Int. J. Sci. R. Tech., 2026 3(11)
A Multidisciplinary peer-reviewed Journal
www.ijsrtjournal.com [ISSN: 2394-7063]

Original Article

Smart Interactive Wearable Posture Correction System Using Raspberry Pi Pico W and Cloud Analytics

Nikhil Barade*, Tanmay Chabukswar, Jay Mirge

Department of Electronics and Telecommunication Engineering, Dhole Patil College of Engineering, Pune, Maharashtra, India.

ABSTRACT

The proliferation of sedentary occupations and the ubiquitous use of computational devices have precipitated a global surge in posture-related musculoskeletal disorders. This paper presents the design of an IoTenabled wearable posture correction system utilizing a Raspberry Pi Pico W microcontroller. The proposed arrangement comprises an MPU6050 Inertial Measurement Unit (IMU) to track cervical inclination, a resistive flex sensor for thoracic curvature measurement, an SX1308 boost converter for power regulation, and a vibration motor for haptic biofeedback. To prevent false-positive alerts during transient daily movements, the edge firmware employs a 5-second hysteresis algorithm. Beyond localized tactile actuation, the system leverages the MQTT protocol to stream kinematic telemetry over Wi-Fi to a cloud broker and MySQL database. Users interact with the system via "Posture App," a Python-based desktop application offering real-time dual-axis graph visualization and personalized threshold calibration. The project report frames the contribution as an integrated prototype design and control architecture for long-term ergonomic tracking.

Keywords: Posture correction, Raspberry Pi Pico W, MPU6050, Wearable sensors, MQTT, IoT, Haptic biofeedback

INTRODUCTION

Maintaining optimal posture is critical for evenly distributing gravitational forces across the musculoskeletal system. However, the ubiquitous use of computers and prolonged desk-bound occupations naturally encourages individuals to adopt forward-leaning, slouched positions. Traditional methods of posture correction and assessment heavily rely on manual goniometry or periodic physiotherapy sessions, which are limited because posture is a subconscious, involuntary habit.

1.1 *Problem and motivation*

A wearable posture monitoring system must do more than simply read sensor angles. It must overcome the limitations of stationary clinical assessments by providing immediate haptic biofeedback to the user during sustained slouching, while simultaneously transmitting kinematic telemetry to a cloud database. Triggering alerts for every minor movement causes alarm fatigue; therefore, filtering transient noise and differentiating between harmless motions and sustained poor posture is a significant design challenge.

1.2 *Aim and contribution*

This work focuses on a design for automatic spinal misalignment detection and intelligent haptic actuation using a Raspberry Pi Pico W. The IMU and flex sensor supply real-time kinematic signals; the Pico W processes these signals and executes a hysteresis algorithm to command a transistor-switched vibration motor. The design also incorporates MQTT cloud telemetry, a MySQL database, and the Posture App dashboard. The intended contribution is an integrated control concept for an untethered biomedical demonstrator.

1.3 *Related Work*

Systems frequently deploy 6-axis sensors, such as the MPU6050, to measure spatial orientation. Studies demonstrate that placing an IMU at the cervical spine provides highly accurate measurements of the Craniovertebral Angle. Advanced systems incorporate resistive flex sensors along the thoracic and lumbar junctions to map structural bending.

1.4 *Positioning of the proposed design*

The project is not presented as a new biomechanical sensing principle. Instead, it applies a low-cost embedded controller (Raspberry Pi Pico W) to overcome

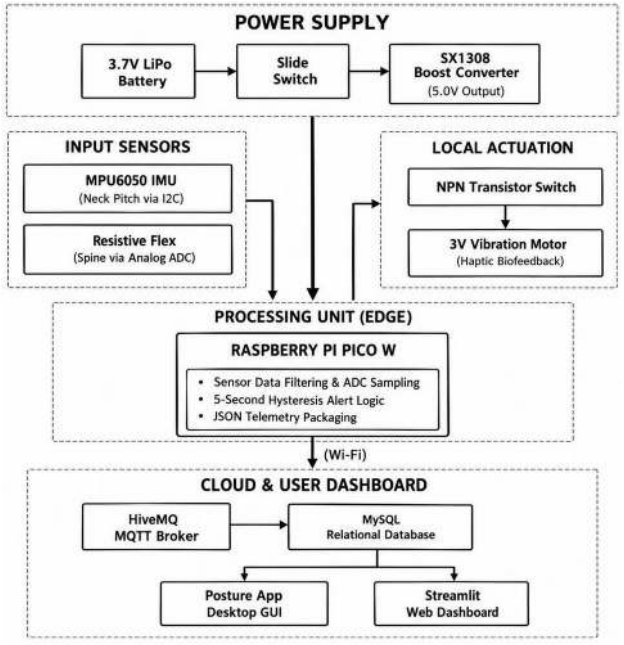
Relevant conflicts of interest/financial disclosures: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

