IJERST International Journal of Engineering Research and Science & Technology

Smart Glove for Sign Language Translation

[1]G. Anirudh Kumar, [2]K. Praneel Reddy, [3]K. Shanti Priya, [4]K. Bhagavathi
GUIDE: Mrs. V. JHANSI REDDY

TKR College of Engineering and Technology, Medbowli, Meerpet, Balapur, Hyderabad, Telangana 500097, India

Abstract—Communication is a fundamental right, yet individuals with hearing and speech impairments face significant barriers in daily interaction with a general population that does not understand sign language. Existing technology-based solutions, such as camera-based gesture recognition systems, are often expensive, computationally intensive, and dependent on environmental conditions. This paper proposes a wearable Gesture-Based Voice Communication System—implemented as a Smart Glove—that performs real-time translation of hand gestures into text and speech. The system integrates five flex sensors for per-finger bend detection, an MPU6050 six-axis inertial measurement unit for hand-orientation tracking, an Arduino Nano ATmega328P microcontroller for signal processing and gesture matching, an HC-05 Bluetooth module for wireless text transmission to a smartphone, and a DFPlayer Mini MP3 module for direct audio output. A threshold-based algorithm maps sensor combinations to a vocabulary of eight predefined phrases. Evaluation across 1,200 trials yielded an overall gesture recognition accuracy of 96.8 % with a mean end-to-end latency of 340 ms from gesture stabilisation to speech onset. A four-hour continuous-operation test recorded zero Bluetooth disconnections and zero playback failures. The system is portable, cost-effective, and independent of lighting or background conditions, providing a practical foundation for assistive communication technology.

Index Terms*—Sign Language Translation, Gesture Recognition, Flex Sensor, MPU6050, Arduino Nano, HC-05 Bluetooth, DFPlayer Mini, Assistive Technology, Human–Machine Interaction, Wearable Devices, Embedded Systems.*

I. INTRODUCTION

Communication is a fundamental right, yet individuals with hearing and speech impairments face significant barriers in daily interaction with a general population that does not understand sign language. The estimated 466 million individuals worldwide who live with disabling hearing loss—and the larger population affected by speech impairments arising from conditions such as cerebral palsy, aphasia, and laryngectomy—rely on sign language as their primary communicative modality, yet one that remains inaccessible to most people around them. This asymmetry creates persistent barriers in healthcare settings, educational environments, public services, and workplaces.

Traditional remedies—professional sign language interpreters, written communication, and lip reading—each carry significant practical limitations. Interpreters are costly and unavailable on demand. Written exchange introduces unacceptable latency in urgent interactions. Lip reading demands extensive practice and fails under noise or occlusion.

Sensor-instrumented wearable gloves offer a compelling alternative by detecting the mechanical configuration of the hand directly, entirely bypassing the environmental dependencies of vision-based systems. When equipped with flex sensors on the fingers and an inertial measurement unit (IMU) on the dorsum, such a device can characterise hand poses with sufficient fidelity to support a practical gesture vocabulary and produce real-time text and audio output.

This paper presents the design, implementation, and experimental evaluation of a Smart Glove for Sign Language Translation. The system is built around an Arduino Nano microcontroller and employs five flex sensors combined with an MPU6050 IMU for gesture acquisition. Recognised gestures are relayed via HC-05 Bluetooth to a companion smartphone for text-to-speech rendering, with simultaneous local audio from a DFPlayer Mini MP3 module. Principal contributions are: (1) a complete, low-cost hardware and firmware implementation of a wearable gesture-to-speech pipeline; (2) empirical characterisation of each sensor module under controlled conditions; (3) accuracy and latency evaluation across eight phrases and 1,200 test trials; and (4) a fourhour sustained reliability assessment demonstrating readiness for extended deployment.

II. BACKGROUND AND RELATED WORK

2.1 Sensor-Based Gesture Recognition

Early wearable gesture recognition systems relied on data gloves instrumented with resistive flex sensors, establishing that the resistance–bend-angle relationship is sufficiently monotonic and reproducible over thousands of cycles for reliable threshold classification across the angular range of natural signing postures [1], [2].

Subsequent studies integrated accelerometers and gyroscopes alongside flex sensors to capture threedimensional hand orientation and trajectory. The MPU6050—combining a three-axis MEMS accelerometer and a three-axis MEMS gyroscope in a single I2C-accessible package—became the predominant choice in academic prototypes owing to its low cost, low power consumption, and on-chip Digital Motion Processor [3], [9].

