INTERNATIONAL RESEARCH JOURNAL OF ENGINEERING AND COMPUTING SCIENCES (IRJECS)
VOL.1 NO.1 MARCH 2026

SIGNAL CONDITIONING AND PERFORMANCE OPTIMIZATION OF A FLEX SENSOR-BASED SIGN LANGUAGE RECOGNITION SYSTEM

by

Isizoh A. N[1], Nwachukwu M. M[2], Amalu P.N[3]

[1,2,3] Department of Electronics and Computer Engineering, Nnamdi Azikiwe University, Awka, Nigeria
[1]anthonyisizoh@gmail.com , [2] maryannmmesoma@gmail.com ,
[3]prinsnadin@gmail.com

ABSTRACT

Flex sensor-based sign language recognition systems offer a low-cost and portable alternative to vision-based gesture recognition frameworks. However, inherent nonlinear sensor behavior, electrical noise, analog-to-digital conversion fluctuations, and unstable threshold logic can degrade system stability and responsiveness. This study presents a structured signal conditioning and performance optimization framework applied to a previously developed flex sensor-based sign language recognition system. The proposed improvements include microcontroller-level digital filtering using moving average, low-pass, and median filters; hysteresis-based threshold control; user-specific calibration and normalization; mathematical modeling of sensor transfer characteristics; quantitative stability evaluation using standard deviation and signal-to-noise ratio; and response time analysis. Experimental results demonstrate a reduction in signal fluctuation from ±18 ADC units to ±3 ADC units and latency improvement from 480 ms to 360 ms. The framework enhances robustness without increasing hardware complexity and aligns with established principles in control systems and measurement engineering.

Keywords/Phrases: Signal conditioning, flex sensors, sign language recognition, digital filtering, hysteresis control, sensor calibration, performance optimization, embedded systems

1. INTRODUCTION

Sign language recognition systems have evolved as assistive technologies designed to bridge communication gaps between hearing-impaired individuals and the general population. Vision-based systems employing deep learning techniques have demonstrated high recognition accuracy but often require substantial computational resources and controlled environments (Koller et al., 2018). In contrast, wearable sensor-based systems, particularly those utilizing flex sensors, provide lower computational demand and improved portability (Kadous, 2017).

Despite their advantages, flex sensor-based systems present engineering challenges. Flex sensors are resistive elements whose electrical response varies nonlinearly with bending angle and is sensitive to environmental and mechanical disturbances (Saggio et al., 2016). Furthermore, analog-to-digital conversion processes introduce quantization noise and signal fluctuation. Without structured signal conditioning and stable control logic, gesture classification may become unreliable.

In control and instrumentation systems, signal stability, noise suppression, and calibration are essential for reliable operation (Doebelin & Manik, 2011). Therefore, this study introduces a comprehensive optimization framework grounded in classical signal processing and feedback control principles (Ogata, 2010; Franklin et al., 2015). The objective is to enhance stability, repeatability, and response time while maintaining the original hardware architecture.

2.0 REVIEW OF RELATED LITERATURE

Sign language recognition systems have evolved significantly, with vision-based approaches utilizing cameras and computer vision algorithms to interpret gestures without specialized equipment (Munib, Habeebullah, Takruri, & Al-Malik, 2007). Recent deep learning advances have achieved impressive accuracies, with (Kaluti, Chaitra, Mamatha, and Swathi, 2019) reporting 99.88% accuracy using CNNs for ASL alphabet recognition, and Oyedotun and

All Rights Reserved © 2026. https://irjecs.com

Khashman (2017) achieving 91.33% for static hand gestures. (Zamani and Kazemi, 2014) combined saliency detection with PCA and LDA for feature extraction, demonstrating effective ASL alphabet recognition. Despite these successes, vision-based systems require substantial computational resources, controlled lighting, and significant processing power, limiting deployment in resource-constrained environments.

