International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

Design and Implementation of a Low-Cost Wearable-Sensor Interface for Wireless Control of a Six-Axis Robotic Arm

Agraram Ganesh Goud

Submitted: 01/05/2024 Revised: 08/06/2024 Accepted: 18/06/2024

Abstract: Natural and accessible control interfaces are important for robotic manipulation in education, remote handling, assistive systems and small-scale automation. This paper presents the design and implementation of a low-cost wearable-sensor interface for wireless control of a six-axis robotic arm with a gripper. The transmitter is a glove instrumented with three flex sensors and two MPU6050 inertial sensors. An Arduino Nano performs sensor acquisition, start-up calibration and threshold-based gesture mapping, while an HC-05 Bluetooth link transmits compact character commands to an Arduino Uno on the robotic arm. The receiver coordinates a PCA9685 servo driver, six MG996R-class servo motors and an A4988-driven NEMA 17 stepper motor. The mechanical structure was manufactured from open-source-derived, 3D-printed components, with the stepper motor selected for base rotation because the assembled arm exceeded the practical torque available from a standard servo at that joint. The completed prototype reproduced mapped finger, wrist and arm gestures as joint motions, including base rotation, shoulder and elbow positioning, wrist orientation and gripper opening and closing. Implementation testing also identified reversed motion commands caused by sensor orientation and command mapping; these were corrected by exchanging the corresponding command conditions in firmware. The work demonstrates a self-contained microcontroller-based architecture that does not require an external camera, workstation or cloud service. The contribution is an experimentally implemented engineering platform and a transparent account of its calibration, command mapping, hybrid actuation and integration challenges. Quantitative recognition accuracy and latency measurements were not available in the submitted project records and are therefore identified as priorities for future validation.

Keywords— *human–robot interaction; gesture control; robotic arm; wearable sensors; flex sensor; MPU6050; Arduino; Bluetooth; teleoperation; 3D printing.*

1. Introduction

Robotic manipulators are increasingly used in manufacturing, inspection, rehabilitation, education and remote handling. Their effectiveness depends not only on mechanical accuracy but also on the way human operators communicate intent to the machine. Conventional interfaces such as teach pendants, joysticks and keyboards are reliable, but they can require training and can interrupt the direct relationship between a user's movement and the robot's response. Gesture-based interaction offers a more natural alternative because motion of the hand or upper limb can be converted into robot commands without requiring the operator to continuously manipulate a separate control device [1], [2].

Gesture interfaces are commonly divided into vision-based and wearable-sensor approaches. Vision systems can estimate hand pose without physical contact, but performance can depend on lighting, occlusion, camera position, reflective surfaces and computational resources. Wearable approaches place inertial, flex, strain, pressure or electromyographic sensors on the user. They require the user to wear a device, yet they are generally independent of the camera field of view and can provide direct measurements of finger bending and limb

Independent Researcher, Watford, United Kingdom

Work carried out during the MSc in Advanced Robotics, School of Engineering and Materials Science, Queen Mary University of London, United Kingdom

Correspondence: ganeshgoud874@gmail.com

orientation [3]–[5]. For portable and low-cost robotics, a sensor glove combined with an embedded microcontroller is therefore an attractive architecture.

Prior research has demonstrated sophisticated glove-based recognition, including embedded inertial classification, continuous flex-sensor gesture recognition, machine-learning gloves and commercial haptic teleoperation systems [5]–[9]. However, many reported systems concentrate on gesture classification rather than physical robot control, or they depend on high-cost commercial robots, optical tracking systems or external computers. At the opposite end of the spectrum, many educational robotic-arm demonstrations provide limited technical documentation, do not discuss calibration and troubleshooting, or report functions that are not supported by reproducible implementation details. A practical research contribution can therefore arise from a fully integrated, self-contained prototype that documents the complete path from wearable sensing to mechanical actuation.

The project reported here developed a wearable glove and a six-axis, 3D-printed robotic arm using widely available components. Three flex sensors capture finger bending, while two MPU6050 devices capture hand and wrist orientation. The glove-side Arduino Nano calibrates sensor thresholds and transmits character commands through Bluetooth. The robot-side Arduino Uno decodes those commands and controls a combination of servo and stepper actuators. A NEMA 17 stepper motor is used at the base because the assembled mechanical structure created a torque requirement that could not be met reliably by the initially considered servo arrangement. This hybrid actuation decision is a central practical feature of the implementation.

The project records state that the prototype was built, integrated and functionally tested, and that six-axis motion, gesture interaction and gripper control were achieved. The records also document an important integration problem: several observed motions were reversed relative to their intended directions. The problem was traced to sensor placement and command logic and was addressed by swapping the relevant conditions and transmitted characters. Reporting such implementation evidence is valuable because it shows how a seemingly simple gesture-to-motion mapping is affected by coordinate conventions, sensor orientation and servo installation.

