International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 5, Sep-Oct-2026, ISSN (Online): 2395-566X

Universal Rep Counter

M. Pavani[1], K. Shreya[2], Kb. Chetana Reddy[3], M. Deepika[4], M. Ranjith Kumar[5]
Mranjithkumar5425@Gmail.Com
Hyderabad Institute Of Technology And Management

ABSTRACT- Manual repetition counting during strength training is prone to human error, and conventional motion-based rep counters cannot verify whether a user has actually completed the required Range of Motion (ROM) for an exercise. This often results in the counting of partial or incomplete repetitions, damaging the reliability of workout data. This paper proposes a Universal Rep Counter with ROM Detection, an embedded system built around an MPU6050 inertial measurement unit (IMU) and an ESP32 microcontroller. The system fuses accelerometer and gyroscope data using a complementary filter to estimate the joint or limb angle in real time, and validates each repetition against a configurable minimum-maximum ROM threshold using a finite-state counting algorithm. Only movements that traverse the full required range and return to the starting position are registered as valid repetitions. The design incorporates an OLED/LCD display for real-time feedback and push-button controls for exercise selection, calibration, and reset, making the system adaptable to a wide range of resistance exercises including bicep curls, squats, bench presses, shoulder presses, rows, and lat pull-downs. This paper presents the system architecture, sensor-fusion methodology, rep-validation algorithm, and expected behavior based on simulated sensor data, and outlines a roadmap for hardware validation and future extensions such as automatic exercise recognition and AI-based form analysis.

Keywords-- IMU, MPU6050, ESP32, Range of Motion, Repetition Counting, Complementary Filter, Wearable Fitness Technology, Embedded Systems.

I. INTRODUCTION

Strength training is a core component of physical fitness, rehabilitation, and athletic performance programs. A fundamental metric in any such program is the repetition (rep) count, which is used to track training volume, monitor progress, and structure workout intensity. Traditionally, this count is maintained manually by the exerciser or a trainer, a method that is inherently susceptible to distraction, fatigue-induced miscounting, and simple human error, particularly during high-intensity or high-repetition sets.

Several commercial wearable devices and smartphone applications attempt to automate rep counting using onboard accelerometers. However, most of these systems rely purely on motion or peak detection: any sufficiently large acceleration spike is registered as a repetition, regardless of whether the user actually moved through the biomechanically correct range of motion. This creates a critical gap between quantity and quality of repetitions: a half-completed curl or a partial squat can be counted identically to a full, correctly executed one, which not only inflates the recorded training volume but can also mask poor exercise form that may lead to injury over time.

This project addresses that gap by introducing Range of Motion (ROM) validation as a first-class requirement for repetition counting, rather than treating rep counting as a pure motion-detection problem. The proposed system, the Universal Rep Counter with ROM Detection, uses an IMU sensor (MPU6050) interfaced with an ESP32 microcontroller to continuously track the angular displacement of the exercising limb or equipment. A repetition is registered only when the tracked angle crosses a pre-defined minimum threshold, proceeds to a pre-defined maximum threshold, and returns to the starting position: completing a full valid movement cycle. The system is designed to be exercise-agnostic within the domain of resistance training, supporting exercises such as bicep curls, squats, bench presses, shoulder presses, leg presses, leg extensions, rows, and lat pull-downs, with the ROM thresholds reconfigurable per exercise through push-button controls.

The remainder of this paper is organized as follows: Section II reviews related work in IMU-based activity and repetition recognition. Section III describes the overall system architecture. Section IV details the ROM detection and sensor-fusion methodology. Section V presents the repetition-counting algorithm as a finite-state process. Section VI discusses expected system behavior based on simulated sensor data, given that the project is currently at the design stage. Section VII outlines advantages and limitations, Section VIII discusses applications, Section IX presents future scope, and Section X concludes the paper.

II. LITERATURE REVIEW

A substantial body of work has explored the use of wearable inertial sensors for human activity and exercise recognition. Early consumer-facing systems, such as step counters, use accelerometer signal patterns to detect periodic motion; for instance, Patil et al. describe a step-counting system built on the ESP32 and MPU6050 combination, demonstrating the

© 2026 IJSRET

suitability of this hardware pairing for real-time motion sensing and IoT-connected fitness tracking.

