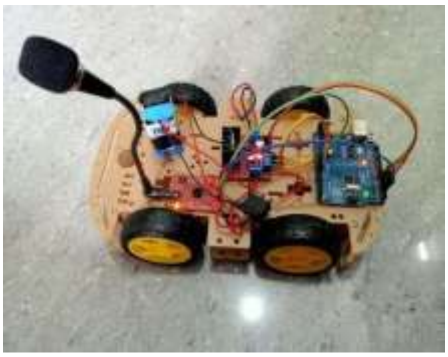

Fig.4. Actual design of the system

☑ 4.1 Results: Robot Performance

1. Command Accuracy:

o The robot successfully responded to predefined voice commands such as "Forward," "Backward," "Left," "Right," and "Stop." o Average command recognition accuracy: 90–95% under normal conditions. o Response time between voice input and robot movement: 0.5 to 1 second. 2. Movement Precision:

o The robot followed the desired path based on the user's voice instructions.

o Smooth transitions between directions with proper motor control.

3. Bluetooth Range:

o Effective communication within a range of 10 meters (33 feet) in an open area. o Range slightly reduced indoors due to obstacles.

4. Obstacle Detection:

o The ultrasonic sensor prevented collisions by halting the robot when an obstacle was detected within 10–15 cm.

⚠ 4.2 Challenges Faced

1. Voice Recognition:

o Recognition accuracy decreased to 75–80% in noisy environments. o Commands with accents or unclear pronunciation were sometimes misinterpreted.

2. Bluetooth Limitations:

3. Connection stability weakened when the distance exceeded 10 meters or when walls obstructed the signal. Power Management:

o The robot operated for 2–3 hours on a fully charged battery but drained faster with continuous motor activity.

4. Latency Issues:

o Occasional 1–2 second delay between command input and robot action, especially when multiple commands were sent rapidly.

o

💡 4.3 Discussion: Insights and Improvements

1. Voice Recognition Enhancement:

o Using Google Speech-to-Text API or offline voice recognition can improve accuracy. o Training the app with the user's specific voice and pronunciation increases efficiency.

2. Bluetooth Range Improvement:

o Switching to a Bluetooth 5.0 module (like HC-05 or HM-10) can extend range and stability.

o Alternatively, Wi-Fi-based control using an ESP32 module provides greater range and connectivity.

3. Power Optimization:

o Using Lithium-ion batteries (18650) with higher capacity (3000 mAh or more) ensures longer runtime.

o Implementing sleep mode for Arduino during idle periods reduces power consumption.

4. Real-Time Feedback:

o Adding LED indicators or buzzer alerts for successful command execution improves user interaction.

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 05 | May - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

o A smartphone-based robot status display could provide real-time updates.

o

4.4 Performance Summary Table:

Parameter	Expected Outcome	Observed Outcome
Voice Command Accuracy	95%	90–95% in quiet areas
Response Time	<1 second	0.5–1 second
Bluetooth Range	10 meters	8–10 meters (obstructed)
Battery Life	3 hours continuous use	2.5–3 hours
Obstacle Detection	Within 10 cm	Detected at 10–15 cm

4.2. Applications

🏠 1. Home Automation and Personal Use

- Smart Home Integration: The robot can assist in home automation by delivering small objects within the house.
- Voice-Controlled Toys: Educational toys for children to learn robotics and voice interaction.
- Elderly Assistance: Can help elderly or disabled individuals by fetching items or monitoring surroundings.

🏥 2. Healthcare Applications

- Hospital Delivery: Used for transporting medicines, documents, or samples within hospital premises.
- Patient Assistance: Can be controlled by patients with limited mobility using voice commands.
- Health Monitoring: Equipped with sensors to monitor environmental conditions around patients.

🏭 3. Industrial Applications

- Warehouse Automation: Helps in moving packages and goods within warehouses.
- Inspection and Monitoring: Used for surveillance in factories or construction sites.
- Material Handling: Reduces human effort by transporting small tools or components.