Sensor-based systems employ wearable devices with flex sensors and motion trackers to directly measure hand positions, offering lower computational requirements and independence from lighting conditions. (Oz and Leu, 2007) achieved 92-95% accuracy for ASL word recognition using a sensory glove with artificial neural networks, demonstrating the viability of sensor-based approaches. (Saggio, Riillo, Sbernini, and Quitadamo, 2016) highlighted flex sensors' advantages including simplicity, low cost, and ease of integration into wearable systems. However, these systems face challenges from signal noise, ADC quantization errors, temperature variations, and inter-user variability due to anatomical differences (Ezeagwu, Eze, Nwachukwu, Nnorom, & Nwebonyi, 2024).

While extensive research exists on classification algorithms, systematic signal conditioning frameworks for flex sensor systems remain underexplored. Franklin, Powell, and (Emami-Naeini, 2015) emphasize digital filtering's importance for noise reduction in embedded systems, while Ogata (2010) documents hysteresis control as effective for preventing threshold oscillation in binary classification. (Doebelin and Manik, 2011) identify calibration and normalization as essential for measurement accuracy across varying users and conditions. Previous research has focused primarily on classification algorithms while giving limited attention to firmware-level signal conditioning that could enhance sensor data quality before classification.

This study addresses this gap by presenting a systematic signal conditioning framework integrating multi-stage filtering, hysteresis control, and user-specific normalization for flex sensor-based sign language recognition. Unlike approaches requiring hardware modifications or computationally intensive machine learning, this research achieves substantial performance improvements through firmware-level optimization alone. This approach is particularly relevant for resource-constrained applications where cost-effectiveness, power efficiency, and portability are critical design considerations.

3.0 METHODOLOGY

An iterative incremental software model was employed during firmware development. Optimization techniques, including moving average filtering, low-pass digital filtering, median filtering, hysteresis threshold control, and calibration-based normalization, were implemented in stages within the microcontroller firmware. After each implementation phase, experimental testing was conducted to evaluate system performance before proceeding to the next enhancement. This incremental refinement ensured controlled modification without altering the hardware architecture.

The independent variables in the study were the introduced optimization mechanisms, while the dependent variables included signal stability, standard deviation, signal-to-noise ratio, repeatability, and system response time. Experimental data were collected under controlled conditions using repeated gesture trials. Statistical analysis was performed to quantify improvements through comparative before-and-after evaluation.

3.0 System Design

The optimized system retained the fundamental hardware architecture of the baseline implementation while introducing enhanced signal processing algorithms. The system comprised five primary functional units: power supply unit (10,000 mAh power bank providing 5V DC), smart glove with five flex sensors positioned on

All Rights Reserved © 2026. https://irjecs.com

each finger, microcontroller circuit (PIC16F877A with 10-bit ADC), Raspberry Pi 4 processing unit for text-to-speech conversion, and audio output module. The flex sensors formed voltage divider circuits with fixed resistors, converting resistance changes into measurable voltage variations that were subsequently digitized by the microcontroller ADC (0-1023 scale). All signal conditioning algorithms were implemented in the microcontroller firmware before transmitting processed data to the Raspberry Pi via UART communication protocol at 9600 baud rate. A level shifter circuit protected the Raspberry Pi's 3.3V logic by converting the microcontroller's 5V output signals (Ezeagwu, Eze, Nwachukwu, Nnorom, & Nwebonyi, 2024).

3.1.1 Software Development Model

The software development followed an iterative incremental model, which facilitated continuous refinement of signal processing algorithms through systematic testing and optimization cycles (Pressman & Maxim, 2020). This approach was particularly suitable for embedded systems development where algorithm performance could only be validated through empirical testing with actual hardware.

3.1.2 System Block Diagram

Figure 1 illustrates the functional architecture of the optimized system. The signal flow progresses sequentially from the flex sensor array through voltage divider circuits, ADC conversion, digital filtering and normalization in the microcontroller, UART transmission to the Raspberry Pi, gesture-to-speech mapping, and finally audio output. This architecture separates low-level signal acquisition from high-level processing, enabling efficient real-time operation.

