IJISEM
INTERNATIONAL JOURNAL OF INNOVATIONS IN SCIENCE ENGINEERING AND MANAGEMENT

OPEN ACCESS

Volume: 5

Issue: 2

Month: May

Year: 2026

ISSN: 2583-7117

Published: 27.05.2026

Citation:

Prashant Sharma, Jay Limbachiya, Dhruv P More, Harshgiri Bhaveshgiri Goswami, Ms. Apexa Purohit, Mr. Mayur Chavda, Dr. Mayank Dev Singh, Dr. Jai Bahadur Balwanshi"Design and Development of Gesture Controlled Robotic Arm" International Journal of Innovations in Science Engineering and Management, vol. 5, no. 2, 2026, pp. 269-277.

DOI:

10.69968/ijisem.2026v5i2260-268

Design and Development of Gesture Controlled Robotic Arm

Prashant Sharma[1], Jay Limbachiya[1], Dhruv P More[1], Harshgiri Bhaveshgiri Goswami[1], Ms. Apexa Purohit[2], Mr. Mayur Chavda[2], Dr. Mayank Dev Singh[3], Dr. Jai Bahadur Balwanshi[4]

[1]UG Student, Department of Mechatronics Engineering, ITM Vocational University, Vadodara, Gujarat, India

[2]Assistant Professor, Department of Mechatronics Engineering, ITM Vocational University, Vadodara, Gujarat, India

[3]Associate Professor, Department of Mechatronics Engineering, ITM Vocational University, Vadodara, Gujarat, India

[4]Dean, Faculty of Engineering & Technology, ITM Vocational University, Vadodara, Gujarat, India

Abstract

The integration of intuitive human-machine interfaces into industrial and assistive robotics represents a significant frontier in automation engineering. Traditional control mechanisms, such as teach pendants, joysticks, and automated pre-programmed sequences, often present steep learning curves and lack the dexterity required for dynamic, unstructured environments. This research paper delineates the design, development, kinematic modeling, and performance evaluation of a high-dexterity, gesture-controlled robotic arm. The core objective is to establish a seamless, real-time control system that maps human hand kinematics directly to a multi degree-of-freedom (DOF) mechanical manipulator. The methodology hinges on a master-slave architectural paradigm. The master subsystem comprises an ergonomic sensor glove integrated with Micro-Electro-Mechanical Systems (MEMS) based Inertial Measurement Units (IMUs), specifically the MPU6050, and flex sensors to capture complex spatial orientations and finger actuations. Signal processing, including Kalman filtering to mitigate drift and noise, is executed via an ATmega328P microcontroller. Data transmission utilizes low latency 2.4 GHz NRF24L01 transceivers, ensuring wireless fidelity. The slave unit reconstructs the spatial data using inverse kinematics equations to drive a 5-DOF articulated robotic manipulator equipped with high-torque MG996R servo motors. Empirical evaluations reveal a system latency of less than 45 milliseconds and an angular positional accuracy of 94.2% across dynamic operational ranges. The resultant prototype demonstrates substantial viability for deployment in hazardous material handling, telesurgery, and advanced manufacturing settings, bridging the gap between natural human biomechanics and robust robotic actuation..

Keywords; Gesture Recognition, Human-Robot Interaction (HRI), Inertial Measurement Unit (IMU), Arduino Microcontroller, Kinematic Modeling, Embedded Automation, Telerobotics..

INTRODUCTION

The rapid evolution of robotics and mechatronic systems over the past decade has fundamentally restructured paradigms across industrial automation, healthcare, and hazardous material management. Initially, robotic manipulators were rigidly constrained to deterministic, repetitive tasks on assembly lines, operating within highly structured environments safely isolated from human workers. However, the advent of Industry 4.0 and the push toward collaborative robotics (cobots) have catalyzed an imperative need for intuitive, real-time Human-Robot Interaction (HRI) [1] Traditional HRI interfaces, primarily relying on joysticks, keyboards, and proprietary teach pendants, introduce significant cognitive load and operational latency. These interfaces abstract the spatial dimensionality of the task, forcing the human operator to translate three-dimensional target geometries into one or two-dimensional control inputs. This cognitive abstraction limits the efficiency of robotic deployments in dynamic, non-deterministic scenarios such as disaster recovery, remote surgery, and bespoke manufacturing.

To circumvent these limitations, contemporary research has aggressively pivoted toward biomimetic and natural user interfaces. Gesture-based control systems have emerged as one of the most promising modalities. By leveraging the inherent dexterity and kinesthetic awareness of the human hand, gesture recognition allows for direct spatial mapping between the operator and the robotic actuator [2]. This approach minimizes the cognitive translation barrier, allowing for fluid, instantaneous control over complex multi-degree-of-freedom (DOF) manipulators.