The main contributions of this paper are:

- A complete low-cost architecture linking a wearable glove to a six-axis robotic arm without an external camera, workstation or network service.
- A hybrid actuation design that combines a NEMA 17 stepper motor for base rotation with multiple servo motors for articulated joints and gripper motion.
- A reproducible calibration and deterministic gesture-mapping method implemented on resource-constrained Arduino controllers.
- A documented account of prototype integration, functional validation, command-direction errors, corrective firmware changes and current limitations.

The remainder of the paper is organized as follows. Section 2 reviews glove-based gesture sensing and robotic teleoperation. Section 3 describes the system requirements and architecture. Section 4 explains calibration and control mapping. Section 5 presents the mechanical, electronic and software implementation. Section 6 reports functional validation. Sections 7 and 8 discuss the contribution, limitations and future work, followed by the conclusion in Section 9.

2. Related Work

2.1 Wearable sensing for gesture recognition

Glove-based systems have a long history in human–computer and human–robot interaction. Dipietro, Sabatini and Dario surveyed glove technologies and highlighted the trade-off between sensing capability, wearability, calibration and application-specific requirements [3]. Flex sensors remain popular because their resistance changes with bending and they can be mounted directly along a finger. Inertial measurement units complement flex sensors by measuring orientation and motion of the hand or wrist. Together, these modalities allow a compact glove to distinguish finger state from gross limb orientation.

Mummadi et al. developed an embedded IMU glove that recognized 22 hand signs and performed classification on the wearable device itself [5]. Their work demonstrated the feasibility of real-time embedded recognition, but the application centred on gesture classification rather than direct control of an articulated robot. Chuang et al. investigated continuous finger gestures using wireless flex-sensor gloves [6]. Their results show that flex sensors can support temporal gesture recognition, although the associated training and recognition pipeline is more complex than the deterministic threshold method used in the present prototype.

Recent work has expanded the number and type of sensors incorporated into data gloves. Filipowska et al. combined flex, force and inertial sensing in a low-cost machine-learning glove for dynamic gesture recognition [10]. Saeed et al. reviewed sensor-glove systems and identified recurring challenges such as sensor selection, calibration, dataset design, user dependence and the lack of standardized evaluation [11]. These findings are relevant to the present work: a threshold-based controller reduces computational complexity, but it requires careful calibration and its performance may vary across users or glove placement.

2.2 Gesture control and robotic teleoperation

Gesture recognition becomes a robotic control interface when recognized movements are mapped to joint, velocity, pose or task commands. Salman et al. presented a wireless, 3D-printed robotic hand system controlled through a glove with flexible force sensors [7]. That work is closely related in its use of low-cost fabrication and wearable sensing, but it primarily reproduced hand motion in a robotic hand or forearm. The present project extends the concept to an articulated arm that includes base, shoulder, elbow, wrist and gripper functions.

Industrial and research systems often use more powerful sensing and control infrastructure. Roda-Sanchez et al. compared RGB-D and wearable IMU gesture recognition for human–robot interaction in remanufacturing and showed that wearable sensing can avoid several environmental dependencies of camera-based interfaces [8]. Vysocký et al. used hand gestures for robot-path definition in a collaborative workspace [12]. Their vision-based approach supports direct industrial path programming, but it requires camera-based hand tracking and a more substantial processing platform.

Lu et al. used commercial data gloves and a fuzzy logic controller to teleoperate a UR3e robot and gripper [9]. Yu et al. combined a bending-sensor glove with OptiTrack tracking for agricultural robotic-arm control [13]. These systems provide advanced control and application-specific validation, but their commercial gloves, industrial robot platforms or optical tracking infrastructure are less suitable for a minimal educational or exploratory prototype. The system in this paper occupies a different position: it prioritizes accessibility, portability and direct microcontroller implementation over sophisticated classification, haptic feedback or high-precision external tracking.

2.3 Positioning of the present work

The present design does not claim a new gesture-recognition algorithm. Its contribution is the integration of sensing, wireless communication, embedded control, hybrid actuation and 3D-printed mechanics into a self-contained working platform. The design deliberately uses threshold logic rather than machine learning because the target was transparent, easily modified control on Arduino-class hardware. Table 1 positions the system against representative studies. The comparison also makes clear the principal limitation: unlike several referenced systems, the project records do not contain a controlled dataset or quantitative recognition metrics.