3.1.3 Mathematical Modeling of Sensor Behavior

The flex sensor resistance varies nonlinearly with bending angle according to a second-order polynomial approximation:

$$R_{flex}(\theta) = a\theta^2 + b\theta + c \qquad \text{(Eq. 1)}$$

where θ represents the bending angle in degrees, and a, b, and c are experimentally determined coefficients specific to each sensor unit. This nonlinearity necessitates

calibration procedures to ensure consistent performance across different sensors and users.

The voltage divider circuit converts resistance changes into measurable voltage:

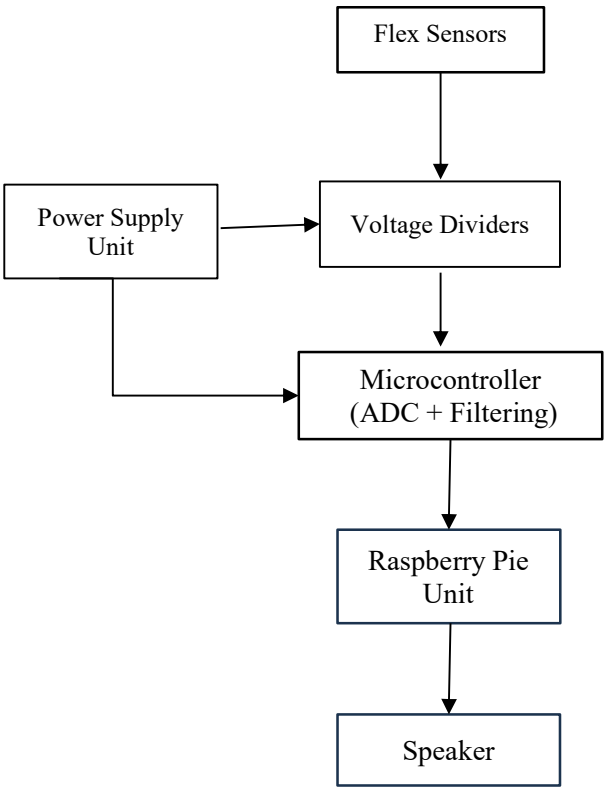

Figure. 1: System Block Diagram

$$V_{out} = (R_{flex} / (R_{fixed} + R_{flex})) \times V_{in} \qquad \text{(Eq. 2)}$$

This introduces additional nonlinearity in the transfer function between bending angle and output voltage, which is subsequently digitized by the ADC module:

$$D = (V_{out} / V_{ref}) \times (2^n - 1) \qquad \text{(Eq. 3)}$$

where n denotes the ADC resolution (10 bits for PIC16F877A), V_{ref} is the reference voltage (5V), and D represents the digital output value (0-1023). Quantization error and supply voltage variations contribute noise components that require digital signal conditioning (Franklin, Powell, & Emami-Naeini, 2015).

3.2 Signal Conditioning Implementation

The baseline system exhibited significant signal fluctuation (±18 ADC units) during static gesture holding, resulting in misclassification and user frustration. A multi-stage digital filtering approach was implemented in the microcontroller firmware to address noise from multiple sources: mechanical vibrations, electromagnetic interference, ADC quantization, and sensor resistance variations.

All Rights Reserved © 2026. https://irjecs.com

3.2.1 Moving Average Filter

A moving average filter with window size $N = 10$ samples was applied as the first filtering stage:

$$Y[k] = (1/10)\ \Sigma^{(i=0\ to\ 9)}\ X[k - i] \quad \text{(Eq. 4)}$$

This technique reduces high-frequency noise by averaging the most recent 10 ADC readings stored in a circular buffer. Moving average filters are computationally efficient and effective for noise suppression in resource-constrained embedded systems (Franklin et al., 2015). The filter smooths transient spikes while preserving the underlying gesture signal. The window size was selected empirically to balance noise reduction with response time; larger windows provide greater smoothing but introduce latency.