single-sensor inaccuracies via a dual-sensor array, and implements temporal hysteresis logic to eliminate false-positive alarms. Compared to existing commercial prototypes lacking cloud connectivity, this design integrates IoT telemetry to allow historical ergonomic tracking via a dedicated Posture App dashboard.

2. System Architecture

The proposed system is divided into a sensing stage, an edge-processing stage, a power-management stage, and a cloud-telemetry stage.

Block Diagram:



2.1 *Microcontroller and Sensing Subsystems*

The physical layer is driven by the Raspberry Pi Pico W development board. Spatial sensing is achieved using the MPU6050, strategically positioned over the C7 vertebra. By actively reading the accelerometer's vectors over the I2C bus, the firmware calculates the absolute sagittal inclination of the user's neck. A 2.2inch resistive flex sensor is deployed along the thoracic junction in a voltage divider configuration to measure the structural bending radius.

2.2 *Power Management and Actuation*

The untethered prototype is powered by a 3.7V Lithium-Polymer (LiPo) battery. The SX1308 DC-DC stepup boost converter module steps up the raw battery voltage to a stable 5.0V for the Pico W's VBUS pin. The haptic biofeedback mechanism relies on a 3V Eccentric Rotating Mass (ERM) coin motor. To protect the microcontroller, the motor is driven by a low-side switching circuit utilizing a BC547 NPN transistor and a flyback diode.

2.3 *Cloud Telemetry and User Interface*

The Raspberry Pi Pico W utilizes the MQTT protocol to connect to a public HiveMQ cloud broker. Serialized JSON payloads containing posture data are published and permanently stored in a MySQL relational database.

The "Posture App" desktop application subscribes to these topics to render synchronous dual-axis graphs.

3. Proposed Control Method

The control sequence is based on temporal hysteresis. The edge firmware actively filters out motion noise by tracking the duration of postural deviation.

3.1 *Control sequence*

1) Initialize the Raspberry Pi Pico W, I2C bus, ADC pins, and MQTT client. 2) Keep the haptic motor disabled by default. 3) Read the MPU6050 acceleration vectors and flex sensor analog voltage. 4) Calculate cervical pitch and structural bending. 5) If values are within neutral thresholds, remain in standby and publish "Neutral". 6) If thresholds are breached, start a timer. 7) If the breach is sustained continuously for 5 seconds, enable the transistor to drive the haptic motor and publish "Slouching Warning". 8) Return to standby once the user corrects their alignment.

3.2 *Implementation details requiring specification*

The microcontroller is programmed using MicroPython. The threshold values (e.g., 25 degrees for pitch, 2300 ADC for flex) are empirically calibrated but can be adjusted dynamically via the Posture App. The transistor isolates the Pico W's GPIO from the heavy 40mA+ current draw of the motor.

4. Prototype Design and Specifications

Table 1: Reported Design Specifications

Parameter	Specification stated in project report
Controller	Raspberry Pi Pico W development board
Primary supply	3.7V Lithium-Polymer (LiPo) Battery
Boost Converter	SX1308 Module (regulated to 5.0V output)
Spatial Sensor	MPU6050 (I2C protocol)
Structural Sensor	2.2-inch Resistive Flex Sensor (ADC voltage divider)

Parameter	Specification stated in project report
Actuation	3V ERM vibration motor with BC547 NPN transistor
Cloud Protocol	MQTT via HiveMQ broker
User Interface	Posture App (Python/CustomTkinter) and MySQL DB

5. Evaluation Methodology and Results Status

The physical bench testing and simulation evaluate sensing accuracy, hysteresis reliability, and telemetry latency.