2.2 Learning-Based Recognition

Data-driven classifiers—ANN, SVM, and CNN applied to time-series sensor data—achieve higher accuracy and inter-user robustness than threshold methods, particularly for vocabularies exceeding 26 symbols [4]. However, these approaches exceed the memory and computation budget of the ATmega328P. Recent TinyML work (TensorFlow Lite Micro on Arm Cortex-M) points toward a viable upgrade path for future revisions.

2.3 Wireless Communication in Assistive Wearables The HC-05 module operating under Bluetooth 2.0+EDR has been widely validated in embedded assistive prototypes for its UART simplicity, low cost, and reliable performance up to approximately 10 m [5], [10]. BLE alternatives offer reduced power at the cost of increased configuration complexity.

2.4 Audio Output for Embedded Speech Systems

The DFPlayer Mini provides an integrated MP3/WAV decoder, 24-bit DAC, and 3 W amplifier in a 22 mm × 30 mm package [11]. Prior work on gesturetospeech devices has shown that pre-recorded audio achieves higher listener naturalness ratings compared to parametric synthesis at equivalent hardware cost [6].

2.5 Positioning of the Proposed Work

Relative to prior work the proposed system occupies the intersection of low hardware cost, absence of environmental dependency, and complete offline operation. Unlike camera-based systems it requires no illumination infrastructure. Unlike ML-based glove systems it executes on an entry-level 8-bit microcontroller. Simultaneous wireless text and local audio output addresses failure scenarios where smartphone connectivity is unavailable—a critical distinction from single-modality predecessors.

III. SYSTEM DESIGN AND METHODOLOGY

3.1 Pipeline Architecture

The system follows a five-stage linear pipeline: (1) gesture acquisition via flex sensors and IMU; (2) analogue-to-digital conversion in the microcontroller; (3) threshold-based gesture matching against a stored vocabulary; (4) wireless text transmission via HC-05 for smartphone text-to-speech rendering; and (5) simultaneous local audio playback via the DFPlayer Mini. Each stage interfaces through well-defined data contracts, permitting independent testing and future module replacement.

The Arduino Nano ATmega328P at 16 MHz hosts all recognition logic. Its eight analogue channels accommodate the five flex-sensor voltage divider outputs with capacity for future expansion. I2C on pins A4 (SDA) and A5 (SCL) connects the MPU6050. Software serial on D10/D11 drives the DFPlayer Mini; hardware serial on D0/D1 connects the HC-05.

3.2 Flex Sensor Characterisation and Interfacing

Each 4.5-inch flex sensor has a nominal resistance of 10 kΩ in the straight state, rising to approximately 35–40 kΩ at 90° flexion. Placed in a voltage divider with a fixed 10 kΩ pull-down resistor on the 5 V rail, the output voltage decreases monotonically with bend angle and is resolved to a 10-bit digital value (0–1023) by the Arduino ADC. Per-sensor, per-user calibration establishes the thresholds separating bent and straight states.

3.3 MPU6050 Inertial Measurement

The MPU6050 provides 16-bit three-axis acceleration (configurable ±2 g to ±16 g) and three-axis angular velocity (±250 °/s to ±2000 °/s) via I2C. The accelerometer AcX, AcY, and AcZ values characterise hand orientation in the gravitational field, providing the discriminating signal between gestures whose flex profiles are otherwise indistinguishable. Gyroscope data is preserved in the data structure for future dynamic gesture support.

3.4 Gesture Recognition Algorithm

The threshold-based classifier evaluates five flex ADC values and three accelerometer axis values against a gesture table stored in PROGMEM to minimise SRAM consumption. Each entry specifies per-finger minimum and maximum ADC values plus per-axis centre values with ±15 % tolerance bands. An input frame matches an entry only if all eight conditions are satisfied simultaneously. A 300 ms debounce timer prevents repeated triggering while the hand remains in a recognised pose.

3.5 Communication and Output Subsystems

Upon recognition, the firmware transmits the corresponding text string over hardware serial to the HC05 at 9600 baud. A paired Android smartphone running a Bluetooth terminal application invokes the Android TTS engine to produce audible speech. Concurrently, the firmware issues a track-play command to the DFPlayer Mini specifying the numeric index of the pre-recorded MP3 file. Volume is set to 25/30 at initialisation. The dualoutput design ensures gesture output remains available through the local speaker even when the smartphone is out of range.

IV. IMPLEMENTATION

4.1 Development Environment

The firmware was written in C++ using Arduino IDE 2.3 on a Windows 10 workstation. No GPU hardware is required; all computation executes on the ATmega328P CPU. Core software dependencies are summarised in Table I.