Beyond simple step counting, research into automated exercise repetition counting has generally followed a few distinct methodological directions. One widely used approach is peak detection, in which the raw or pre-processed IMU signal is treated as a 1D waveform and each significant peak is counted as a repetition. As documented in a review of on-device repetition-counting systems, this approach is simple to implement but is highly sensitive to variations in a user's exercise form, often generating multiple spurious peaks per repetition or missing weak ones, which leads to over- or under-counting; amplitude thresholding on a per-exercise basis has been shown to substantially improve the reliability of this method, with reported accuracy as high as 99.4% under controlled conditions.

Other approaches reported in the literature include Hidden-Markov-Model-based sequence decoding, neural-network classifiers that output a binary repetition/non-repetition signal, and Dynamic Time Warping to compare an observed movement pattern against a reference templsate, with accuracies ranging widely depending on exercise variability and sensor placement .

At the system level, RecoFit, an arm-worn IMU-based system for tracking repetitive free-weight and callisthenic exercises, demonstrated that exercise segmentation, recognition, and counting could be performed jointly, achieving over 95% segmentation precision/recall and counting accuracy within one repetition roughly 93% of the time, although its recognition pipeline required a multi-second window to identify the exercise being performed.

More recent work has moved toward exercise-agnostic, few-shot learning methods that aim to count repetitions for previously unseen exercise types using a single trained model, reducing the need for exercise-specific calibration. While these data-driven approaches are powerful, they typically require substantial training data, cloud or edge inference capability, and are not always well suited to low-cost, standalone embedded deployments.

A common thread across this literature is that the majority of IMU-based counters are optimized for detecting that a repetition-like motion occurred, rather than verifying that the motion satisfied a specific biomechanical range of motion. The proposed system differs from these approaches by treating ROM compliance as the primary gating condition for a valid count, using a lightweight, deterministic threshold-and-state-machine approach that can run entirely on a low-cost microcontroller without requiring a trained model, cloud connectivity, or exercise-specific datasets: while still remaining configurable across the many resistance exercises identified in Section I.

III. PROPOSED SYSTEM ARCHITECTURE

The proposed system consists of five functional blocks:

- The Imu Sensing Unit
- The Microcontroller Processing Unit
- The User Interface (Display and Buttons)
- 4. The ROM detection and rep-validation logic and
- 5. The power supply unit.

1. Imu Sensing Unit
The MPU6050 is a 6-axis IMU integrating a 3-axis MEMS accelerometer and a 3-axis MEMS gyroscope on a single chip, communicating with the host microcontroller over the I²C protocol. Depending on the exercise, the sensor module is mounted either directly on the exercising limb (e.g., forearm for bicep curls, thigh for squats) using a strap, or on the moving component of the exercise equipment (e.g., the lever arm of a leg press or the cable attachment of a lat pull-down machine), so that the sensor's angular displacement directly reflects the limb or equipment's range of motion.

2. Microcontroller Processing Unit
An ESP32 microcontroller was selected as the processing core due to its dual-core processor, sufficient clock speed for real-time filtering computations, native I²C support, low cost, and built-in Wi-Fi/Bluetooth connectivity, which supports the future mobile-connectivity extension discussed in Section IX. The ESP32 reads raw accelerometer and gyroscope values from the MPU6050 at a fixed sampling interval, applies the sensor-fusion algorithm described in Section IV, and executes the rep-validation state machine described in Section V.

3. User Interface
An OLED or character LCD display provides real-time feedback, including the current repetition count, the selected exercise, and the set/rest status. Four push buttons allow the user to:

- cycle through the list of supported exercises, each with pre-set default ROM thresholds;
- start or pause the active set;
- reset the repetition counter; and
- enter a calibration mode to manually adjust the minimum and maximum ROM thresholds for the selected exercise, accommodating users with different body proportions or flexibility.