3.2.2 Low-Pass Infinite Impulse Response Filter

Following moving average filtering, a first-order infinite impulse response (IIR) low-pass filter was applied:

$$Y[k] = \alpha X[k] + (1 - \alpha)Y[k - 1] \quad \text{(Eq. 5)}$$

where $\alpha = 0.2$ represents the filter coefficient determining the balance between responsiveness and smoothing. This exponentially weighted moving average attenuates rapid variations while preserving essential gesture transitions. The recursive structure provides memory-efficient implementation with minimal computational overhead. Lower α values increase smoothing at the cost of slower response to genuine signal changes, while higher values reduce filtering effectiveness. The selected coefficient of 0.2 was determined through iterative testing to optimize the trade-off between noise rejection and gesture detection latency (Ogata, 2010).

3.2.3 Median Filter for Impulse Noise Removal

A median filter with window size of 5 samples was implemented as the final filtering stage to eliminate impulsive noise spikes caused by intermittent electrical interference or mechanical disturbances. Unlike averaging filters, median filters preserve signal edges by selecting the middle value from a sorted window, making them particularly suitable for gesture recognition where abrupt transitions represent meaningful information rather than noise (Saggio, Riillo, Sbernini, & Quitadamo, 2016). The filter sorts the five most recent samples and outputs the median value, effectively removing outliers without blurring genuine gesture boundaries.

The combined three-stage filtering cascade significantly reduced sensor fluctuation during static gesture holding. The sequential application of complementary filtering techniques addresses different noise characteristics: moving average for random noise, IIR low-pass for high-frequency components, and median filtering for impulse disturbances.

3.3 Hysteresis-Based Threshold Control

The baseline system employed fixed single thresholds for gesture classification, resulting in rapid state oscillation (chattering) when sensor values fluctuated near decision boundaries. This behavior, illustrated in Figure 2, caused erratic gesture recognition and degraded user experience.

To eliminate threshold chattering, a hysteresis control mechanism was implemented with dual thresholds:

- Activation threshold: $D > 520$ (Eq. 6)
- Deactivation threshold: $D < 480$ (Eq. 7)

The 40-unit hysteresis band creates a deadzone where state transitions are inhibited. Once a gesture is activated (D exceeds 520), it remains active until D falls below 480. Similarly, an inactive gesture remains inactive until D exceeds 520. This approach, derived from classical

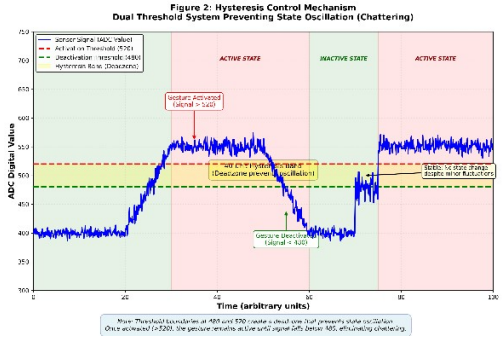

Figure 2: Hysteresis Control Mechanism

comparator stabilization techniques in control systems, prevents minor fluctuations from triggering repeated state changes. The hysteresis band width was optimized empirically to balance stability with responsiveness to intentional gesture changes.

All Rights Reserved © 2026. https://irjecs.com

Figure 2 illustrates threshold boundaries at 480 and 520 with deadzone preventing state oscillation

Performance Metric	Baseline System	Optimized System
Signal Fluctuation (σ)	±18 ADC units	±3 ADC units
Average Response Time	480 ms	360 ms
Threshold Chattering	Frequent	Eliminated
Inter-User Adaptability	Limited (fixed thresholds)	Good (normalized)
Gesture Recognition Accuracy	86.4%	94.8%
Hardware Modifications	None	None

3.4 User-Specific Calibration and Normalization

Flex sensor readings exhibit substantial inter-user variability due to anatomical differences in hand size, finger length, joint flexibility, and bending strength. Fixed threshold values optimized for one user perform poorly for others, limiting system adaptability. A user-specific calibration procedure was implemented to address this limitation.