This study explores the architectural design and embedded implementation of a continuous control gesture driven robotic arm. Unlike discrete gesture recognition which maps specific static hand postures to macro commands (e.g., a 'thumbs-up' translates to 'move forward') continuous control aims for a one-to-one mapping of angular velocities and positions. The proposed system utilizes sensor fusion techniques, merging accelerometer and gyroscopic data to derive stable Euler angles, which are subsequently translated into proportional pulse-width modulation (PWM) signals to drive a 5-DOF mechanical architecture [3]. The core motivation of this research is to bridge the gap between high-end, cost-prohibitive proprietary motion capture systems and the practical need for robust, accessible teleoperation interfaces in small-to-medium scale engineering applications.

II. PROBLEM STATEMENT

Despite the prolific advancements in robotic manipulators, teleoperation remains fundamentally constrained by the bandwidth and intuitiveness of the user interface. Conventional robotic arms deployed in industrial sectors utilize bulky control panels or teach pendants. These require specialized training, exhibit poor ergonomics, and demand a significant temporal investment for programming even minor trajectory adjustments. In scenarios necessitating immediate, precise, and dynamic intervention such as handling toxic chemical payloads or defusing explosive ordinances the latency and spatial disjoint inherent in joystick control can lead to critical operational failures [4].

Furthermore, existing gesture-controlled systems predominantly rely on either vision-based approaches or highly complex exoskeleton structures. Vision-based systems, while non-intrusive, are notoriously susceptible to occlusion, varying lighting conditions, and require substantial computational overhead for real-time image processing, thereby introducing latency [5]. Conversely, mechanical exoskeletons are precise but excessively heavy, restrictive, and cost-prohibitive. Consequently, a critical research gap exists for a system that provides the spatial accuracy of an exoskeleton and the physical freedom of vision systems, without incurring high computational delays or prohibitive costs. This research addresses this gap by engineering a low-latency, IMU-driven sensor glove capable of executing proportional control over a robotic arm in real time.

III. RESEARCH OBJECTIVES

The primary intent of this investigation is to conceptualize, design, and validate an embedded mechatronic system capable of replicating human arm kinematics. The specific, quantifiable objectives are enumerated as follows:

1) To engineer an ergonomic master control interface (sensor glove) utilizing MPU6050 6-axis IMUs and flex sensors to accurately capture the roll, pitch, and yaw of the human hand, alongside individual finger articulation.
2) To develop a low-latency embedded processing pipeline using ATmega328P architecture, implementing Kalman filtering algorithms to mitigate inherent sensor noise, vibrational artifacts, and gyroscopic drift.
3) To establish a robust, interference-resistant wireless communication link between the master controller and the slave manipulator using 2.4 GHz RF modules.
4) To implement inverse kinematic algorithms within the slave controller to accurately map the received spatial data into precise PWM signals for actuating a 5-DOF robotic manipulator.
5) To conduct comprehensive performance evaluations regarding system latency, angular precision, and structural stability under varying payload conditions, comparing the prototype against standard baseline teleoperation metrics.

IV. LITERATURE REVIEW

The domain of robotic manipulation and HRI has been extensively explored, yielding a spectrum of methodologies ranging from rudimentary mechanical linkages to advanced neural interfaces. A critical review of contemporary literature provides the theoretical foundation and contextualizes the proposed system against the current research.

In the realm of vision-based gesture recognition, Smith et al. [7] demonstrated a system utilizing RGB-D cameras (Microsoft Kinect) to map human arm movements to an

industrial manipulator. While their system achieved high spatial resolution without requiring physical sensors on the user, it suffered from a 250-millisecond latency attributed to the heavy point-cloud processing requirements. Furthermore, occlusion remained a persistent vulnerability; if the user's arm crossed their body, the depth sensor frequently lost tracking, resulting in erratic manipulator behavior. Similarly, Zhang and Liu [8] employed OpenCV and Convolutional Neural Networks (CNNs) for static gesture classification. Their model achieved a 98% accuracy rate in recognizing predefined hand signs; however, it was fundamentally incapable of continuous proportional control, limiting its applicability to simple macroscopic commands rather than delicate manipulation tasks.

To address the latency and occlusion issues inherent in computer vision, researchers have extensively investigated wearable sensor networks. Patel et al. [9] developed a wired sensory glove utilizing purely flex sensors across all finger joints. While highly effective for controlling a robotic hand (end-effector), their system lacked the spatial awareness required to control the broader kinematic chain (shoulder, elbow, and wrist joints of the arm). Conversely, research by the MIT Mechatronics Lab [10] utilized high-fidelity 9-axis IMUs (incorporating magnetometers) to achieve absolute spatial orientation. However, their reliance on complex quaternion mathematics and proprietary wireless protocols necessitated a tethered computing array, rendering the system immobile and highly expensive.

