International Journal of Science and Research Archive
eISSN: 2582-8185
Cross Ref DOI: 10.30574/ijsra
Journal homepage: https://ijsra.net/

(Research Article)

Gesture-controlled robotic arm using MPU6050 IMU and embedded microcontroller an embedded system approach for real-time human–robot interaction

Shreya Singh *, Vipin Kumar Modi, Sumit Shukla, Abhishek Kumar, Saumya Pandey and Nadeem Ahmad

Department of Electronics and Communication Engineering, Goel Institute of Technology and Management, Lucknow, India.

International Journal of Science and Research Archive, 2026, 19(01), 232-238

Publication history: Received on 21 February 2026; revised on 31 March 2026; accepted on 3 April 2026

Article DOI: https://doi.org/10.30574/ijsra.2026.19.1.0641

Abstract

This paper presents the design and development of a real-time gesture-controlled robotic arm using an inertial measurement unit (IMU) sensor and an embedded microcontroller platform. The system utilizes the MPU6050 sensor to capture hand motion data, including accelerometer and gyroscope readings, which are processed by an Arduino Nano to control the movement of a multi-degree-of-freedom robotic arm. The proposed system translates human hand gestures into corresponding robotic arm movements, enabling intuitive human–robot interaction [1, 2]. The architecture focuses on low-cost implementation, real-time response, and ease of integration. Experimental results demonstrate reliable motion tracking and accurate gesture-based control, making the system suitable for applications in assistive robotics, pick-and-place operations, and industrial automation. The developed prototype highlights the potential of wearable motion-sensing technology for natural and efficient robotic manipulation.

Keyword: Gesture recognition; Robotic arm; MPU6050; Arduino Nano; Embedded systems; Human–Machine interaction

1. Introduction

The interaction between humans and robotic systems has progressively shifted from rigid, hardware-dependent control mechanisms toward more intuitive and adaptive interfaces. As robotic manipulators become increasingly integrated into industrial automation, medical assistance, and hazardous environment operations, the demand for natural control strategies has intensified. Enabling robots to respond directly to human gestures represents a significant step towards reducing operational complexity and improving accessibility [3]. Conventional robotic arm control typically relies on predefined programming, mechanical switches, or external control panels. While effective in structured environments, such approaches often require technical expertise and limit real-time adaptability. In contrast, gesture-based control systems aim to translate human hand movements into similar actions on the robot arm without intermediary mechanical interfaces. This method not only enhances user convenience but also improves responsiveness in dynamic operational scenarios.

Recent progress in embedded system design and miniaturized sensing components has enabled the development of compact and economically viable motion-capture interfaces. Microcontroller platforms such as Arduino provide a practical foundation for acquiring sensor data, executing control algorithms, and driving actuators within a unified hardware framework. By integrating inertial sensing elements directly with a wired communication interface, hand orientation and movement data can be transmitted to a robotic manipulator with minimal delay. This direct signal transmission approach enhances synchronization between human motion and robotic response while reducing susceptibility to communication instability. Compared to vision-based gesture recognition systems, sensor-driven architectures eliminate dependence on external cameras and lighting conditions, resulting in consistent performance

---

[*] Corresponding author: Shreya Singh

and lower computational requirements. Although gesture-based control systems have advanced considerably, precise replication of human motion remains a technical challenge. Measurement inaccuracies due to sensor noise, cumulative drift in inertial readings, and mechanical limitations of servo actuators can affect positional accuracy. Additionally, coordinating multiple joints smoothly within a multi-degree-of-freedom manipulator demands efficient signal processing and calibrated motion mapping. Balancing performance accuracy with hardware simplicity and cost constraints requires careful system-level integration.

This study presents the design and implementation of a wired gesture-controlled robotic arm that mirrors human hand movements through real-time motion sensing and embedded processing. Orientation data acquired from a wearable sensing module are processed by a microcontroller-based control unit, which generates corresponding control signals for servo-driven joints. The system architecture emphasizes deterministic response, reduced latency, and stable gesture-to-motion translation without reliance on wireless transmission. The primary contribution of this work is the realization of a low-latency, wired control framework capable of reliable and repeatable gesture-based manipulation. The developed system demonstrates the feasibility of implementing intuitive robotic control using cost-effective components, with potential applications in assistive devices, educational robotics platforms, laboratory automation, and controlled industrial environments.