4. Rom Detection And Rep-Validation Logic
This is the core contribution of the system. Rather than counting every detected motion event, the firmware

© 2026 IJSRET

continuously compares the fused angle estimate against exercise-specific threshold values and only increments the counter when a complete, valid movement cycle has occurred, as detailed in Sections IV and V.

5. Power Supply

The system is designed to run from a rechargeable lithium-polymer battery regulated to the 3.3V logic level required by the ESP32 and MPU6050, making the unit fully portable and suitable for both body-worn and equipment-mounted configurations.

IV. ROM DETECTION METHODOLOGY

Accurate ROM detection depends on obtaining a stable, drift-free estimate of the joint or equipment angle from noisy sensor data. Using the accelerometer alone provides an absolute tilt reference (derived from the direction of gravity) but is highly sensitive to vibration and dynamic acceleration during movement. Using the gyroscope alone provides a smooth, high-bandwidth angular rate signal, but integrating this rate over time to obtain an angle accumulates drift error. To address this, the system uses a complementary filter to fuse both signals, a standard and computationally inexpensive technique for MPU6050-based orientation estimation.

1. Accelerometer-Based Tilt Angle

From the raw accelerometer readings (ax, ay, az), the tilt angle about the axis of interest is computed using the geometric relationship between the gravity vector components, for example:

$$\text{angle_acc} = \text{atan2}(ay, \sqrt{(ax^2 + az^2)}) \times (180 / \pi)$$

This estimate is accurate under static or slow-moving conditions but becomes noisy during rapid limb acceleration, such as at the start or end of a repetition.

2. Gyroscope-Based Angle Integration

The gyroscope provides an angular rate (°/s) about the relevant axis, which is integrated over each sampling interval (dt) to update the angle estimate:

$$\text{angle_gyro}(t) = \text{angle}(t-1) + \text{gyro_rate} \times dt$$

This estimate is smooth and responsive but drifts over longer periods due to sensor bias and integration error.

C. Complementary Filter Fusion

The two estimates are fused using a weighted combination that favors the gyroscope for short-term responsiveness while periodically correcting drift using the accelerometer's long-term reference:

$$\text{angle}(t) = \alpha \times (\text{angle}(t-1) + \text{gyro_rate} \times dt) + (1 - \alpha) \times \text{angle_acc}$$

A typical value of $\alpha = 0.98$ is used, giving 98% weight to the gyroscope-integrated angle and 2% weight to the accelerometer-derived angle per update cycle, a configuration commonly reported in MPU6050-based orientation-tracking implementations. Prior to use, the system performs a brief calibration routine while the user is stationary, during which gyroscope bias is sampled and subtracted from subsequent readings.

3. Rom Threshold Definition

For each supported exercise, two threshold angles are defined: theta_min, representing the starting/rest position of the movement, and theta_max, representing the fully contracted or extended position that constitutes correct completion of the exercise. Table I lists representative (illustrative, adjustable) threshold ranges for a subset of supported exercises.

TABLE I. Representative ROM threshold ranges by exercise (values are configurable via calibration mode).

Exercise	θ_min (rest, °)	θ_max (target, °)	Sensor Mount Location
Bicep Curl	≈ 10–20	≈ 130–150	Forearm (wrist strap)
Squat	≈ 0–10	≈ 80–100	Thigh
Shoulder Press	≈ 0–10	≈ 150–170	Upper arm
Bench Press	≈ 0–10	≈ 70–90	Forearm / bar attachment
Lat Pull-Down	≈ 0–10	≈ 90–110	Cable attachment / forearm

These default values are stored in the firmware and can be fine-tuned by the user through the calibration button, accommodating individual anthropometric differences and equipment variations.

4. Repetition-Counting Algorithm

Repetition validity is determined using a finite-state machine (FSM) that tracks the progression of the fused angle relative to the configured thresholds. The FSM has four states: IDLE, ASCENDING, PEAK_REACHED, and DESCENDING. A repetition is counted only upon a complete IDLE → ASCENDING → PEAK_REACHED → DESCENDING → IDLE transition cycle.

5. Algorithm 1: Rom-Validated Rep Counting

- Initialize state = IDLE; rep_count = 0.