5.1 *Test plan*

Test	Procedure	Data to record
T1: Sensor Baseline	Measure ADC and pitch in neutral sitting posture.	Baseline voltage, baseline IMU pitch angle.
T2: Transient Motion	User performs quick reaching/bending tasks (<5s).	Motor state (should remain off), hysteresis timer logic.
T3: Sustained Slouch	User holds poor posture for >5 seconds.	Motor actuation response, calculated pitch/flex values.
T4: Telemetry Latency	Observe MQTT publish to Dashboard render time.	Network latency in milliseconds, payload integrity.
T5: Calibration Override	Adjust threshold in Posture App during operation.	Firmware threshold update success.

5.2 *Findings supported by the report*

The integration of the 5-second temporal hysteresis algorithm performed exactly as engineered; transient physical movements did not trigger haptic feedback, effectively eliminating false-positive alarms. The serialized JSON payloads were successfully intercepted by the Python backend and accurately logged into the MySQL relational database.

5.3 *Performance metrics*

Postural compliance can be estimated through the Posture Health Score metric within the Posture App, defined as the ratio of "Neutral" telemetry points to total telemetry points recorded during a session.

DISCUSSION

The main design value is the integration of dual-sensor kinematic profiling with an edge-computed hysteresis algorithm to eliminate alarm fatigue. By transitioning data to a MySQL cloud backend, the system allows for personalized biometric calibration rather than enforcing a rigid threshold.

6.1 *Limitations*

Because the prototype lacks an offline SD-card module, it relies entirely on a continuous 2.4 GHz Wi-Fi connection; telemetry logging halts if the network drops. Furthermore, the SX1308 boost converter exhibits an inherent idle current draw that impacts long-term standby battery efficiency, and resistive flex sensors are susceptible to mechanical fatigue over prolonged use.

7. Conclusion and Future Work

This paper described a Raspberry Pi Pico W-based design for a smart wearable posture correction system. The architecture combines an MPU6050, a flex sensor, hysteresis-controlled haptic feedback, and MQTT cloud telemetry. The central design objective was to filter out transient motion noise while accurately logging ergonomic data via the Posture App. Future work includes transitioning to a custom surface-mount printed circuit board (PCB) to reduce dimensions, implementing offline data logging via the Pico W's onboard flash, and applying embedded Machine Learning (TinyML) for dynamic activity recognition.

Acknowledgment

The author acknowledges the Department of Electronics and Telecommunication Engineering, Dhole Patil College of Engineering, Pune, for the academic foundation supporting this work.

REFERENCES

1. A. Wong et al., "Real-time Posture Monitoring Systems Using Wearable Sensors: A Review," IEEE Sensors Journal,vol. 21, pp. 10564-10576, 2021.
2. J. Hans et al., "Effectiveness of Haptic Biofeedback in Correcting Spinal Posture," IEEE Journal

of Biomedical andHealth Informatics, vol. 24, no. 5, pp. 1320-1328, 2020.

3. Raspberry Pi Foundation, Raspberry Pi Pico W Official Documentation. Available: https://www.raspberrypi.com/documentation/microcontrollers/
4. InvenSense Inc., MPU-6000 and MPU-6050 Product Specification Revision 3.4. Available: https://invensense.tdk.com/
5. Spectra Symbol, Flex Sensor FS-L-0055 Datasheet. Available: https://www.spectrasymbol.com/
6. Matt's Electronics, SX1308 Boost Converter Technical Specifications and Performance Data.
7. MicroPython Official Documentation. Available: https://docs.micropython.org/
8. OASIS Standard, MQTT Version 3.1.1 Documentation. Available: https://mqtt.org/
9. Oracle Corporation, MySQL 8.0 Reference Manual. Available: https://dev.mysql.com/doc/
10. Python Software Foundation, Python 3.11 Official Documentation. Available: https://docs.python.org/3/ S.No Name of Author Designation Department 1 Nikhil Nanasaheb Barade Student E&TC Engineering 2 Tanmay Balaji Chabukswar Student E&TC Engineering 3 Jay Madhukar Mirge Student E&TC Engineering 4 Prof.Neha Dumne Head of Department E&TC Engineering 5 Prof.Dr.Bhausaheb Eknath Shinde Dean Research & Development, Assistant Professor E&TC Engineering

HOW TO CITE: Nikhil Barade, Tanmay Chabukswar, Jay Mirge, Smart Interactive Wearable Posture Correction System Using Raspberry Pi Pico W and Cloud Analytics, Int. J. Sci. R. Tech., 2026, 3 (11), 458-461. https://doi.org/10.5281/zenodo.24030117