2. Proposed approach

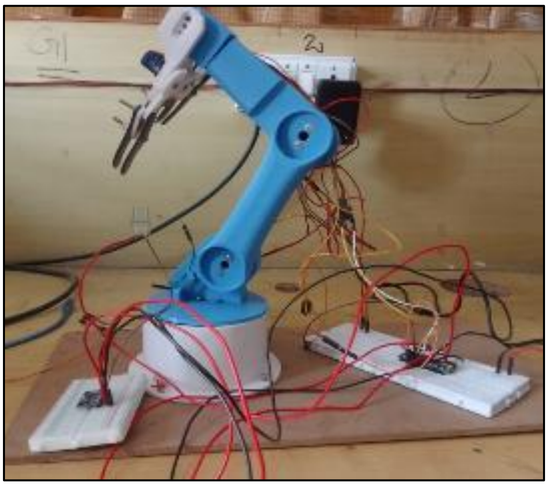

Figure 1 The gesture robotic arm assembled and connected

The proposed system is designed to translate human hand gestures into corresponding movements of a multi-degree-of-freedom robotic arm through real-time sensor acquisition and embedded processing. The overall architecture consists of a wearable sensing module, a microcontroller-based control unit, and a servo-actuated robotic manipulator. The sensed motion parameters are processed and directly mapped to joint actuation signals using a wired communication framework to ensure deterministic response and minimal latency.

2.1. System Architecture

The control strategy is based on capturing both hand orientation and finger bending information. Orientation data are obtained using an inertial measurement unit, while finger motion is detected through resistive flex sensing. The processed signals are interpreted by the microcontroller, which generates pulse-width modulation (PWM) outputs to drive the servo motors controlling the robotic joints.

2.2. Hardware Components

2.2.1. Inertial Measurement Unit

Hand orientation and angular motion are measured using the MPU6050 sensor module. This device integrates a three-axis accelerometer and a three-axis gyroscope within a single chip, enabling measurement of linear acceleration and angular velocity along three orthogonal axes[4]. The sensor communicates with the microcontroller via the I^2C protocol, allowing efficient real-time data acquisition. The combined accelerometer and gyroscope readings facilitate estimation of hand tilt and rotational movement required for robotic arm positioning.

2.2.2. Flex Sensor

Finger bending motion is detected using a resistive flex sensor. The resistance of the sensor varies proportionally with the degree of bending. By incorporating the sensor into a voltage divider configuration, analog voltage variations corresponding to finger movement are obtained and fed into the analog input pins of the microcontroller. This mechanism enables control of gripping or claw movement in the robotic arm.

2.2.3. Microcontroller Unit

The control logic is implemented using the Arduino Nano, which is based on the ATmega328P microcontroller [5]. The board is selected due to its compact size, sufficient analog and digital I/O pins, PWM capability, and low power consumption. The microcontroller performs sensor data acquisition, signal conditioning, gesture interpretation, and servo control signal generation.

2.2.4. Servo Actuators

Joint movement of the robotic arm is achieved using a combination of MG995 high-torque servo motors and micro servo motors. The MG995 servos are employed in joints requiring higher torque output, such as the base or shoulder mechanism, while micro servos are utilized for lighter joints and end-effector movement. These servo motors operate using PWM control signals generated by the microcontroller, enabling precise angular positioning within a defined range.

2.2.5. Power Supply Unit

A regulated 5V power supply is used to provide stable operating voltage to the microcontroller and servo motors. Proper power regulation is essential to prevent voltage fluctuations that may affect sensor accuracy and actuator performance. The power system is designed to handle the current requirements of multiple servo motors operating simultaneously.

2.3. Working

The system operates by capturing real-time hand motion and converting it into corresponding robotic arm movement through embedded signal processing. The MPU6050 continuously measures angular velocity and acceleration along three axes to determine hand orientation, while the flex sensor detects finger bending through resistance variation. These sensor outputs are read by the Arduino Nano, where the raw data are processed and mapped to predefined angular positions. Based on the interpreted gesture, the microcontroller generates pulse-width modulation (PWM) signals to control the servo motors, including MG995 units for high-torque joints and micro servos for lighter movements [6, 7]. As the user moves their hand or bends their fingers, the robotic arm replicates the motion in real time through proportional joint actuation. The direct wired communication ensures minimal delay and stable synchronization between human gesture and robotic response [8, 9, 10].