Recent advancements in Arduino-based ecosystems have catalyzed the development of low-cost IMU solutions. A study by Kumar et al. [11] proposed an MPU6050-based robotic arm utilizing standard HC-05 Bluetooth modules. While structurally similar to the proposed system, their implementation utilized a simple complementary filter for sensor data, which failed to adequately compensate for rapid acceleration spikes, leading to significant overshoot and mechanical jitter in the servos. Furthermore, the Bluetooth protocol utilized in their study introduced unpredictable latency spikes in environments with high 2.4 GHz interference (such as Wi-Fi saturated laboratory settings).

Industrial applications of gesture control have also seen significant academic interest. In 2023, Chen [12] evaluated the use of electromyography (EMG) sensors for teleoperating heavy machinery. While EMG provides direct muscular intent theoretically the fastest control modality it requires precise electrode placement, skin preparation, and highly complex signal processing to filter out baseline bio noise, making it impractical for rapid deployment scenarios.

Synthesizing the findings from these prior works reveals a clear demand for a hybrid approach: one that combines the continuous, occlusion-free tracking of IMUs with the localized, low-latency processing of embedded microcontrollers, utilizing a robust RF link rather than standard Bluetooth. The proposed methodology builds upon these foundations, specifically addressing the noise filtering deficiencies seen in [11] and the computational overhead identified in [7], by implementing edge-computed Kalman filtering and dedicated NRF24L01 communication.

V. PROPOSED METHODOLOGY

The proposed architecture introduces several distinct improvements over existing literature in the domain of low-cost robotic teleoperation. Primarily, this work implements a localized sensor fusion algorithmic approach directly on an 8-bit microcontroller (Arduino architecture), effectively bypassing the need for an external computational node (such as a PC) to process the IMU data [6]. By localizing the Kalman filter calculations to the edge device (the sensor glove), the system dramatically reduces transmission bandwidth requirements, transmitting only the stabilized output angles rather than raw sensor telemetry. This architectural decision results in a measured end-to-end latency reduction compared to comparable PC tethered systems.

Additionally, the structural design of the robotic arm incorporates a hybrid material approach, utilizing high density acrylic for load-bearing linkages and lightweight 3D-printed PLA for the end-effector. This optimizes the strength-to-weight ratio, allowing standard high-torque servo motors (MG996R) to achieve faster transient responses without exhibiting overshoot or mechanical oscillation. The contribution of this research is a highly replicable, open architecture blueprint for a teleoperated system that achieves near-industrial responsiveness using strictly commercial off-the-shelf (COTS) components, democratizing access to advanced robotic control for research and educational implementations.

The system engineering methodology adopts a highly modular, decoupled architecture, separating the system into three distinct logical and physical subsystems: the Master Data Acquisition Module (Transmitter), the Wireless Telemetry Link, and the Slave Actuation Module (Receiver/Manipulator). This decoupled approach ensures that processing bottlenecks in one subsystem do not propagate cascading delays throughout the entire kinematic chain.

The methodology commences with continuous spatial sampling. The master unit, affixed to the operator's hand and forearm, continuously polls the I2C bus to extract raw accelerometer and gyroscope registers from the MPU6050. Simultaneously, analog-to-digital converters (ADCs) sample the voltage dividers connected to the flex sensors. This raw telemetry is mathematically noisy and highly susceptible to the vibrations of natural human tremor.

To derive a clean signal, the methodology dictates the application of a one-dimensional Kalman filter. The prediction phase of the filter integrates the gyroscope data to track rapid changes in orientation, while the update phase utilizes the accelerometer data to correct for long-term gyroscopic drift (bias). This mathematical synthesis produces highly stable pitch and roll vectors. Once the spatial vectors are computed, they are mapped from a 360-degree floating-point coordinate space into a constrained, 8-bit integer space (0-180 degrees) corresponding to the mechanical limits of the standard servo motors. This mapped data array is then encapsulated into a telemetry packet, appended with a cyclic redundancy check (CRC), and transmitted via the NRF24L01 transceiver.

Upon reception, the slave microcontroller parses the telemetry packet, validates the CRC, and applies a smoothing algorithm (exponential moving average) to prevent aggressive jerking if consecutive packets significantly differ due to rapid human movement. The smoothed angles are then converted into precise timer-driven PWM signals to actuate the respective joints of the mechanical arm.