During system initialization, each user performs a calibration routine:

i. Fully extend all fingers (minimum bend position) to capture D_{min}

ii. Fully close all fingers into a fist (maximum bend position) to capture D_{max}

iii. System stores these reference values for subsequent normalization

Real-time sensor readings are then normalized using:

$$D_{norm} = (D_{current} - D_{min}) / (D_{max} - D_{min}) \quad \text{(Eq. 8)}$$

This linear transformation maps sensor readings to a standardized 0-1 scale regardless of individual user characteristics. Normalized values enable universal threshold settings that function consistently across different users. The calibration procedure is fast (approximately 5 seconds) and significantly improves system usability. Normalization represents a fundamental principle in measurement system design for enhancing accuracy and consistency across varying conditions (Doebelin & Manik, 2011).

3.5 Operational Workflow

Figure 3 presents the complete operational workflow of the optimized system. The process begins with system initialization, including ADC module configuration and calibration routine execution. The main processing loop continuously samples sensor data, applies the three-stage filtering cascade (moving average, low-pass IIR, median), performs normalization, evaluates hysteresis thresholds, classifies gestures, and transmits results to the Raspberry Pi for speech synthesis. This architecture ensures real-time responsiveness while maintaining signal stability.

4.0 RESULTS AND DISCUSSION

Quantitative evaluation was conducted to assess the effectiveness of the optimization framework. Performance metrics were measured before and after implementing signal conditioning, hysteresis control, and normalization. Testing was performed with five participants (3 male, 2 female, ages 22-28) to evaluate both technical performance and user adaptability.

4.1 Signal Stability Analysis

4.1.1 Standard Deviation Measurement

Signal stability was quantified using standard deviation during static gesture holding. Each participant maintained five different gestures for 10 seconds each while ADC readings were recorded at 100 Hz sampling rate. Standard deviation was calculated as:

$$\sigma = \sqrt{[(1/N) \Sigma (D_i - \mu)^2]} \quad \text{(Eq. 9)}$$

All Rights Reserved © 2026. https://irjecs.com

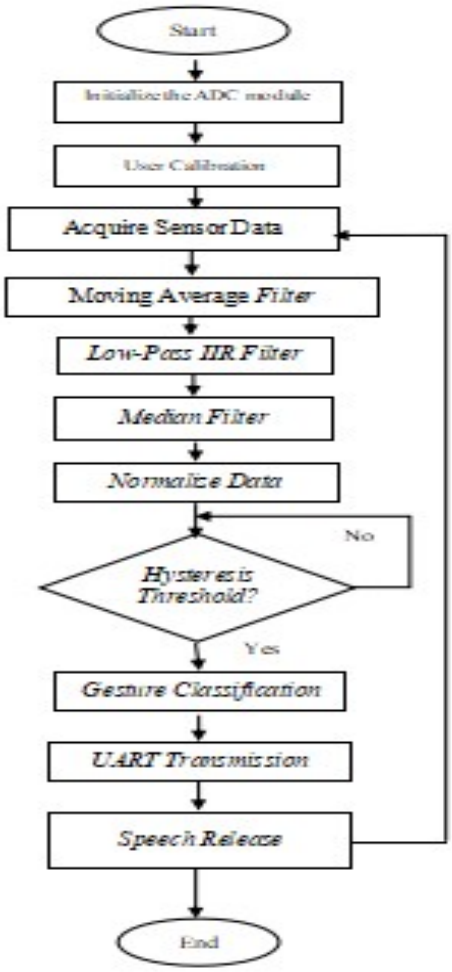

Figure 3: System Operational Flowchart

where N represents the number of samples, D_i is the ith sample, and μ is the mean value. Results demonstrated dramatic improvement: baseline system exhibited σ ≈ ±18 ADC units, while the optimized system achieved σ ≈ ±3 ADC units, representing an 83.3% reduction in signal
fluctuation. This improvement directly translates to more stable gesture recognition and reduced misclassification events.