Table 1. Comparison with representative gesture-interface systems

Study	Sensing	Robot/output	Processing	Strength	Limitation relative to this work
Mummadi et al. [5]	IMUs in glove	Gesture classes	Embedded classifier	Large embedded gesture set	No articulated robot control
Chuang et al. [6]	Flex sensors	Finger gestures	Training/recognition	Continuous gesture	Not an arm

Study	Sensing	Robot/output	Processing	Strength	Limitation relative to this work
			pipeline	recognition	teleoperation platform
Salman et al. [7]	Flex-force glove	3D-printed robotic hand/forearm	Wireless embedded control	Low-cost physical prototype	More limited arm articulation
Roda-Sanchez et al. [8]	IMU wristband / RGB-D	Industrial HRI commands	Edge processing	Quantitative comparison	Higher system complexity
Lu et al. [9]	Commercial data glove	UR3e and gripper	PC + fuzzy control	Advanced teleoperation	High-cost hardware
Vysocký et al. [12]	Vision-based hand tracking	Industrial robot path	Camera/computer	Direct path definition	Camera and workspace dependence
Yu et al. [13]	Bending sensors + OptiTrack	Agricultural arm	Multi-program server	Application-specific control	External optical tracking
Present work	3 flex + 2 MPU6050	Six-axis 3D-printed arm + gripper	Arduino Nano/Uno	Self-contained and low-cost	No quantitative metrics

3. System Requirements and Architecture

3.1 Design objectives

The system was designed around seven practical objectives: (1) construct an operational robotic-arm prototype controlled by human hand gestures; (2) detect finger bending and wrist/arm orientation using wearable sensors; (3) provide wireless communication between glove and arm; (4) coordinate servo and stepper actuators; (5) support approximately six axes of motion plus gripper operation; (6) keep the glove and arm portable; and (7) provide a platform that could later be expanded with additional sensors or higher-level control. A payload range of 600 g to 1 kg was stated as a design target in the project plan, but the supplied records do not contain a controlled payload test and this value is therefore not reported as a measured result.

3.2 Overall architecture

The implementation is divided into a transmitter and receiver. The transmitter is the wearable glove. Its three flex sensors measure finger bending and its two MPU6050 units measure inertial orientation. An Arduino Nano reads the sensors, performs an initial calibration, evaluates threshold conditions and converts each recognized state into a single-character command. An HC-05 module transmits the command over a serial Bluetooth link configured at 4800 baud.

The receiver is mounted on the robotic arm. A second HC-05 module passes received characters to an Arduino Uno. The Arduino selects the corresponding movement function and applies software limits before commanding either the PCA9685 servo driver or the A4988 stepper driver. The PCA9685 provides multi-channel pulse-width modulation for the servo motors. The NEMA 17 stepper motor rotates the base, while servo motors actuate the remaining arm joints and gripper. Figure 1 summarizes the data and power-control path.

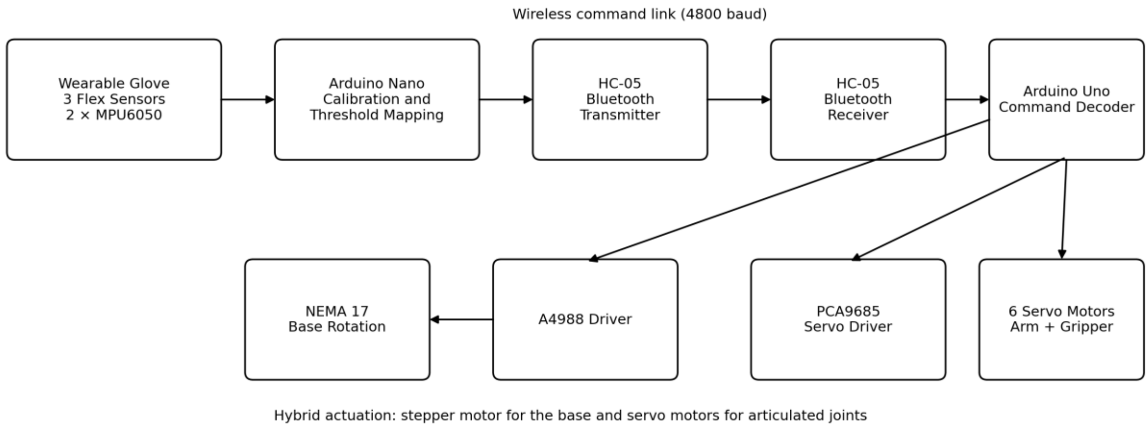

Figure 1. System architecture from wearable sensing to hybrid robotic-arm actuation.

3.3 Mechanical system

The mechanical arm was assembled from 3D-printed parts based on an existing open-source design. The supplied archive contains separate STL files for the base servo mount, bearing holder, centre arm, upper and lower arm members, bushings and bearing guides. Using separate printable components allowed replacement and iterative assembly without machining a monolithic structure. Bearings and bolted joints were used to support rotation and reduce friction at the articulated sections.

The project encountered a practical base-joint issue during assembly: the completed arm was too heavy for reliable base rotation using a conventional servo. The design was therefore adapted to use a NEMA 17 stepper motor with an A4988 driver. This change separated the higher-torque base requirement from the angular positioning of the other joints, which continued to use servo motors. Figure 2 presents the CAD assembly render provided with the mechanical design archive, and Figure 3 shows the constructed gripper and upper arm.

Figure 2. CAD render of the open-source-derived articulated arm used for the prototype.