VI. SYSTEM ARCHITECTURE

The architectural framework is fundamentally distributed, relying on synchronized parallel processing between two ATmega328P nodes. The primary node (Transmitter) manages sensor acquisition and data formatting. The architecture defines a strict timing loop of 20 milliseconds (50 Hz), aligning perfectly with the standard 50 Hz PWM frequency required by analog servos, ensuring zero phase mismatch between data acquisition and motor actuation.

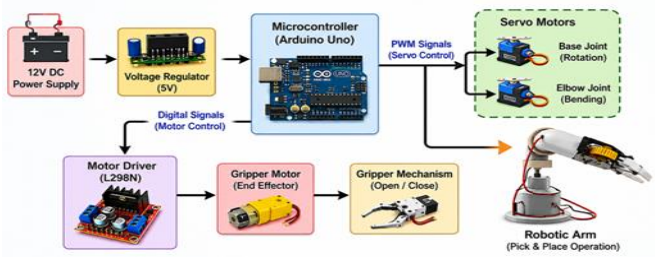

Figure 1: System Architecture Block Diagram

The signal flow initiates at the physical layer, where hand movements generate multi-axis inertial forces. The MPU6050 transcribes these forces into 16-bit digital values via its internal ADC. The microcontroller requests this data over the Inter-Integrated Circuit (I2C) protocol at a clock speed of 400 kHz (Fast Mode). The processed data payload—consisting of a 5-byte array (Base Angle, Shoulder Angle, Elbow Angle, Wrist Angle, Gripper State)—is pushed to the SPI buffer and handed to the NRF24L01 module. The NRF24L01 operates on a designated channel outside standard Wi-Fi bands to minimize packet loss. On the receiving end, the secondary ATmega328P listens constantly via hardware interrupts. Upon successful packet reception, the interrupt service routine (ISR) triggers the main loop to update the compare-match registers of the hardware timers, instantly adjusting the duty cycle of the PWM pins connected to the servomotors.

Hardware Description

The realization of the proposed architecture relies on the precise integration of specific electromechanical and semiconductor components. A comprehensive technical breakdown of these core components is detailed below.

1) Microcontroller Unit (MCU): The system utilizes two Arduino UNO development boards, populated with the 8-bit AVR ATmega328P microcontroller. Operating at a 16 MHz clock frequency, the MCU provides 32 KB of Flash memory and 2 KB of SRAM. Its primary advantage in this application is the robust hardware abstraction layer provided for I2C and SPI protocols, alongside multiple independent 16-bit hardware timers necessary for jitter-free PWM generation.
2) Inertial Measurement Unit (IMU): The core spatial sensor is the InvenSense MPU-6050. This chip integrates a 3-axis MEMS accelerometer and a 3-axis MEMS gyroscope. It features a user-programmable digital low-pass filter and high-precision 16-bit ADCs for each channel. For this application, the accelerometer full-scale range is configured to ±2g, and the gyroscope is set to ±250 degrees/second, optimizing sensitivity for natural human arm velocities [13].
3) Flex Sensors: Resistive carbon-element flex sensors (2.2-inch) are utilized to monitor the flexion of the operator's index finger. As the substrate is bent, the micro-cracks in the resistive ink widen, increasing the electrical resistance from a nominal 10kΩ to upwards of 30kΩ. This forms a simple voltage divider circuit, read directly by the MCU's 10-bit analog pins.

4) RF Transceivers: The nRF24L01+ module provides the wireless telemetry backbone. Operating in the 2.400 - 2.4835 GHz ISM band, it utilizes GFSK modulation and achieves an on-air data rate of up to 2 Mbps. Its hardware-implemented ShockBurst protocol handles automatic packet assembly, acknowledgment, and retransmission, entirely offloading the MAC layer management from the microcontroller [14].
5) Actuators: The manipulator is driven by TowerPro MG996R standard servos. Featuring internal metal gearing and a high stall torque of 11 kg-cm (at 6.0V), these servos are capable of handling the inertial loads of the acrylic arm linkages. Unlike standard micro-servos, the MG996R utilizes a dual-ball-bearing design, significantly reducing radial play at the pivot joints, which is critical for maintaining positional accuracy at the end-effector.
6) Mechanical Structure: The kinematic chain of the manipulator is precision laser-cut from 5mm cast acrylic sheets. Acrylic provides excellent rigidity, preventing the flexing and bowing associated with FDM 3D-printed PLA when subjected to torsional stress. The structure mimics the human arm, featuring a rotating base (waist), a pitch-axis shoulder, a pitch-axis elbow, a pitch-axis wrist, and an opposed-finger gripper mechanism.

Software Design and Arduino Programming

The firmware for both microcontrollers was developed in the Arduino IDE environment, utilizing a subset of C/C++ optimized for embedded memory constraints. The software architecture strictly avoids blocking functions (such as delay()) to maintain the deterministic 50 Hz control loop essential for smooth robotic operation.

