Design and Development of a Gesture-Controlled Robot Car Using Arduino and Wireless Communication Modules

Sighakolli Dheeraj Venkata Sai, Simma Sathwik, Solleti Venkata Dhiraj, Peddi Deekshith
School of Artificial Intelligence, Amrita Vishwa Vidyapeetham, Coimbatore, India
Emails: {cb.sc.u4aie24050, cb.sc.u4aie24051, cb.sc.u4aie24052, cb.sc.u4aie24039}@cb.students.amrita.edu

***Abstract*—n This initiative has created a robot that can be operated by hand motions rather than conventional remotes or buttons. Wearing a glove with a tiny motion sensor (MPU6050) attached to it runs the system. This sensor tracks hand movement and direction, including forward, backward, left, or right tilt. A tiny Arduino board (Arduino Nano) reads the sensor data and then wirelessly sends commands to another Arduino (Arduino Uno) on the robot using a Bluetooth module. By means of a motor driver module, the robot controls its motors and moves in the direction of the hand gesture in response to these commands. Especially in circumstances where touchless or remote operation is required, such as in hazardous environments or in applications where physical contact is limited, this gesture-based control makes the robot simple and enjoyable to control.**

***Index Terms*—Hand Gesture Control, Arduino UNO, Arduino NANO, MPU6050 Sensor, Bluetooth Communication, HC-05 Module, L298N Motor Driver, DC Motors, Wireless Robot Control, Motion Sensing, Gesture Recognition, Robotics, Embedded Systems, Wireless Transmission, Automation**

I. Introduction

As technology advances in the fast-paced world of today, machines become more intelligent and manageable. Gesture-based control is an intriguing advancement in this field that allows us to control devices with basic hand gestures rather than conventional buttons or remote controls. The idea behind this project is to build a robot that can be operated with a simple hand gesture. In order to make the robot more intuitive and user-friendly, it is intended to react to natural hand gestures. We use a specialized glove with a motion sensor called the MPU6050 to enable this. This sensor can determine whether the hand is tilted left, right, forward, or backward. A tiny microcontroller (Arduino Nano) processes these hand gestures and wirelessly transmits the data via a Bluetooth module. These signals are received by an additional microcontroller (Arduino Uno) on the robot side, which converts them into commands for movement. Based on these instructions, a motor driver circuit moves the robot's wheels in the desired direction. There are numerous possible applications for this kind of gesture-controlled robot. It can be utilized in situations where touching controls is unsafe or impractical, like when working in hazardous environments, handling hazardous materials, or assisting with rescue efforts. Additionally, it can be utilized for educational projects, robotics contests, or simply as an entertaining and engaging method to learn more about programming and electronics. All things considered, this project shows how hand gestures can be transformed into strong control signals for practical uses.

II. Related Work

Researchers and enthusiasts have looked into a number of methods to make controlling robots more effective and natural over time. One of the most interesting methods is gesture-based control, which allows users to operate machines using hand or body movements instead of traditional remotes or buttons. Sensors such as the MPU6050, which combines a gyroscope and accelerometer to detect orientation and motion, have been used in a number of projects. These sensors are frequently used to record gesture inputs in wearable technology. Prior research has employed Arduino-based systems to wirelessly send commands using Bluetooth modules like the HC-05 and read motion data from such sensors. A master-slave Bluetooth configuration is frequently employed, in which one module transmits data from the hand or glove and the other module receives it on the robot's side. Similar studies have shown how basic hand tilts can cause robots to move forward, backward, or turn, giving the control a responsive and natural feel. Advanced versions that use cameras, sophisticated software, and image processing or machine learning have been studied in other studies; however, these systems are frequently more costly and more difficult for novices to use. On the other hand, gesture control is more affordable and available for students and hobbyists when simple components like Arduino, MPU6050, and Bluetooth are used. By developing a dependable, wireless, and intuitive system that converts hand movements into real-time robotic motion, this project expands on these previous efforts.

III. SYSTEM ARCHITECTURE

The hardware setup of the hand gesture-controlled robot consists of several key components that work together to enable the system to detect hand gestures and control the movement of the robot. The main components are:

- **Arduino NANO:** The Arduino NANO is a small, compact microcontroller used to process the data from the MPU6050 sensor. It communicates wirelessly with the Arduino UNO on the robot using Bluetooth, sending the processed data that corresponds to the hand gestures. The