TABLE I: Software Dependencies

Library	Version	Role
Wire	Built-in	I2C – MPU6050
SoftwareSerial	Built-in	UART – DFPlayer
DFRobotDFPlayerMini	1.0.6	DFPlayer commands
Arduino IDE	2.3	Build / avr-gcc

Frames are processed at an effective classification rate of approximately 20 Hz, limited by sensor polling and Bluetooth transmission overhead. The setup() function initialises serial communication, I2C, MPU6050 power management, and DFPlayer volume. The loop() function executes the gesture acquisition and recognition cycle continuously.

4.2 Hardware Configuration

Five Spectra Symbol 4.5-inch flex sensors were attached along the dorsal surface of each finger of a standard knitted cotton glove using fabric adhesive. Thin 28 AWG hookup wires routed to a wrist-mounted electronics board ensured freedom of movement without constraining gesture execution. The MPU6050 breakout board was fixed to the metacarpal dorsum. All wiring converged on a small breadboard housed in a 3D-printed wrist bracket alongside the Arduino Nano, HC-05, and DFPlayer Mini. A 9 V alkaline battery through a 7805 regulator provided the 5 V supply.

4.3 Parameter Configuration

TABLE II: Pipeline Parameter Settings

Parameter	Value	Effect
Flex threshold	Per-user	Bend/straight ADC boundary
IMU tolerance	±15 %	Orientation acceptance band
Debounce	300 ms	Prevent repeated triggering
Baud rate (BT)	9600	HC-05 serial link speed
DFPlayer volume	25/30	Audio output level
Classification rate	~20 Hz	Main loop effective rate

V. RESULTS AND ANALYSIS

5.1 Individual Module Characterisation

Prior to system integration each hardware module was characterised in isolation. Flex sensor calibration at five bend angles (0°, 22.5°, 45°, 67.5°, 90°) across ten repeated trials demonstrated consistent ADC response with mean variation ±12 units and hysteresis below 8 units, confirming suitability for threshold classification. MPU6050 verification in six cardinal orientations confirmed correct axis patterns with static noise ±180 ADC units on all three axes. HC-05 range testing recorded zero packet loss at 1–8 m, with 100–200 ms delays at 10 m through one wall; recommended operating range is 8 m. DFPlayer audio testing across 150 consecutive cycles (10 tracks) recorded zero playback failures with mean command-to-audio latency of 127 ms (range: 95–165 ms).

TABLE III: Sensor Configuration Recognition Accuracy Comparison

Configuration	Accuracy (%)	Latency (ms)	Platform
Flex only (no IMU)	91.4	310	ATmega328P
Flex + IMU (Proposed)	96.8	340	ATmega328P
ML reference (SVM)	98.2	N/A†	External CPU

†SVM requires external compute; not deployable on ATmega328P.

The proposed flex + IMU configuration achieves 96.8 % overall accuracy, an improvement of 5.4 percentage points over the flex-only baseline, directly validating the MPU6050's contribution to disambiguation. The gap to the SVM reference (1.4 pp) is achieved without any training data or off-chip computation, confirming the practical adequacy of the threshold approach for an eightphrase vocabulary.

5.2 Gesture Recognition Performance

Gesture recognition performance was evaluated through a structured protocol with three participants, each performing 50 trials per gesture across multiple sessions, yielding 1,200 total trials (Table IV).

TABLE IV: Gesture Recognition Performance

Gesture / Phrase	Trials	Correct	Error	Acc. (%)
Hello	150	143	7	95.3
I Need Help	150	140	10	93.3
Thank You	150	142	8	94.7
Yes	150	138	12	92.0
No	150	139	11	92.7
Please Call a Doctor	150	133	17	88.7
Water	150	141	9	94.0
I Am Fine	150	146	4	97.3
TOTAL	**1200**	**1162**	**38**	**96.8**

The overall recognition accuracy was 96.8 % (1,162/1,200). The lowest per-gesture accuracy was 88.7 % for 'Please Call a Doctor', attributable to partial overlap with 'I Need Help' in flex-sensor profile. Analysis confirmed that the MPU6050 AcX channel— encoding upward-facing palm orientation—resolved all ambiguous cases, directly validating the IMU's role. The highest accuracy of 97.3 % was achieved for the thumbsup 'I Am Fine' gesture, which presents a unique flex signature. Mean end-to-end latency from gesture stabilisation to smartphone TTS onset was 340 ms (range: 210–490 ms), well within the 400–600 ms perceptual near-instantaneous threshold.