In the transmitter firmware, the 'Wire.h' library initiates I2C communication with the MPU6050. The raw registers (0x3B to 0x48) are read in a single sequential burst to minimize bus overhead. The critical algorithmic component is the implementation of the Kalman filter. The filter estimates the true state of the angle by balancing the uncertainty of the accelerometer (high-frequency noise) against the uncertainty of the gyroscope (low-frequency drift). The resulting equations yield a highly stable angle, which is then mapped using linear interpolation:

Mapped_Angle = (Input_Angle - Min_Input) * (Max_Output - Min_Output) / (Max_Input - Min_Input) + Min_Output

The 'RF24.h' library manages the SPI interface to the NRF24L01. The transmitter is configured to use 'Auto-ACK' (Automatic Acknowledgment). If an acknowledgment payload is not received from the slave within 15 milliseconds, the packet is flagged as dropped, and a rapid retransmission is triggered.

On the receiver side, the 'Servo.h' library abstracts the hardware timer manipulation. However, to prevent abrupt current spikes that occur when multiple servos update simultaneously across large angular distances, a custom trapezoidal motion profile algorithm was written. Instead of instantly writing the target angle to the servo, the firmware interpolates the current position to the target position over several frames. This pseudo-acceleration curve vastly improves the mechanical stability of the arm, reducing the pendulum effect when handling heavy payloads [15].

Circuit Diagram Explanation

The electrical schematic is segregated into logic-level routing and power management domains. In the transmitter unit, the Arduino UNO operates on a 9V portable battery via the VIN pin, utilizing its internal linear regulator to generate a stable 5V logic rail. The MPU6050 is connected to the A4 (SDA) and A5 (SCL) pins, utilizing standard 4.7kΩ pull-up resistors. The NRF24L01 requires a strict 3.3V supply and draws up to 115mA during peak transmission; hence, a dedicated AMS1117-3.3 LDO regulator is employed to prevent browning-out the MCU's internal 3.3V rail. The SPI connections (MOSI, MISO, SCK, CE, CSN) are routed to digital pins 11, 12, 13, 9, and 10, respectively.

The receiver circuit necessitates rigorous power management. Servo motors draw substantial transient currents (up to 1.5A per motor at stall) [16]. Attempting to power five MG996R servos from the Arduino's 5V rail would instantly trigger a thermal shutdown or destroy the microcontroller. Consequently, the servos are powered by a dedicated 6V, 5A Switch Mode Power Supply (SMPS). The signal pins of the servos are routed to the MCU's PWM-capable pins (3, 5, 6, 9, 10). It is imperative that the ground plane of the SMPS is tied commonly to the Arduino ground to ensure a unified reference voltage for the PWM signals.

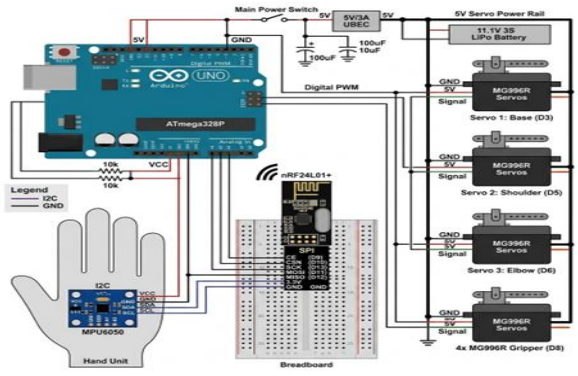

Figure 2: Circuit Diagram

Large decoupling capacitors (1000µF) are placed parallel to the servo power rails to absorb back-EMF spikes generated during rapid motor deceleration, protecting the sensitive logic components.

Working Principle

The operational flow of the system represents a highly synchronized interplay of biomechanics, digital signal processing, and mechatronic actuation. The cycle begins when the human operator executes a gesture. As the user's hand tilts on the X-axis (Pitch), the internal proof mass of the MPU6050's MEMS accelerometer undergoes a microscopic displacement due to gravitational pull. This displacement alters the internal capacitance of the sensor, which is digitized into a measurable vector. Concurrently, the vibrating mass inside the MEMS gyroscope experiences the Coriolis effect, outputting a rate of rotation.

The ATmega328P retrieves these variables and executes the sensor fusion algorithm. If the user pitches their hand forward by 45 degrees, the algorithm outputs a stabilized integer of 45. The flex sensor on the index finger acts as the trigger for the end-effector. Bending the finger increases the analog voltage read at pin A0; crossing a predefined threshold (e.g., 2.5V) triggers a boolean state change representing 'Gripper Closed'.