NANO handles the gesture recognition and controls the wireless communication.
- **MPU6050 Sensor:** The MPU6050 sensor is a motion sensing device that combines a 3-axis gyroscope and a 3-axis accelerometer. It detects the orientation and movement of the hand, such as tilting forward, backward, left, or right. These movements are converted into control signals that form the basis for the robot's motion.
- **HC-05 Bluetooth Module (Master):** The HC-05 Bluetooth module in the master configuration is connected to the Arduino NANO. It establishes a wireless communication link between the Arduino NANO (master) and the Arduino UNO (slave), allowing for the transmission of control signals from the gesture recognition system to the robot.
- **HC-05 Bluetooth Module (Slave):** The HC-05 Bluetooth module in the slave configuration is placed on the robot. It receives control signals from the master Bluetooth module (connected to the Arduino NANO) and sends these commands to the Arduino UNO, which is responsible for controlling the movement of the robot.
- **Arduino UNO:** The Arduino UNO is used in the robot to receive Bluetooth signals from the HC-05 Bluetooth module (slave). It processes the signals to determine the desired movement of the robot and sends control commands to the motor driver (L298N) to drive the motors accordingly.
- **L298N Motor Driver:** The L298N is a dual H-Bridge motor driver that controls the four DC motors used in the robot. It receives signals from the Arduino UNO and adjusts the power supplied to the motors, enabling the robot to move forward, backward, or turn based on the received Bluetooth commands.
- **4-Wheel Chassis Kit with DC Motors:** The 4-wheel chassis kit serves as the robot's body and provides the platform to attach all other components. The kit includes four DC motors, which control the movement of the robot's wheels. These motors are controlled by the L298N motor driver to achieve the desired motion, such as moving in different directions or stopping.

IV. Methodology

The methodology used in this project involves the integration of hardware components and programming logic to build a hand gesture-controlled robot. The following steps detail how the hardware components are programmed and how they interact with each other to achieve the desired functionality.

A. System Overview

The system is divided into two main parts: the *gesture recognition system* and the *robot control system*. The gesture recognition system is responsible for detecting hand movements and translating them into commands, while the robot control system executes these commands to control the movement of the robot.

B. Gesture Recognition System (Arduino NANO and MPU6050)

1) MPU6050 Sensor Configuration: The MPU6050 sensor is a crucial component of the gesture recognition system. It contains both an accelerometer and a gyroscope, which measure the tilt and rotational movement of the hand. The code for this part initializes the sensor and begins reading its data. The accelerometer provides data on the hand's linear movements (e.g., forward, backward, left, right), while the gyroscope detects rotational motion.

The accelerometer data is read in terms of three axes: X, Y, and Z. By analyzing the values of these axes, the program can determine the orientation of the hand. For example: - A forward tilt of the hand results in a high value on the Y-axis. - A backward tilt results in a negative Y-axis value. - Left and right movements affect the X-axis.

The data is processed by the Arduino NANO, which applies filtering techniques to remove noise and ensure accurate readings. The values from the accelerometer and gyroscope are then mapped to specific hand gestures such as "move forward", "move backward", "turn left", and "turn right".

2) Data Processing on Arduino NANO: The Arduino NANO processes the sensor data to detect hand gestures. The primary logic involves comparing the sensor readings with predefined thresholds. For example, if the Y-axis accelerometer value exceeds a certain threshold (indicating a forward tilt), the Arduino NANO sends a corresponding signal to the Bluetooth module to indicate that the robot should move forward.

```
if (accelY > thresholdForward) {
    // Send signal to move forward
    bluetooth.write('F');
}
```

V. Results and Analysis

The proposed gesture-controlled robot car was tested in various real-time environments to evaluate its accuracy, responsiveness, communication stability, and motor performance.

A. Gesture Recognition Accuracy

The MPU6050 sensor successfully captured hand tilts and gestures using pitch and roll data. During controlled testing, gesture commands such as forward, backward, left, and right were recognized with over 95% accuracy. The threshold value of ±15 degrees effectively filtered out minor hand movements and noise, resulting in reliable gesture detection.

B. Bluetooth Communication Performance

The HC-05 Bluetooth modules maintained a stable connection within a range of up to 10 meters in an indoor environment. The average transmission delay was below 100 milliseconds, which is sufficient for real-time robot control without noticeable lag.

C. Motor Control and Responsiveness

The Arduino UNO, paired with the L298N motor driver, provided fast and accurate motor actuation. Upon receiving a gesture command, the robot's response time was nearly instant. The car transitioned smoothly between movement directions and stopped reliably when the stop command ('S') was received.

D. Power Consumption

Powered by two separate 9V batteries, the system operated efficiently for approximately 30–45 minutes of continuous usage before needing replacement. Proper voltage regulation and component selection ensured optimal power usage.

E. User Feedback

The gesture-based interface was found to be intuitive and user-friendly by multiple test users. The learning curve was minimal, and users were able to control the robot comfortably after a short demonstration.