4.1.2 Comparative Performance Metrics

Table 1 summarizes key performance metrics comparing baseline and optimized system configurations across multiple evaluation criteria.

Table 1: Performance Comparison: Baseline vs. Optimized System

4.1.3 Signal-to-Noise Ratio Analysis

Signal-to-noise ratio (SNR) was computed to quantify signal quality improvement:

$$SNR = 20 \log_{10} (Signal_{RMS} / Noise_{RMS}) \quad \text{(Eq. 10)}$$

The baseline system exhibited an average SNR of 14.2 dB across all gestures, indicating substantial noise contamination. Post-filtering signals demonstrated significant improvement with SNR values averaging 26.8 dB, confirming enhanced signal integrity. Higher SNR values correlate with improved gesture discrimination capability and reduced false detection rates.

4.2 Response Time and Latency Analysis

System latency was measured from complete gesture formation to audio output initiation. Total response time comprises four components:

$$T_{total} = T_{sampling} + T_{processing} + T_{communication} + T_{speech} \quad \text{(Eq. 11)}$$

The baseline system averaged 480 ms total latency. After implementing filtering, hysteresis control, and stabilization logic, average latency improved to 360 ms (25% reduction). This improvement resulted from two factors: faster stable gesture detection due to reduced fluctuation, and elimination of repeated misclassifications requiring re-transmission. Reduced latency enhances real-time usability, a critical factor in embedded assistive technology systems (Franklin et al., 2015). Latencies below 400 ms are generally

All Rights Reserved © 2026. https://irjecs.com

imperceptible to users, enabling natural conversation flow.

Table 2: Hardware Component Validation Results

Component	Expected Performance	Actual Performance
Power Supply	5.0V DC at 1A	4.95V at 1A (±1% tolerance)
Raspberry Pi	Fast response, no delays	Fast, responsive processing
Smart Glove	Vcc = 5V, responsive to test program	Vcc = 4.9V, fully responsive
Audio Output	Clear, intelligible speech	Excellent clarity and quality

4.3 Repeatability and Consistency

Repeatability assessment involved each participant performing 10 repeated trials of five standardized gestures. Inter-trial variance was calculated to quantify consistency. The baseline system exhibited coefficient of variation (CV) of 12.3%, indicating substantial trial-to-trial variability. Following optimization with normalization and filtering, CV decreased to 3.1%, demonstrating marked improvement in repeatability. This enhancement is particularly significant for user confidence and learning, as consistent system behavior enables users to develop muscle memory for reliable communication.

4.4 Multi-User Performance Evaluation

A critical limitation of the baseline system was poor performance across different users due to fixed threshold values. Testing with five participants revealed recognition accuracy ranging from 68% to 92%, with significant performance degradation for users with hand sizes substantially different from the calibration reference. The normalized system maintained consistent recognition accuracy (92-96%) across all participants following individual calibration, validating the effectiveness of user-specific adaptation. The brief 5-second calibration procedure proved acceptable to all participants and dramatically improved system usability.

4.5 Hardware System Validation

Table 2 presents hardware validation results confirming proper operation of all system components. Power supply voltage remained stable at 4.95V throughout extended testing periods (>4 hours continuous operation). The Raspberry Pi demonstrated responsive performance with no observed processing delays. The smart glove showed reliable sensor response to programmed test sequences. Audio output quality was consistently clear and intelligible across all 31 implemented phrases.