3. Block diagram

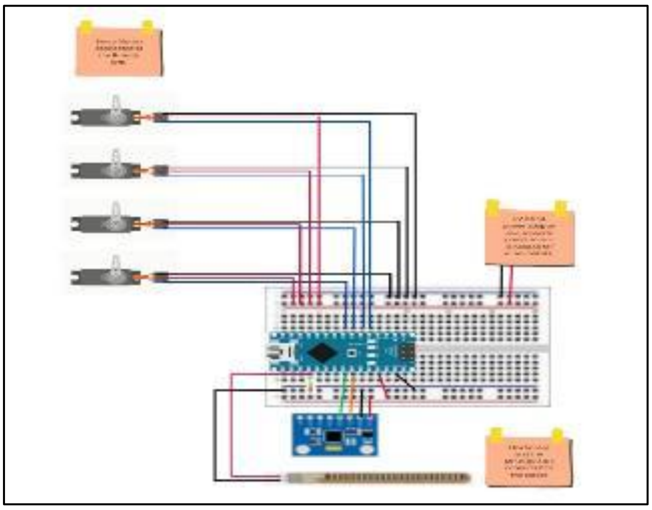

Figure 2 The Block Diagram of the robotic arm

4. CODE

```
// Gesture Controlled Robotic ARM (Arduino Nano & MPU6050) with Flex Sensor

#include <Wire.h>

#include <Servo.h>

Servo servo_1;

Servo servo_2;

Servo servo_3;   // Gripper controlled by flex sensor

Servo servo_4;

const int MPU_addr = 0x68;

int16_t axis_X, axis_Y, axis_Z;

int minVal = 265;

int maxVal = 402;

double x, y, z;

const int flexPin = A0;   // Flex sensor connected to A0

void setup() {

Serial.begin(9600);

Wire.begin();

Wire.beginTransmission(MPU_addr);

Wire.write(0x6B);

Wire.write(0);

Wire.endTransmission(true);

 servo_1.attach(2);

 servo_2.attach(3);

 servo_3.attach(4);

 servo_4.attach(5);

}

void loop() {

 // Read MPU6050

Wire.beginTransmission(MPU_addr);
```

```
Wire.write(0x3B);

Wire.endTransmission(false);

Wire.requestFrom(MPU_addr, 14, true);

axis_X = Wire.read() << 8 | Wire.read();

axis_Y = Wire.read() << 8 | Wire.read();

axis_Z = Wire.read() << 8 | Wire.read();

int xAng = map(axis_X, minVal, maxVal, -90, 90);

int yAng = map(axis_Y, minVal, maxVal, -90, 90);

int zAng = map(axis_Z, minVal, maxVal, -90, 90);

x = RAD_TO_DEG * (atan2(-yAng, -zAng) + PI);

y = RAD_TO_DEG * (atan2(-xAng, -zAng) + PI);

z = RAD_TO_DEG * (atan2(-yAng, -xAng) + PI);

 // -------- FLEX SENSOR CONTROL --------

int flexValue = analogRead(flexPin);   // Read flex sensor

 // Example calibration values (may vary)

int gripperAngle = map(flexValue, 300, 700, 0, 180);

gripperAngle = constrain(gripperAngle, 0, 180);

servo_3.write(gripperAngle);

Serial.print("Flex Sensor Value: ");

Serial.print(flexValue);

Serial.print("  Gripper Angle: ");

Serial.println(gripperAngle);

// -------- Forward / Reverse --------

if (x >= 0 && x <= 60) {

int mov1 = map(x, 0, 60, 0, 90);

Serial.print("Forward/Reverse = ");

Serial.println(mov1);

servo_1.write(mov1);

}
```

```
else if (x >= 300 && x <= 360) {

 int mov2 = map(x, 360, 300, 0, 180);

Serial.print("Up/Down = ");

Serial.println(mov2);

servo_2.write(mov2);

}

// -------- Left / Right --------

if (y >= 0 && y <= 60) {

int mov3 = map(y, 0, 60, 90, 180);

Serial.print("Left = ");

Serial.println(mov3);

servo_4.write(mov3);

}

else if (y >= 300 && y <= 360) {

 int mov3 = map(y, 360, 300, 90, 0);

Serial.print("Right = ");

Serial.println(mov3);

servo_4.write(mov3);

}

delay(20);

}
```