© 2026 IJSRET

- Calibrate gyroscope bias while the sensor is stationary.
- On each sampling interval, read (ax, ay, az, gx, gy, gz) and compute the fused angle using the complementary filter (Section IV-C).

- If state = IDLE and angle ≥ θ_min + hysteresis, set state = ASCENDING.
- If state = ASCENDING and angle ≥ θ_max − hysteresis, set state = PEAK_REACHED.
- If state = PEAK_REACHED and angle ≤ θ_max − hysteresis, set state = DESCENDING.
- If state = DESCENDING and angle ≤ θ_min + hysteresis, increment rep_count by 1, set state = IDLE, and update the display.
- If, from ASCENDING, the angle falls back toward θ_min without ever reaching θ_max (i.e., an incomplete movement), reset state = IDLE without incrementing rep_count.

1. Repeat from Step 3 until the user pauses or ends the set via the control buttons.

The hysteresis margin in Steps 4–7 prevents sensor noise near a threshold boundary from causing rapid, spurious state transitions or double counting. Step 8 is the mechanism that specifically rejects partial repetitions — movements that begin but do not reach the required maximum ROM are discarded rather than counted, directly addressing the core limitation of conventional motion-only rep counters identified in Section I.

V. EXPECTED BEHAVIOUR AND SIMULATION-BASED DISCUSSION

As this project is currently at the design and simulation stage, no physical hardware trial data is presented in this paper. Instead, this section describes the expected behaviour of the system based on the algorithm's design logic and on simulated sensor traces, and outlines the validation methodology intended for the hardware implementation phase.

For a correctly executed repetition, the fused angle output is expected to follow an approximately smooth, single-peaked trajectory over time - rising from near θ_min, crossing θ_max, and returning to θ_min - causing the FSM to traverse all four states exactly once and increment the counter by one. For an incomplete repetition, where the user does not reach the required extension or contraction, the simulated angle trace rises from θ_min but reverses direction before reaching θ_max; under Algorithm 1, this correctly results in no increment, whereas a conventional peak- or motion-threshold counter operating on raw acceleration magnitude alone would likely register this as a valid repetition, since a motion event still occurred.

Prior to physical prototyping, the design is intended to be validated in two stages: first, through firmware-level simulation using an ESP32/MPU6050 emulator (such as the Wokwi simulation platform) with synthetically generated or pre-recorded IMU waveforms representing correct, partial, and noisy repetitions, to verify that the FSM and complementary filter produce the expected state transitions and rep counts; and second, through bench-top testing with the physical sensor mounted on a mechanical fixture moved through known, controlled angles, to quantify real-world measurement accuracy, hysteresis tuning requirements, and drift behaviour over an extended set duration. This staged validation approach allows the counting algorithm's logical correctness to be verified independently of hardware-specific noise characteristics before full physical deployment.

Based on the algorithm's design and the accuracy figures reported for comparable threshold-based IMU repetition-counting methods in the literature (up to 99.4% for well-tuned amplitude thresholding), the proposed ROM-gated approach is expected to achieve high counting accuracy for controlled, single-joint exercises, with the primary anticipated sources of error being sensor mounting slippage, incorrect per-user threshold calibration, and drift during long sets, all of which are addressed through the calibration workflow and periodic accelerometer-based correction described in Section IV.

VI. ADVANTAGES AND LIMITATIONS

1. Advantages

- Rejects partial or incomplete repetitions, improving the accuracy and training-relevance of the recorded rep count compared to motion-only counters.
- Low-cost, portable, and battery-powered, using widely available and inexpensive components (MPU6050, ESP32).
- Configurable per-exercise ROM thresholds support a wide range of resistance exercises without redesigning the hardware.
- Does not require an internet connection, cloud inference, or a trained machine-learning model to operate, unlike many recognition-based approaches in the literature.
- Modular design allows the sensor to be body-worn or equipment-mounted depending on the exercise.

2. Limitations

- Requires manual selection of the exercise type; it does not yet automatically recognize which exercise is being performed.