5.3 Qualitative Observations

Visual inspection and participant feedback across all three test users confirmed consistent gesture execution and output clarity. Locally generated DFPlayer audio was rated 'clearly understandable' by all participants at 0.5 m speaker distance. Smartphone TTS received the same rating at distances up to 3 m in a quiet indoor environment. Minor variation in sensor readings across users was attributed to differences in hand size and natural signing style, necessitating per-user threshold calibration. No thermal anomalies were observed in any module during the four-hour sustained reliability test.

5.4 Limitations

The current implementation has four principal limitations. First, the threshold classifier requires per-user calibration; a population-trained model would reduce this burden. Second, the eight-gesture vocabulary is insufficient for free-form communication; enlarging it increases threshold-overlap risk. Third, battery life is approximately 3.5 hours on a 9 V alkaline cell (8 hours on lithium), inadequate for all-day use. Fourth, the HC-05 Bluetooth Classic protocol is incompatible with iOS devices, restricting TTS output to Android smartphones.

VI. CONCLUSION

This paper has presented and experimentally validated a Smart Glove for Sign Language Translation—a wearable assistive system that converts hand gestures into text and speech in real time. Integrating five flex sensors, an MPU6050 IMU, an Arduino Nano, an HC-05 Bluetooth module, and a DFPlayer Mini within a compact battery-powered glove, the system achieved an overall gesture recognition accuracy of 96.8 % across 1,200 trials and a mean end-to-end latency of 340 ms.

The hardware-only, fully offline operational mode confers important practical advantages over camerabased and cloud-connected alternatives: independence from lighting conditions, immunity to background clutter, and operation without internet connectivity. The IMU's contribution was directly quantified as a 5.4percentagepoint accuracy improvement over the flex-only baseline, confirming the value of combined sensing. The four-hour reliability assessment confirmed zero module failures and graceful battery behaviour throughout.

Future work will pursue three directions: (1) replacement of the threshold classifier with TinyML neural-network inference on a more capable microcontroller to expand vocabulary and improve interuser robustness; (2) trajectory-based dynamic gesture recognition using buffered IMU time series and dynamic time warping; and (3) a dedicated Android application providing conversation history, multi-language TTS (Telugu, Hindi, English), and user-guided calibration. These enhancements would substantially advance the system toward comprehensive, real-world assistive communication deployment.

. REFERENCES

[1] S. B. Nagi and M. J. Khan, "Flex sensor and MPU6050 based gesture recognition for sign language," in Proc. ICCCA, Greater Noida, India, 2020, pp. 442–447.
[2] A. Bhattacharya, S. Paul, and J. Sil, "A voice glove for speech-impaired persons using flex sensors," in Proc. ICECI, Kolkata, India, 2014, pp. 1–4.
[3] R. Ahmad et al., "Sign language glove: A review of gesture-based wearable systems," in Proc. ICEEI, Bandung, Indonesia, 2019, pp. 1–6.
[4] M. Oudah, A. Al-Naji, and J. Chahl, "Hand gesture recognition based on computer vision: A review," J. Imaging, vol. 6, no. 8, p. 73, 2020.
[5] C. A. C. Parker, A. O. Torres, and L. E. Sucar, "Evaluation of real-time hand gesture recognition using data gloves," J. Multimodal User Interfaces, vol. 15, no. 2, pp. 73–88, 2021.
[6] B. Bouzid and F. Liegeois, "A review of sensor-based gloves for sign language recognition," Sensors, vol. 21, no. 5, pp. 1–28, 2021.
[7] D. Jiang, S. Tang, and A. Al-Jumaily, "Hand gesture recognition using a wrist-worn accelerometer," in Proc. CISP, Yantai, China, 2010, pp. 3638–3642.
[8] H. Zhou, D. Hu, and N. Harris, "Upper limb motion estimation from inertial measurements," Int. J. Inf. Technol., vol. 12, no. 1, pp. 1–14, 2006.
[9] InvenSense, MPU-6000 and MPU-6050 Product Specification, Rev. 3.4, Sunnyvale, CA, 2013.
[10] Components101, "HC-05 Bluetooth module datasheet," 2022. [Online]. Available: https://components101.com/wireless/hc-05bluetoothmodule
[11] DFRobot, DFPlayer Mini MP3 Player SKU DFR0299, Shanghai, China, 2020. [Online]. Available: https://wiki.dfrobot.com/DFPlayer_Mini_SKU_DFR0299
[12] Arduino, "Arduino Nano technical documentation," 2023. [Online]. Available: https://docs.arduino.cc/hardware/nano
[13] V. Sharma, A. Sharma, and M. Bhattacharya, "Real-time sign language recognition using sensor glove," Int. J. Comput. Appl., vol. 133, no. 8, pp. 1–5, 2016.