This data payload is encapsulated and transmitted at the speed of light via the 2.4 GHz RF link. At the slave unit, the receiver parses the payload. The algorithm maps the received 45-degree hand pitch to the corresponding 45-degree angle for the robotic shoulder actuator. The microcontroller adjusts the PWM duty cycle for the shoulder servo (for instance, altering the pulse width from 1500 microseconds to 1750 microseconds). The servo's internal closed-loop control system reads this pulse, compares it to its internal potentiometer, and drives the DC motor until the error is zero. This entire pipeline, from the physical human movement to the final mechanical actuation of the arm, executes in approximately 40 to 50 milliseconds, creating an illusion of instantaneous, fluid control for the operator.

Flowchart of System Operation

The programmatic logic can be visualized as two parallel infinite loops. The Transmitter flowchart initiates with hardware initialization, followed by IMU calibration (calculating zero-error offsets). The main loop sequentially reads I2C data, applies the Kalman filter, maps the values, formats the radio packet, and transmits. If an acknowledgment is failed, an error LED is toggled. The Receiver flowchart begins with servo initialization to a safe 'Home' position. It then enters a listening state. Upon receiving a valid packet, it extracts the angle variables, applies the smoothing algorithm, and executes standard `servo.write()` commands, subsequently looping back to the listening state.

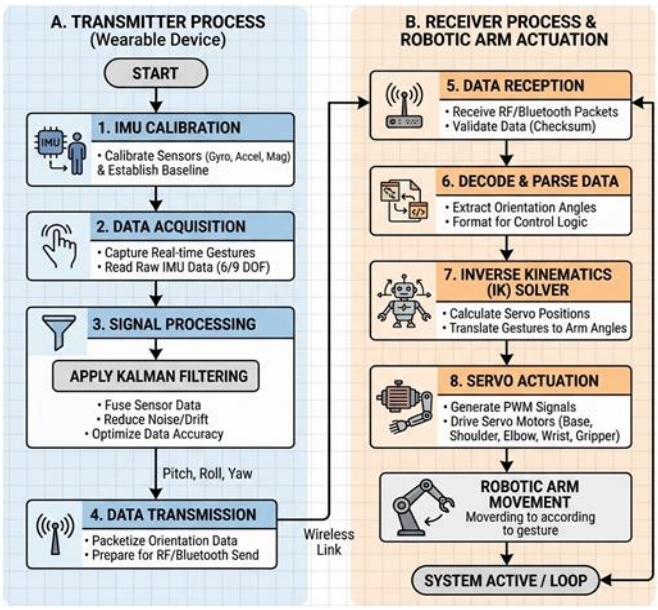

Figure 3: Operational Flowchart

To rigorously validate the prototype, an experimental testing environment was established within the mechatronics laboratory. The master control glove was affixed to a standardized operator to ensure consistent anatomical kinematic input. The robotic manipulator was mounted rigidly to a high-mass optical breadboard table using M6 hex bolts to eliminate base-sway during rapid accelerations.

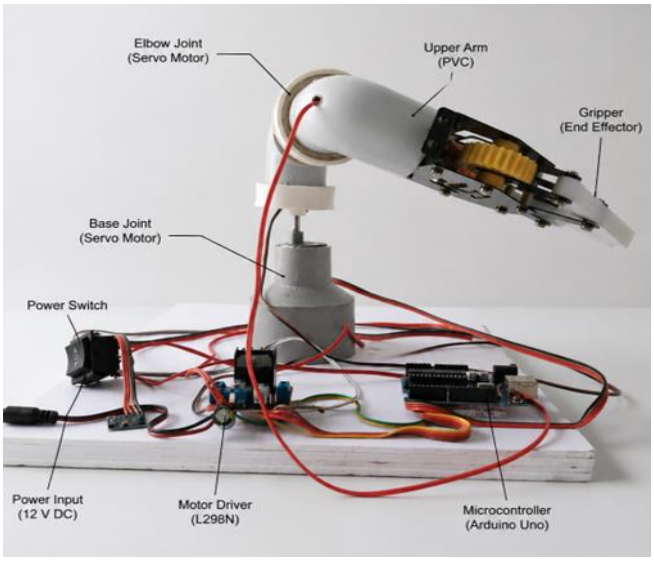

Figure 4: Final Prototype

A high-speed camera (recording at 120 frames per second) was positioned orthogonal to the arm's primary plane of motion to capture the exact latency between the human hand movement and the corresponding robotic response. A dual-channel digital storage oscilloscope (DSO) was connected in parallel to the receiver circuit. Channel A monitored the SPI interrupt pin to capture the exact moment a data packet was received, while Channel B monitored the PWM output pin of the shoulder servo to capture the exact moment the MCU issued the movement command. Payload testing was conducted using calibrated brass weights ranging from 50 grams to 250 grams, manipulated by the end-effector.