© 2026 IJSRET

- Threshold-based detection may need per-user calibration to account for differences in limb length and flexibility.
- As a single-IMU system, it captures ROM along the primary axis of motion and may not fully capture compound or multi-planar movements without additional sensors.
- The current design has not yet been validated on physical hardware; reported behaviour is based on algorithmic design and simulation.

3. Applications

The proposed system has applications across several domains. In personal fitness and home gyms, it offers an affordable way to track accurate, quality-controlled repetitions without a personal trainer. In physiotherapy and rehabilitation, ROM-gated counting can help ensure patients complete prescribed exercises through their full clinically required range of motion, supporting recovery tracking. In sports training and coaching, coaches can use the objective repetition and ROM-compliance data to monitor athlete technique consistency across a training block. In gym and fitness-equipment integration, the sensor module can be embedded directly into machines such as leg presses or lat pull-down stations to provide built-in, equipment-level rep tracking for multiple users.

4. Future Scope

- Automatic exercise recognition using onboard classification, removing the need for manual exercise selection.
- Workout history logging with persistent storage (e.g., SD card or flash) and session summaries.
- Repetition-speed and tempo measurement to analyze concentric/eccentric phase timing.
- Posture and form monitoring using additional IMUs or barometric/pressure sensors for multi-point tracking.
- Mobile app connectivity via the ESP32's built-in Bluetooth/Wi-Fi for real-time data visualization and long-term progress tracking.

6. AI based fitness analysis, using accumulated sensor data to provide personalized feedback on form quality and training progression.

VII. CONCLUSION

This paper presented the design of a Universal Rep Counter with Range of Motion Detection, an embedded system that combines an MPU6050 IMU, an ESP32 microcontroller, a complementary-filter-based sensor-fusion algorithm, and a finite-state repetition-validation scheme to address a specific shortcoming of conventional rep counters: their inability to distinguish between complete and partial repetitions. By requiring that a tracked limb or equipment angle traverse a configurable minimum-to-maximum range and return to its starting position before a repetition is counted, the system is designed to provide more meaningful and reliable workout data than motion-detection-only alternatives, while remaining low-cost, portable, and adaptable across a wide range of strength-training exercises. As the project currently stands at the design and simulation stage, the immediate next step is hardware prototyping and empirical validation of the counting accuracy described conceptually in this paper, followed by the broader set of enhancements - automatic exercise recognition, workout history, tempo analysis, posture monitoring, mobile connectivity, and AI-based analysis - outlined in Section IX.

REFERENCES

1. M. Patil, M. Zala, and N. Mehendale, “Step Counter Using ESP-32,” SSRN Electronic Journal, Nov. 2024. [Online]. Available: https://ssrn.com/abstract=5011341
2. Random Nerd Tutorials, “ESP32 MPU-6050 Accelerometer and Gyroscope (Arduino),” randomnerdtutorials.com. [Online]. Available: https://randomnerdtutorials.com/esp32-mpu-6050-accelerometer-gyroscope-arduino/
3. Author(s) of “Pūoio: On-device Real-Time Smartphone-Based Automated Exercise Repetition Counting System,” arXiv:2308.02420. [Online]. Available: https://arxiv.org/pdf/2308.02420
4. “ExerSense: Real-Time Physical Exercise Segmentation, Classification, and Counting Algorithm Using an IMU Sensor,” arXiv:2004.10026. [Online]. Available: https://arxiv.org/pdf/2004.10026
5. Y. Lim and S. Lee, “Metric-Based Few-Shot Learning for Exercise Repetition Counting with IMU Data,” arXiv:2410.00407, Oct. 2024. [Online]. Available: https://arxiv.org/html/2410.00407v1
6. Tutorialspoint, “Interfacing ESP32 with MPU6050,” tutorialspoint.com. [Online]. Available: https://www.tutorialspoint.com/esp32_for_iot/interfacing_esp32_with_mpu6050.htm
7. InvenSense Inc., “MPU-6000/MPU-6050 Product Specification,” Datasheet, InvenSense, Sunnyvale, CA.
8. Espressif Systems, “ESP32 Series Datasheet,” Espressif Systems, Shanghai, China.

© 2026 IJSRET