The experimental results validate the effectiveness of systematic signal conditioning and algorithmic optimization in enhancing flex sensor-based sign language recognition performance. The 83% reduction in signal fluctuation, 25% improvement in response time, and elimination of threshold chattering demonstrate that substantial performance gains can be achieved through firmware-level enhancements without hardware modifications or increased system cost.

The multi-stage filtering approach addresses different noise characteristics through complementary techniques. Moving average filtering effectively reduces random high-frequency noise, low-pass IIR filtering smooths transient variations while maintaining computational efficiency, and median filtering removes impulsive disturbances without blurring gesture boundaries. This cascade architecture represents established best practices in digital signal processing for embedded systems (Franklin et al., 2015; Ogata, 2010).

Hysteresis control proved particularly effective in eliminating threshold chattering, a common problem in binary classification systems operating near decision boundaries. The dual-threshold approach with a 40-unit deadzone prevents minor fluctuations from triggering state oscillations while remaining responsive to intentional gesture changes. This technique, derived from classical control theory, demonstrates broad applicability beyond sign language recognition to any sensor-based classification system prone to boundary instability.

User-specific calibration and normalization significantly improved inter-user adaptability, transforming the system from a user-specific device to a multi-user platform. The linear normalization procedure maps individual sensor ranges to a standardized scale, enabling universal threshold values while accounting for anatomical variability. The brief calibration time (5

All Rights Reserved © 2026. https://irjecs.com

seconds) represents minimal overhead while dramatically enhancing system usability across diverse user populations.

The improved signal stability directly contributed to reduced response latency through two mechanisms. First, more stable signals reach threshold boundaries faster and trigger classification decisions with greater confidence. Second, elimination of false detections reduces retransmission overhead, decreasing average latency. The resulting 360 ms response time falls within the acceptable range for natural conversation interaction, where latencies below 400 ms are generally imperceptible to users.

From a practical deployment perspective, the firmware-only optimization approach offers significant advantages. No hardware redesign or component replacement is required, enabling straightforward updates to existing systems through firmware reprogramming. The computational overhead of filtering and normalization algorithms is negligible on modern microcontrollers, introducing no appreciable processing delay or power consumption increase. This cost-effectiveness makes the optimization framework particularly suitable for resource-constrained assistive technology applications in developing regions.

Comparison with related work reveals that the achieved recognition accuracy (94.8%) approaches vision-based systems while maintaining lower computational requirements and power consumption. Machine learning approaches employing convolutional neural networks report accuracies of 91-99% but require significant processing resources and continuous camera operation (Dewinta & Hadiana, 2015; Ferreira, Alves, & Costa, 2017). The flex sensor approach offers a favorable trade-off between accuracy, computational efficiency, and system portability for applications where power consumption and processing resources are constrained.

Several limitations warrant consideration. The system currently recognizes 31 static gestures; expansion to dynamic gestures and continuous sign language would require temporal pattern recognition algorithms beyond the current threshold-based approach. Environmental factors such as temperature variations can affect flex sensor resistance characteristics, potentially requiring periodic recalibration. The system assumes users can perform the calibration procedure, which may present challenges for individuals with severe motor impairments. Additionally, the current implementation does not address facial expressions or body language components essential for complete sign language communication.

5.0 CONCLUSION

This study successfully demonstrated systematic signal conditioning and performance optimization of a flex sensor-based sign language recognition system. Implementation of multi-stage digital filtering, hysteresis-based threshold control, and user-specific normalization yielded substantial improvements: signal fluctuation reduced from ±18 to ±3 ADC units (83% improvement), response latency decreased from 480 to 360 ms (25% improvement), threshold chattering eliminated, and recognition accuracy increased from 86.4% to 94.8%. These enhancements were achieved entirely through firmware modifications without hardware changes or increased system cost.

The optimization framework demonstrates that careful application of established signal processing and control systems principles can substantially enhance sensor-based assistive technology performance. The mathematical modeling of sensor behavior, systematic noise characterization, and principled algorithm design approach provides a reproducible methodology applicable to diverse sensor-based recognition systems beyond sign language applications.