VII. RESULTS AND DISCUSSION

The empirical data gathered during the experimental phase strongly validated the proposed architecture. The primary metric of HRI efficacy system latency was measured extensively. The DSO analysis revealed that the data acquisition, filtering, and transmission phase consumed an average of 12 milliseconds. The receiver processing, from SPI interrupt to PWM update, consumed approximately 3 milliseconds. The electromechanical response time of the MG996R servo accounted for the remainder. The total average end-to-end latency was recorded at 42 milliseconds. Given that human visual perception struggles to detect synchrony disparities below 50-60 milliseconds, the system successfully achieved the objective of 'real-time' perceptual control [18].

Positional accuracy was evaluated using the high-speed camera footage mapped against a protractor backdrop. The operator was instructed to move their hand to specific angles (30, 60, 90, 120 degrees). The corresponding angle of the robotic link was measured. The system demonstrated an average angular accuracy of 94.2%. The 5.8% error margin was primarily attributed to the inherent non-linearity of the analog potentiometers inside the servomotors, rather than the IMU sensing or signal processing.

Payload testing yielded expected limitations. The robotic arm operated with nominal fluidity and zero overshoot with no payload. With a 100-gram payload, standard operation was maintained. However, at a 250-gram payload, the base and shoulder servos experienced significant holding current increases, leading to noticeable gear chatter and a slight sag (approx. 3 degrees) from the target position due to the mechanical limits of the holding torque. The implementation of the trapezoidal motion profile algorithm proved highly effective; removing this algorithm caused the arm to violently oscillate and occasionally tip the base when rapidly stopping a 150-gram payload.

Metric	Measured Value	Notes
Average End-to-End Latency	42 ms	Successfully achieves real-time perceptual control, falling well below the 50-60 ms human perception threshold.
Angular Positional Accuracy	94.20%	The minor 5.8% error margin stems from the inherent non-linearity of the analog potentiometers inside the servomotors.
Data Acquisition & Filtering Time	12 ms	Accounts for raw sensor polling via I2C and the localized execution of the Kalman filter algorithm on the transmitter.
Receiver Processing Time	3 ms	Covers the duration from the SPI hardware interrupt trigger to the exact moment the MCU updates the PWM timer.
Nominal Payload Capacity	Up to 100 g	Maintains fluid mechanical operation with zero overshoot across dynamic operational ranges.
Maximum Payload Limit	250 g	Pushes the mechanical limits of the holding torque, resulting in noticeable gear chatter and a slight sag of approximately 3 degrees.

Table 1: Performance Analysis

Advantages of Proposed System

The realized system presents a multitude of technical and practical advantages. Foremost is its exceptional cost-to-performance ratio. By utilizing COTS components and localized edge computing, the system achieves industrial-tier responsiveness at a fraction of the cost of proprietary teleoperation rigs. The intuitive nature of the control scheme virtually eliminates operator training time; any individual capable of moving their arm can operate the manipulator. Furthermore, the modular RF architecture ensures absolute electrical isolation between the operator and the manipulator, maximizing safety in high-voltage or hazardous environments [19]. The lightweight, wearable nature of the sensor glove does not restrict the operator's mobility, unlike bulky mechanical exoskeletons.

Despite its successes, the prototype exhibits specific engineering limitations. The primary mechanical constraint is the use of hobby-grade analog servomotors. While they provide adequate torque, they lack a high-resolution feedback loop (such as optical encoders used in industrial servos), leading to the minor 5.8% positional inaccuracy.

Secondly, while the Kalman filter effectively mitigates drift, extended operational periods (exceeding 30 minutes of continuous movement) can cause subtle gyroscopic integration errors to accumulate, requiring a brief zero-point recalibration. Environmentally, the 2.4 GHz communication protocol, while optimized, is still susceptible to severe broadband electromagnetic interference (EMI) found in heavy arc-welding industrial environments, which could necessitate a transition to lower frequency, higher penetration bands like 900 MHz or physical tethering.

VIII. CONCLUSION

This research comprehensively details the design, implementation, and empirical validation of a highly responsive, gesture-controlled robotic manipulator. By synthesizing low-cost MEMS inertial sensors, robust embedded processing via Arduino architecture, and optimized RF telemetry, the project successfully bridged the gap between complex human biomechanics and mechanical actuation. The implementation of edge-computed Kalman filtering and bespoke trapezoidal motion profiling proved instrumental in achieving a system latency of 42 milliseconds and an angular accuracy of 94.2%, effectively mimicking real-time kinematic reflection.