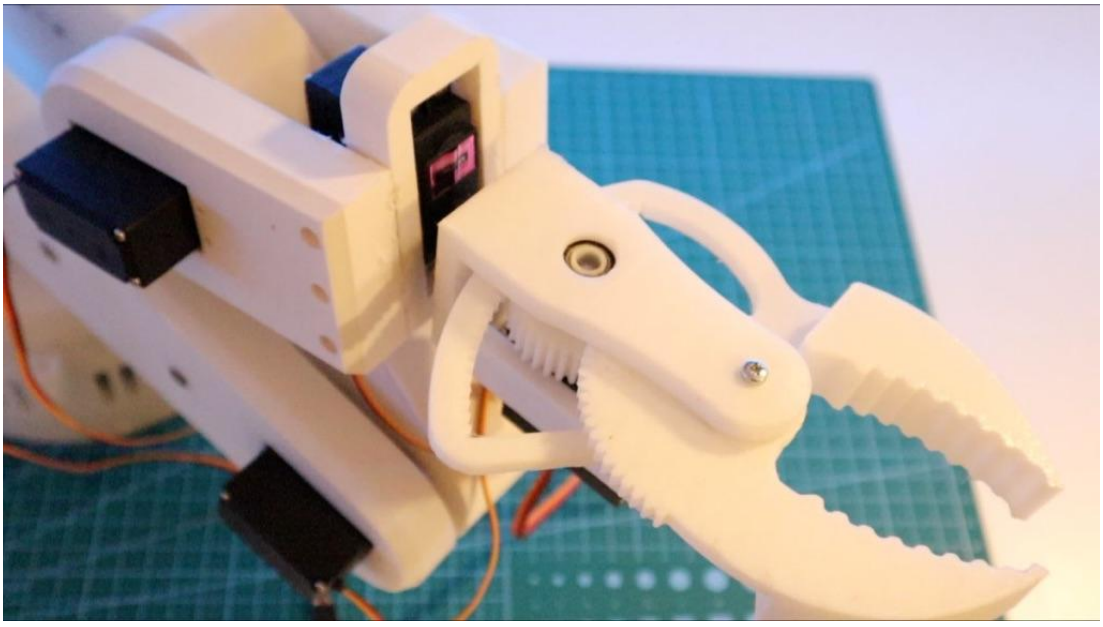

Figure 3. Constructed prototype showing the 3D-printed gripper, gearing, bearings and servo installation.

3.4 Hardware selection

The component selection reflects the project's low-cost and modular goals. Arduino Nano and Uno boards provide sufficient analog inputs, I2C communication and serial interfaces for the required sensor and motor-control tasks. The MPU6050 combines accelerometer and gyroscope sensing in a widely supported package. The PCA9685 reduces the timing and pin burden associated with controlling several servos, while separate motor power supplies prevent the Arduino regulator from supplying the high transient current demanded by the actuators. Table 2 lists the principal hardware and its role.

Table 2. Main prototype hardware

Subsystem	Component	Quantity	Function
Wearable glove	Flex sensor	3	Finger-bending measurement
Wearable glove	MPU6050 IMU	2	Hand, wrist and arm orientation
Wearable glove	Arduino Nano	1	Sensor acquisition, calibration and command encoding
Wireless link	HC-05 Bluetooth module	2	Glove-to-arm serial communication
Robot controller	Arduino Uno	1	Command decoding and actuator coordination
Servo control	PCA9685	1	Multi-channel PWM generation
Arm joints	MG996R-class servo motors	6	Joint and gripper actuation
Base joint	NEMA 17 stepper motor	1	Higher-torque base rotation
Stepper control	A4988 driver	1	Direction and step control
Power	5 V, 2200 mAh battery	1	Servo and low-voltage supply
Power	11.1 V, 2200 mAh LiPo	1	Stepper-motor supply
Glove power	9 V battery	1	Portable glove supply

4. Wearable Gesture Interface and Control Method

4.1 Sensor acquisition and calibration

Each flex sensor forms part of a resistive measurement circuit connected to an analog input on the Arduino Nano. The controller reads the thumb, index-finger and pinkie channels. Two MPU6050 units are connected through I2C at addresses 0x68 and 0x69, allowing the firmware to distinguish the two inertial devices. The documented code sets the command response interval to 1000 ms. This conservative delay simplifies observation during prototype testing, although it also limits interaction speed and should be reduced in a future latency study.

Calibration occurs when the glove controller starts or is reset. A baseline reading is captured for each flex sensor and multiplied by empirical factors to define upper and lower limits. The supplied code uses the factors shown in Table 3. This method adapts the thresholds to the resting sensor values at the beginning of a session and is simpler than storing absolute thresholds. Its limitation is that the calibration represents a single moment and may not compensate for long-term drift, glove movement or different users.