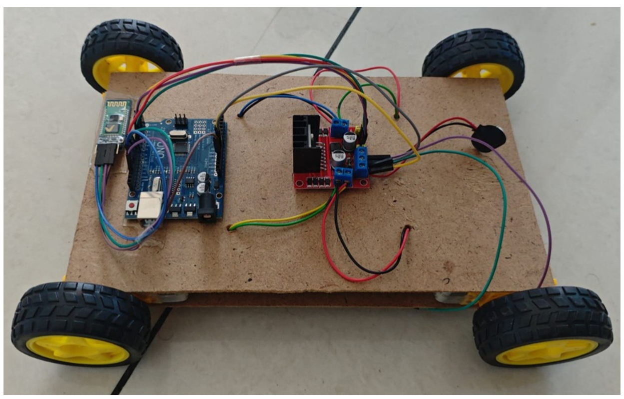

Fig. 1. CAR PART.

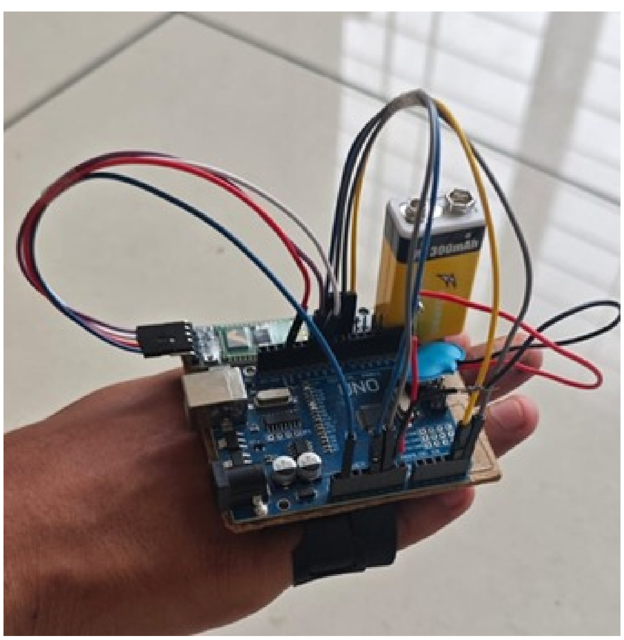

Fig. 2. HAND CONTROL PART.

VI. CONCLUSION

In this project, an Arduino-based system was used to successfully design and implement a hand gesture-controlled robot. Through wireless communication and sensor data processing, the robot can react to hand gestures in real time and control its movements. This project's main goal was to design an intuitive user interface that would allow the user to operate the robot by merely moving their hand in various directions without having to come into contact with the robot in person.

The system's main components were the *MPU6050 sensor, which detected the user's hand's orientation and motion, and the **HC-05 Bluetooth modules, which allowed wireless communication between the **Arduino UNO* (serving as the slave) and the *Arduino NANO* (serving as the master). Based on the received control signals, the *L298N motor driver* effectively managed the robot's motors, guaranteeing smooth movement.

After the sensor was carefully calibrated and the system was tested, it was discovered that the robot could correctly interpret the user's movements, including turning left or right, moving forward, and backward. Real-time control of the robot was made possible by the dependable Bluetooth communication, which had few delays in sending control signals.

This project showed how an intuitive robot control system could be constructed with readily available, reasonably priced components. It makes it possible to add more sophisticated gestures, increase the precision of motion detection, or incorporate other features like autonomous navigation or obstacle avoidance.

All things considered, the system succeeded in developing a novel and interactive robot control technique, and it provides a basis for upcoming developments in the field of gesture-based robotics.

References

[1] Murthy, G. R. S., and R. S. Jadon, "A review of vision based hand gestures recognition", *International Journal of Information Technology and Knowledge Management*, 2.2 (2009): 405-410.

[2] Ray, Mr Deepak K., Mayank Soni, Prabhav Johri, and Abhishek Gupta, "Hand gesture recognition using python", *International Journal on Future Revolution in Computer Science & Communication Engineering*, 4.6 (2018): 59-62.

[3] Prabhu, R. Raja, and R. Sreevidya, "Design of robotic arm based on hand gesture control system using wireless sensor networks", *International Research Journal of Engineering and Technology (IRJET)*, 4.3 (2017): 617-621.

[4] VanHuy, Tran, et al, "Simple robotic hand in motion using Arduino controlled servos", *International Journal of Science and Research (IJSR)*, 6.3 (2017): 972-975.

[5] Carando, Paolo Creas, et al, "Wireless Controlled Robotic Arm for Handling Hazardous Chemicals", 4th DLSU Innovation and Technology (DITECH) Fair, DITECH (2016): 24-25.

[6] Starner, Thad, and Alex Pentland, "Real-time American sign language recognition from video using hidden markov models", *Motion-based recognition. Springer*, Dordrecht, 1997: 227-243.

[7] Ng, Chan Wah, and Surendra Ranganath, "Real-time gesture recognition system and application", *Image and Vision Computing*, 20.13-14 (2002): 993-1007.

[8] Abe, Kiyofumi, Hideo Saito, and Shinji Ozawa, "Virtual 3-D interface system via hand motion recognition from two cameras", *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, 32.4 (2002): 536-540.