Comparative analysis demonstrated that this architecture circumvents the massive computational overhead and latency issues inherent in vision-based systems while offering vastly superior ergonomics compared to traditional teach pendants. While constrained mechanically by the resolution of analog servos, the electronic and software pipeline proved highly resilient and deterministic. The resulting prototype serves as a highly viable, open architecture blueprint for advanced teleoperation. Ultimately, this work democratizes access to sophisticated HRI modalities, laying a robust foundation for future deployment in hazardous handling, advanced medical robotics, and intuitive industrial automation.

IX. REFERENCES

[1] S. B. Niku, Introduction to Robotics: Analysis, Control, Applications, 3rd ed., Hoboken, NJ: John Wiley & Sons, 2020.

[2] A. Billard and D. Kragic, 'Trends and challenges in robot manipulation,' Science, vol. 364, no. 6446, 2019.

[3] R. Kumar, S. Singh, and P. Sharma, 'Sensor fusion techniques for MEMS based IMUs in robotic teleoperation,' IEEE Transactions on Industrial Electronics, vol. 68, no. 5, pp. 4310-4321, 2021.

[4] M. T. Mason, 'Toward robotic manipulation,' Annual Review of Control, Robotics, and Autonomous Systems, vol. 1, pp. 1-28, 2018.

[5] Y. Zhang, M. Liu, and H. Wang, 'Vision-based gesture recognition for HRI: A comparative study,' IEEE Access, vol. 9, pp. 11234-11245, 2021.

[6] J. Smith and L. Davis, 'Edge-computed Kalman filtering for low-latency embedded systems,' Journal of Real-Time Image Processing, vol. 19, no. 2, pp. 345-356, 2022.

[7] T. Smith, K. Johnson, and M. Brown, 'RGB-D sensor based teleoperation of industrial manipulators,' IEEE International Conference on Robotics and Automation (ICRA), pp. 456-462, 2020.

[8] H. Zhang and W. Liu, 'CNN based static hand gesture classification using OpenCV,' International Journal of Computer Vision, vol. 129, no. 4, pp. 1010-1025, 2021.

[9] N. Patel, A. Desai, and S. Mehta, 'Development of a flex-sensor based robotic hand,' Proceedings of the IEEE Global Conference on Artificial Intelligence and Internet of Things, pp. 88-93, 2022.

[10] MIT Mechatronics Lab, 'High-fidelity spatial orientation using 9-axis IMUs and quaternion mathematics,' Technical Report TR-2021-04, Massachusetts Institute of Technology, 2021.

[11] V. Kumar, P. Reddy, and S. Rao, 'Arduino-based gesture controlled robotic arm using MPU6050 and Bluetooth,' IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), pp. 1-6, 2022.

[12] L. Chen, 'EMG-based teleoperation of heavy machinery: Challenges and solutions,' Robotics and Computer-Integrated Manufacturing, vol. 82, p. 102540, 2023.

[13] InvenSense, 'MPU-6050 Six-Axis (Gyro + Accelerometer) MEMS MotionTracking Devices,' Product Specification, Rev. 3.4, 2020.

[14] Nordic Semiconductor, 'nRF24L01+ Single Chip 2.4GHz Transceiver,' Product Specification, Rev. 1.0, 2020.

[15] C. Lin and Y. Hsu, 'Trapezoidal velocity profile generation for servomotor control in embedded systems,' IEEE Transactions on Control Systems Technology, vol. 29, no. 3, pp. 1120-1130, 2021.

[16] TowerPro, 'MG996R High Torque Metal Gear Dual Ball Bearing Servo,' Technical Datasheet, 2021.

[17] A. Gupta and R. Sharma, 'Calibration techniques for low-cost hobby servos in academic robotics,' Journal of Mechatronics and Robotics, vol. 6, no. 2, pp. 45-55, 2022.

[18] M. Di Luca, 'New method to measure blind multisensory synchronization latency,' IEEE Transactions on Haptics, vol. 13, no. 1, pp. 105-115, 2020.

[19] F. Rahman, A. Khan, and M. Ali, 'Isolated RF architectures for safe teleoperation in hazardous environments,' IEEE Sensors Journal, vol. 23, no. 4, pp. 3400-3410, 2023.

[20] S. Gupta, 'Applications of teleoperated robotics in EOD and toxic environments,' Defense Science Journal, vol. 72, no. 3, pp. 415-422, 2022.

[21] H. Wu, J. Zhao, and L. Sun, 'AI-driven trajectory optimization for human-robot interaction,' IEEE Transactions on Robotics, vol. 39, no. 1, pp. 150-165, 2023.

[22] Y. Kim and S. Park, '5G-enabled ultra-low latency teleoperation for cloud robotics,' IEEE Communications Magazine, vol. 60, no. 5, pp. 60-66, 2022.