Table 3. Flex-sensor calibration factors documented in the glove firmware

Sensor channel	Baseline symbol	Upper threshold	Lower threshold
Thumb	x_thumb	1.15 x x_thumb	0.90 x x_thumb
Index finger	x_finger	1.03 x x_finger	0.80 x x_finger
Pinkie	x_pinkie	1.06 x x_pinkie	0.80 x x_pinkie

For a generic sensor i with baseline b_i, the firmware defines an upper threshold u_i = k_i^+ b_i and a lower threshold l_i = k_i^- b_i. A gesture state is generated when the current reading crosses one of these limits. This creates hysteresis-like separation between flexed and relaxed states, although it is not a formal adaptive filter. Inertial states are generated from predefined orientation ranges on the axes of the MPU6050 sensors.

4.2 Gesture-to-joint mapping

The control strategy is deterministic: a recognized finger or orientation state is associated with a joint direction. The glove transmits a character representing the state, and the arm controller calls a dedicated movement function. Uppercase and lowercase forms are used for opposite states or directions in several command pairs. The supplied project documents contain minor inconsistencies in the literal character labels used in different sections; for publication, the final Arduino source should be checked against the released command table. The stable functional mapping, which is the technically important element, is summarized in Table 4.

Table 4. Functional mapping between wearable inputs and robot actions

Wearable input	Detected change	Robot function	Actuator
Thumb flex sensor	Flex / relax	Base-direction command	NEMA 17 stepper
Pinkie flex sensor	Flex / relax	Shoulder closer / farther	Paired shoulder servos
Index flex sensor	Flex / relax	Gripper close / open	Gripper servo
MPU6050 No. 1	Wrist tilt	Wrist up / down	Wrist servo
MPU6050 No. 1	Wrist rotation	Clockwise / counter-clockwise wrist motion	Rotation servo
MPU6050 No. 2	Arm orientation	Elbow bend / extend	Elbow servo
MPU6050 No. 2	Lateral orientation	Base or lateral direction command	Stepper / assigned joint

4.3 Receiver-side motion control

The Arduino Uno continuously checks the serial input. When a command is available, it reads the character and selects a movement function. Servo positions are stored as integer variables with defined minimum and maximum values. A function changes the position by a fixed increment only if the proposed value remains inside the allowed range. The position is then passed to the PCA9685 library, which generates the corresponding servo command. The documented maximum servo angle is 180 degrees, although the safe mechanical range of each assembled joint may be narrower and should take priority over the nominal motor limit.

The stepper motor is controlled through direction and step signals to the A4988 driver. Separate baseRotateLeft() and baseRotateRight() functions select the direction and generate pulses. This structure keeps the command decoder readable and makes it possible to alter individual joint speeds or limits without rewriting the sensing logic. Figure 4 presents the control flow.

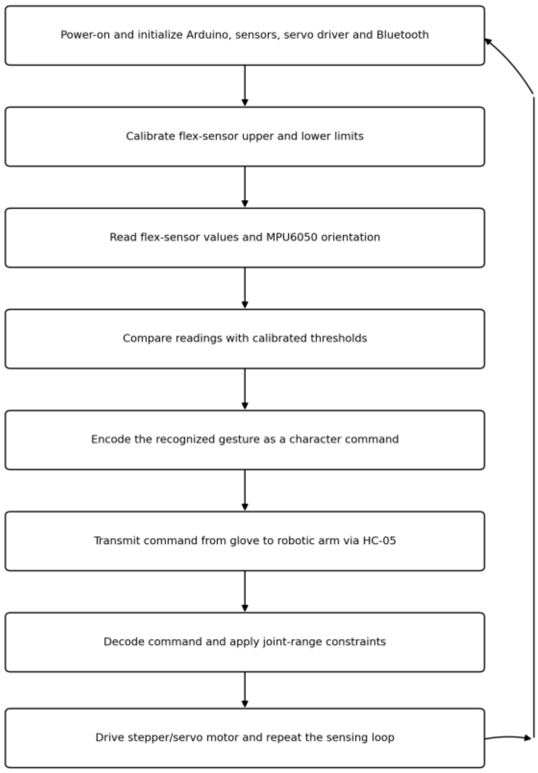

Figure 4. Glove calibration, command generation and receiver-side actuation sequence.

5. Prototype Implementation

5.1 Mechanical assembly

The arm components were fabricated by 3D printing and joined with M3 and M4 fasteners, bearings, servo horns and custom mounting pieces. The gripper includes geared fingers and a servo-driven linkage. The arm archive contains dedicated mounts and mirrored left/right members, indicating that alignment of paired components is essential. During assembly, each servo was positioned near its intended neutral angle before the horn was fixed. This reduces the risk of a joint immediately driving into a mechanical stop when the controller starts.

An external 5 V supply was used for the servo motors because the Arduino board cannot safely provide the combined current required by multiple actuators. Grounds must be common between the logic and actuator supplies so that the PWM and direction signals have a shared reference. The NEMA 17 stepper motor and A4988 driver were powered from the higher-voltage LiPo source described in the project record.