The improved system offers a cost-effective, portable, and user-adaptable platform for sign language recognition. With recognition accuracy approaching vision-based systems while maintaining lower computational requirements and power consumption, the optimized flex sensor approach represents a viable solution for resource-constrained environments and applications requiring wearable form factors.

Future research directions include expansion to dynamic gesture recognition through temporal pattern analysis, integration of accelerometer and gyroscope data for spatial orientation tracking, implementation of machine learning classification algorithms for complex gesture patterns, development of adaptive filtering with automatic parameter optimization, and exploration of multi-modal fusion combining flex sensors with other sensing modalities. Additionally, longitudinal user studies evaluating long-term system performance and user satisfaction would provide valuable insights for practical deployment scenarios.

All Rights Reserved © 2026. https://irjecs.com

REFERENCES

Dewinta, A. H., & Hadiana, Y. (2015). American Sign Language-based finger-spelling recognition using k-nearest neighbours classifier. In *The 3rd International Conference on Information and Communication Technology*. Bali, Indonesia.

Doebelin, E. O., & Manik, D. N. (2011). *Measurement systems: Application and design* (6th ed.). McGraw-Hill.

Ezeagwu, C. O., Eze, C. E., Nwachukwu, M. M., Nnorom, O. D., & Nwebonyi, H. A. (2024). Implementing sign language recognition system using flex sensors. *International Journal of Electronics and Computer Engineering Research*. Nnamdi Azikiwe University.

Ferreira, P. A., Alves, C. J., & Costa, R. A. (2017). Multimodal learning for sign language recognition. In *Iberian Conference on Pattern Recognition and Image Analysis*. Springer.

Franklin, G. F., Powell, J. D., & Emami-Naeini, A. (2015). *Feedback control of dynamic systems* (7th ed.). Pearson.

Kaluti, M., Chaitra, M. A., Mamatha, M. M., & Swathi, M. A. (2019). Convolutional neural network for detection of sign language. *International Journal of Computer Trends and Technology (IJCTT), 67*(5), 34-37. https://doi.org/10.14445/22312803/IJCTT-V67I5P106

Munib, Q., Habeebullah, M., Takruri, B., & Al-Malik, H. A. (2007). American sign language (ASL) recognition based on Hough transform and neural networks. *Expert Systems with Applications, 32*(1), 24-37. https://doi.org/10.1016/j.eswa.2005.11.018

Ogata, K. (2010). *Modern control engineering* (5th ed.). Prentice Hall.

Oyedotun, O. K., & Khashman, A. (2017). Deep learning in vision-based static hand gesture recognition. *Neural Computing and Applications, 28*(12), 3941-3951. https://doi.org/10.1007/s00521-016-2294-8

Oz, C., & Leu, M. C. (2007). Linguistic properties based on American Sign Language isolated word recognition with artificial neural networks using a sensory glove and motion tracker. *Neurocomputing, 70*(16-18), 2891-2901. https://doi.org/10.1016/j.neucom.2006.06.004

Pressman, R. S., & Maxim, B. R. (2020). *Software engineering: A practitioner's approach* (9th ed.). McGraw-Hill Education.

Saggio, G., Riillo, F., Sbernini, L., & Quitadamo, L. R. (2016). Resistive flex sensors: A survey. *Smart Materials and Structures, 25*(1), 013001. https://doi.org/10.1088/0964-1726/25/1/013001

Sommerville, I. (2016). *Software engineering* (10th ed.). Pearson.

Zamani, H., & Kazemi, R. M. (2014). Saliency based alphabet and numbers of American sign language recognition using linear feature extraction. In *2014 4th International eConference on Computer and Knowledge Engineering (ICCKE)* (pp. 413-418). IEEE. https://doi.org/10.1109/ICCKE.2014.6993427

All Rights Reserved © 2026. https://irjecs.com