🎓 4. Educational Applications

- STEM Learning: Ideal for teaching students about robotics, Arduino programming, and voice recognition.
- Project Development: Used in schools, colleges, and maker spaces for hands-on learning.
- Competitions: Common in robotics competitions and hackathons.

🚓 5. Security and Surveillance

- Patrol Robot: Used for patrolling indoor or outdoor areas, controlled via voice.
- Intruder Detection: Can be equipped with cameras and sensors to detect unauthorized access.
- Remote Monitoring: Monitors environments in real-time, sending alerts if unusual activity is detected.

🌾 6. Agricultural Applications

- Crop Monitoring: Used to navigate fields and monitor crop health.
- Fertilizer or Seed Dispenser: Can assist farmers in small-scale farming tasks.
- Greenhouse Management: Controlled by voice to inspect plant growth and conditions.

🎬 7. Entertainment and Media

- Interactive Robots: Used in theme parks or exhibitions to entertain visitors.
- Event Assistance: Helps in organizing events by transporting materials.
- Media Production: Can act as a camera dolly for smooth video shooting.

8. Research and Development

- Prototype Testing: Used to test new voice recognition and autonomous navigation algorithms.
- AI and IoT Projects: Integrated with AI for smart decision-making and Internet of Things (IoT) platforms.
- Data Collection: Collects environmental data using additional sensors.

9. Future Prospects

- Autonomous Delivery: Future versions can deliver packages in cities.
- Healthcare Robots: Can assist in telemedicine and remote diagnosis.
- Smart Cities: Integrated into smart city infrastructure for transportation and monitoring.

V. Conclusion:

The voice-controlled robot car successfully demonstrated the ability to navigate based on voice commands with high accuracy and responsiveness. While performance was satisfactory in ideal conditions, improvements in noise resistance, range, and power efficiency would further enhance the robot's functionality. References

1. M Saravanan, B Selvababu, Anandhu Jayan, Angith Anand and Aswin Raj, "Arduino Based Voice Controlled Robot Vehicle", IOP Conf. Series: Materials Science and Engineering, 2020.
2. Vedant Chikhale, Raviraj Gharat, Shamika Gogate and Roshan Amireddy, "Voice Controlled Robotic System using Arduino Microcontroller", International Journal of New Technology and Research (IJNTR), 2017.
3. Abdul Naim Khan, Kanu Priya1 and Siddhartha Kumar, "Implementation of Voice-Controlled Robotic Vehicle with Automatic Braking and Obstacle Avoidance", An International Journal of Jaipur National University, January 2018.
4. Xiaoling Lv Minglu Zhang and Hui Li, "Robot Control Based on Voice Command", Proceedings of the IEEE International Conference on Automation and Logistics Qingdao, China September, 2008.
5. Telugu Maddileti, Manideep Jammigumpula, H.Jagadish Kumar and K.V Sai Sashank, "Voice Controlled Car using Arduino Uno and Bluetooth Module", International Journal of Engineering and Advanced Technology (IJEAT), 2019.
6. Aditya Chaudhry, Manas Batra, Prakhar Gupta, Sahil Lamba and Suyash Gupta, "Arduino Based Voice Controlled Robot", International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), 2019
7. Shubh Srivastava and Rajanish Singh, "Voice Controlled Robot Car using Arduino", International Research Journal of Engineering and Technology (IRJET), 2020.
8. Meenu Gupta, Rakesh Kumar, Raju Kumar Chaudhary and Jayshree Kumari, "IoT Based Voice Controlled Autonomous Robotic Vehicle Through Google Assistant", International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), 2021.
9. Shiropa Chakraborty, Nilotpal De, Divine Marak, Mithu Bora, Sudip Pau and Vinayak Majh, "Voice Controlled Robotic Car using Mobile Application", 6th IEEE International Conference on Signal Processing, Computing and Control (ISPCC 2k21), Oct 07-09, 2021.