5.2 Glove assembly

The wearable unit was built on a work glove. Flex sensors were aligned with selected fingers, while the inertial sensors were positioned to detect hand and wrist orientation. Resistors provide the flex-sensor voltage-divider circuits and capacitors stabilize the local supply. A braided sleeve organizes and protects the wiring. The Arduino Nano and Bluetooth module are mounted so that the user can move the hand without excessive cable loading. Sensor placement is important: changing the orientation of an MPU6050 changes the sign and axis interpretation of the measured motion, while shifting a flex sensor changes its resting resistance and bending range.

5.3 Software implementation

The project used the Arduino integrated development environment. The arm program includes the HCPCA9685 library and defines the servo-driver I2C address as 0x40. Initial parking positions, increments and minimum/maximum positions are stored for each joint. Serial communication is initialized at 4800 baud. The glove program reads the analog flex channels and both inertial devices, performs the start-up calibration and prints a command when a threshold condition is satisfied.

The implementation can be summarized by the following pseudocode:

1. Initialize I2C sensors, Bluetooth serial link and actuator controllers.
2. Capture baseline flex readings and calculate upper/lower thresholds.
3. Read three flex channels and both MPU6050 devices.
4. Evaluate finger and orientation conditions.
5. Transmit the corresponding command character.
6. At the arm, read the character and select a joint function.
7. Check the requested position against software limits.
8. Update the servo angle or generate stepper pulses.
9. Repeat continuously.

5.4 Operational and safety considerations

Although the prototype is a low-voltage educational platform, its moving joints can pinch fingers and its LiPo battery requires appropriate handling. Testing should begin with the arm elevated or unloaded, conservative joint increments and an accessible power-disconnect point. Servo horns and printed joints should be inspected for looseness before each run. The current prototype uses software limits but does not document physical limit switches, collision sensing or an emergency-stop circuit. It should therefore not be treated as an industrial safety-rated system.

6. Prototype Validation and Results

6.1 Validation approach

The submitted MSc records describe implementation and functional testing rather than a controlled user study. Validation consisted of powering the glove and arm, calibrating the flex sensors, performing the mapped hand and wrist movements, and observing whether the expected joint responded. The tests covered the wireless link, base rotation, shoulder and elbow movements, wrist motions and gripper operation. The project conclusion records successful implementation of the MARK 1 prototype and successful six-axis movement. Because raw logs, repeated trial counts and timing traces were not supplied, the results below are reported as functional outcomes and not as statistical performance measurements.

Table 5. Functional validation documented for the prototype

Test item	Expected behaviour	Documented outcome	Evidence type
Bluetooth command transfer	Glove command received by arm	Functionally achieved	Project report and code description
Base rotation	Left/right base movement	Functionally achieved after direction mapping	Implementation observations
Shoulder motion	Closer/farther arm movement	Functionally achieved	Project conclusion
Elbow motion	Bend/extend	Functionally achieved	Project conclusion
Wrist tilt	Up/down motion	Functionally achieved after mapping correction	Implementation observations
Wrist rotation	Clockwise/counter-clockwise	Functionally achieved	Project conclusion
Gripper	Open/close	Functionally achieved after flex-state correction	Prototype image and implementation notes
Six-axis coordinated platform	Multiple joint commands available	Reported as successfully implemented	Results and discussion section
Payload	600 g–1 kg target	Not quantitatively verified in supplied records	Design objective only
Accuracy and latency	Repeated quantitative metrics	Not measured in supplied records	Limitation

6.2 Direction-mapping fault and corrective action

The most clearly documented integration result was the discovery of reversed functions. The observed left/right, up/down and lateral gripper motions did not initially match the command descriptions. The project report identified two likely causes: physical sensor placement and the direction conditions used in the firmware. The correction exchanged the transmitted characters or the conditional branches associated with the affected orientation ranges. For example, the left/right command conditions and the gripper up/down conditions were swapped so that the physical movement matched the intended action.

This result illustrates a general issue in wearable teleoperation: sensor axes, human anatomical directions, robot joint coordinates and servo installation directions are separate coordinate systems. A command that is mathematically positive at the sensor can produce a mechanically negative joint rotation. The correction is therefore not merely a coding detail; it is a required coordinate-mapping step. A future version should formalize this mapping with a configuration table or calibration wizard rather than hard-coded command exchanges.

6.3 Functional outcome

The prototype established an end-to-end control chain from human gesture to physical robot motion. It demonstrated that three finger-bending channels and two inertial sensors can provide enough discrete inputs to operate the principal joints of a six-axis arm and its gripper. Bluetooth removed the tether between the operator and the arm, while the use of separate glove and arm controllers distributed the processing load. The architecture remained simple enough to be inspected and modified at source-code level.

The physical photographs and design archive also confirm that the work progressed beyond simulation. The gripper mechanism, bearings, servo mounts and printed links were assembled into a mechanical prototype. The base-actuator modification provides evidence of iterative engineering: the original actuator concept was changed after the real structure revealed a torque problem. Such adaptation is a meaningful outcome of prototyping because it cannot be identified from control software alone.