5. Future scope

The current implementation of the gesture-based robotic arm demonstrates the feasibility of controlling a robotic manipulator through real-time hand motion sensing and embedded processing. However, several improvements can be incorporated to enhance system performance and functionality. Future work may focus on improving motion accuracy by integrating advanced filtering techniques such as complementary or Kalman filtering to reduce sensor noise and drift from the MPU6050. Additionally, increasing the number of sensors or implementing more precise motion-tracking modules could allow finer gesture interpretation and smoother robotic movements. Expanding the degrees of freedom of the robotic arm and optimizing the control algorithm could further improve the range and flexibility of operations.

Another potential direction for future development involves incorporating wireless communication and intelligent control techniques into the system. By integrating wireless modules and more powerful embedded platforms alongside the Arduino Nano, the robotic arm could be controlled remotely with greater mobility and reduced physical constraints. Furthermore, implementing machine learning or computer vision-based gesture recognition could significantly enhance the adaptability of the system, enabling it to recognize complex gestures and user-specific motion patterns.

Such improvements could extend the applicability of gesture-controlled robotic systems to advanced industrial automation, medical assistance, rehabilitation devices, and remote operation in hazardous environments.

6. Conclusion

This study demonstrated the successful design and implementation of a gesture-controlled robotic arm using the MPU6050 IMU sensor and a microcontroller, achieving accurate real-time motion tracking and intuitive human–machine interaction. The system effectively translates hand gestures into precise robotic movements, highlighting its reliability, responsiveness, and potential for low-cost automation solutions. Overall, the results validate the feasibility of using inertial sensing for seamless control of robotic systems in various applications. This study will benefit society by enabling accessible, efficient, and user-friendly automation technologies, with future work focusing on improving accuracy, expanding gesture recognition, and integrating advanced AI for broader real-world applications.

Compliance with ethical standards

Acknowledgments

We would like to express our sincere gratitude to the Department of Electronics and Communication Engineering at Goel Institute of Technology and Management for providing the necessary facilities and technical support to carry out this project. We are especially thankful to Mr. Nadeem Ahmad, for valuable guidance, constructive suggestions, and continuous encouragement throughout the development of this work. We also acknowledge the support of laboratory staff members for their assistance in hardware testing and component integration.

Disclosure of conflict of interest

No conflict of interest to be disclosed.

References

[1] T. B. Sheridan, “Human–robot interaction: Status and challenges,” IEEE Transactions on Human-Machine Systems, vol. 44, no. 1, pp. 1–10, Feb. 2014.

[2] S. Mitra and T. Acharya, “Gesture recognition: A survey,” IEEE Transactions on Systems, Man, and Cybernetics, vol. 37, no. 3, pp. 311–324, May 2007.

[3] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction, 2nd ed. Cambridge, MA, USA: MIT Press, 2018.

[4] InvenSense, “MPU-6000 and MPU-6050 Product Specification,” 2013.

[5] Arduino, “Arduino Nano Documentation,” Arduino Official Documentation.

[6] J. J. Craig, Introduction to Robotics: Mechanics and Control, 3rd ed. Upper Saddle River, NJ, USA: Pearson, 2005.

[7] B. Siciliano and O. Khatib, Springer Handbook of Robotics. Berlin, Germany: Springer, 2016.

[8] Michels, D. G., et al. (2025). “Real-Time Hand Gesture Control of a Robotic Arm with Programmable Motion Memory”. Proceedings of the International Conference on Robotics and Automation (ICRA).

[9] Reddy, C. I., et al. (2025). Implementation of a Human Hand Gesture Controlled Robotic Arm using IoT Protocols. International Journal of Scientific Development and Research (IJSDR), Vol. 10, Issue 3.

[10] Koparde, S., et al. (2023). Robotic Arm Using Hand Gesture with AI and Mediapipe Landmark Detection. Journal of Emerging Technologies and Innovative Research (JETIR), Vol. 10, Issue 5.