7. Discussion

The strongest feature of the developed system is its integration simplicity. Unlike camera-based interfaces, the glove is not constrained by line of sight, background clutter or illumination. Unlike commercial teleoperation systems, the controller does not require an industrial robot, proprietary glove or high-performance workstation. The sensor and command logic are executed on two small microcontrollers, and the wireless message is a single character rather than a high-bandwidth stream. This makes the platform suitable for teaching embedded robotics, demonstrating human–robot interaction and testing new joint mappings.

The design also shows the value of combining sensing modalities. Flex sensors are well suited to detecting finger states but do not describe wrist orientation. MPU6050 devices provide orientation and motion information but do not directly measure finger bending. Using both allows the glove to control a larger number of robot functions than either modality would conveniently provide alone. The approach is consistent with the broader literature on sensor gloves, in which flex and inertial sensing are frequently combined [10], [11].

Compared with Salman et al. [7], the present system applies similar low-cost and 3D-printing principles to a more articulated arm-and-gripper platform. Compared with Roda-Sanchez et al. [8] and Vysocký et al. [12], it sacrifices quantitative recognition and industrial path-planning capability in exchange for a smaller and more accessible architecture. Compared with Lu et al. [9] and Yu et al. [13], it avoids commercial haptic gloves and external optical tracking. These differences should not be interpreted as performance superiority; they identify a different design objective centred on accessibility and reproducibility.

The deterministic threshold method has advantages and disadvantages. It is transparent, computationally inexpensive and easy to debug. The user can see exactly which condition produces each command. However, it can be sensitive to sensor drift, glove fit, user anatomy and motion speed. It also converts continuous human movement into discrete increments, which may feel less natural than proportional joint mapping. The 1000 ms response delay documented in the source code is particularly conservative and would be unsuitable for fast teleoperation. It was appropriate for observable prototype testing but should be replaced by non-blocking timing and measured end-to-end latency.

The hybrid motor arrangement is another important design choice. Servos provide convenient position control for arm joints, but a stepper motor was required at the base when the assembled structure exceeded the practical servo torque. The change demonstrates that actuator selection must be based on the actual mass distribution and moment arm, not only on nominal motor specifications. The project would be strengthened by future torque calculations, current measurements and mechanical-load tests.

From a publication perspective, the current evidence supports a design-and-implementation paper but not claims of high recognition accuracy, low latency or payload capability. Those metrics were not present in the supplied project records. The manuscript therefore limits its claims to documented functionality. This distinction is essential for research integrity and provides a clear roadmap for the experiments required in the next prototype iteration.

8. Limitations and Future Work

The main limitation is the absence of a quantitative experimental dataset. The project records do not specify the number of gesture trials, participants, recognition accuracy, false-command rate, end-to-end latency, maximum wireless range, battery duration, positioning error or measured payload. These quantities should be measured before making comparative performance claims. A suitable protocol would recruit multiple users, repeat each gesture under several glove placements, record correct and incorrect commands, and measure the interval between sensor movement and actuator response with synchronized video or timestamped serial logs.

Second, the firmware relies on fixed empirical threshold multipliers and blocking delays. Future work should use moving-average or complementary filtering, non-blocking scheduling and a structured calibration routine. Proportional mapping could convert sensor angle to joint angle rather than producing only incremental commands. A machine-learning classifier could be investigated after a sufficiently large, labelled dataset is collected, but it should be compared against the existing threshold baseline rather than added without evidence.

Third, the mechanical and electrical safety features are limited. Future versions should add an emergency stop, physical joint limits or limit switches, current monitoring, battery protection, cable strain relief and collision detection. Quantitative torque and payload testing should be conducted with progressively increased loads and documented safety factors. The literal Bluetooth command table should also be generated automatically from a single firmware configuration to eliminate the label inconsistencies observed across the project documents.

Finally, the design can be extended toward task-level demonstrations such as pick-and-place, remote handling or assistive object transfer. Such demonstrations should include task-completion time, success rate and user workload. These experiments would move the platform from functional proof of concept toward a stronger human–robot interaction study.

9. Conclusion

This paper presented a low-cost wearable-sensor interface for wireless control of a six-axis 3D-printed robotic arm. The system combines three flex sensors and two MPU6050 inertial sensors on a glove, Arduino Nano and Uno controllers, Bluetooth communication, a PCA9685 servo driver, six servo motors and a NEMA 17 stepper motor. The completed MARK 1 prototype established an end-to-end path from human gesture to base, shoulder, elbow, wrist and gripper motion. The implementation also revealed and corrected reversed command directions caused by sensor orientation and mapping conventions.

The work's contribution is a transparent and accessible engineering implementation rather than a new classification algorithm. It demonstrates how wearable sensing, distributed embedded control, hybrid actuation and 3D-printed mechanics can be combined into a self-contained human–robot interaction platform. The current records support functional feasibility, while quantitative accuracy, latency, payload and multi-user testing remain necessary for a higher-level validation study.

Declarations

Author contribution: Agraram Ganesh Goud conceived, assembled, programmed, integrated and documented the prototype as part of his MSc project. The wording of this statement should be reviewed against any contribution made by the academic supervisor or technical staff before submission.

Funding: No external research funding was identified in the supplied project records.

Conflict of interest: The author declares no known competing interests.

Data and code availability: The project report, photographs, mechanical STL files and Arduino implementation materials are retained by the author. A public repository may be created after the final source code and licensing conditions of the open-source-derived mechanical design are reviewed.

Ethics: The reported prototype testing did not include a documented human-participant study. Any future multi-user evaluation should follow the relevant institutional ethics requirements.

References

[1] T. B. Sheridan, "Human–robot interaction: Status and challenges," Human Factors, vol. 58, no. 4, pp. 525–532, 2016, doi: 10.1177/0018720816644364.

[2] J. Berg and S. Lu, "Review of interfaces for industrial human–robot interaction," Current Robotics Reports, vol. 1, pp. 27–34, 2020, doi: 10.1007/s43154-020-00005-6.

[3] L. Dipietro, A. M. Sabatini, and P. Dario, "A survey of glove-based systems and their applications," IEEE Transactions on Systems, Man, and Cybernetics, Part C, vol. 38, no. 4, pp. 461–482, 2008, doi: 10.1109/TSMCC.2008.923862.

[4] J. P. Wachs, M. Kölsch, H. Stern, and Y. Edan, "Vision-based hand-gesture applications," Communications of the ACM, vol. 54, no. 2, pp. 60–71, 2011, doi: 10.1145/1897816.1897838.

[5] C. K. Mummadi, F. P. P. Leo, K. D. Verma, S. Kasireddy, P. M. Scholl, K. Kempfle, and K. Van Laerhoven, "Real-time and embedded detection of hand gestures with an IMU-based glove," Informatics, vol. 5, no. 2, art. 28, 2018, doi: 10.3390/informatics5020028.

[6] W.-C. Chuang, W.-C. Hwang, T.-M. Tai, D.-R. Huang, and Y.-L. Jhang, "Continuous finger gesture recognition based on flex sensors," Sensors, vol. 19, no. 18, art. 3986, 2019, doi: 10.3390/s19183986.

[7] F. Salman, Y. Cui, I. Zafar, F. Liu, L. Wang, and W. Wu, "A wireless-controlled 3D printed robotic hand motion system with flex force sensors," Sensors and Actuators A: Physical, vol. 309, art. 112004, 2020, doi: 10.1016/j.sna.2020.112004.

[8] L. Roda-Sanchez, C. Garrido-Hidalgo, A. S. García, T. Olivares, and A. Fernández-Caballero, "Comparison of RGB-D and IMU-based gesture recognition for human–robot interaction in remanufacturing," International Journal of Advanced Manufacturing Technology, vol. 124, pp. 3099–3111, 2023, doi: 10.1007/s00170-021-08125-9.

[9] C. Lu, L. Jin, Y. Liu, J. Wang, and W. Li, "Teleoperated grasping using data gloves based on fuzzy logic controller," Biomimetics, vol. 9, no. 2, art. 116, 2024, doi: 10.3390/biomimetics9020116.

[10] A. Filipowska et al., "Machine learning-based gesture recognition glove: Design and implementation," Sensors, vol. 24, no. 18, art. 6157, 2024, doi: 10.3390/s24186157.

[11] Z. R. Saeed et al., "A systematic review on systems-based sensory gloves for sign language pattern recognition: An update from 2017 to 2022," IEEE Access, vol. 10, 2022, doi: 10.1109/ACCESS.2022.3219430.

[12] A. Vysocký, S. Grushko, T. Spurný, R. Pastor, and T. Kot, "Hand gesture interface for robot path definition in collaborative applications: Implementation and comparative study," Sensors, vol. 23, no. 9, art. 4219, 2023, doi: 10.3390/s23094219.

[13] Z. Yu, C. Lu, Y. Zhang, and L. Jing, "Gesture-controlled robotic arm for agricultural harvesting using a data glove with bending sensor and OptiTrack systems," Micromachines, vol. 15, no. 7, art. 918, 2024, doi: 10.3390/mi15070918.

[14] H. Sümbül, "A novel MEMS and flex sensor-based hand gesture recognition and regenerating system using deep learning model," IEEE Access, vol. 12, pp. 133685–133693, 2024, doi: 10.1109/ACCESS.2024.3448232.

[15] P. Neto, M. Simão, N. Mendes, and M. Safeea, "Gesture-based human–robot interaction for human assistance in manufacturing," International Journal of Advanced Manufacturing Technology, vol. 101, pp. 119–135, 2019, doi: 10.1007/s00170-018-2788-x.

[16] M. Caeiro-Rodríguez, I. Otero-González, F. A. Mikic-Fonte, and M. Llamas-Nistal, “A systematic review of commercial smart gloves: Current status and applications,” Sensors, vol. 21, no. 8, art. 2667, 2021, doi: 10.3390/s21082667.