

Experimental Performance Evaluation of 6-Axis MEMS Inertial Sensor for CubeSat Attitude Determination Applications

U. A. Sadiq, U. Yuguda, R. N. Abali, I. Ajigbon, C. F. Ahanonu
National Space Research & Development Agency (NASRDA), Nigeria

Abstract— This paper presents experimental performance evaluation of a low-cost 6-axis Micro-Electro-Mechanical Systems (MEMS) inertial measurement unit (IMU) for CubeSat attitude determination applications. The study employs a practical implementation of pitch, roll, and yaw (PRY) extraction using a low-cost MPU-6050 MEMS inertial measurement unit interfaced with an ATmega328 microcontroller and a 20 characters by 4 lines liquid crystal display (LCD) for real-time visualization. A complementary filter was implemented to fuse accelerometer and gyroscope data, achieving stable and responsive orientation estimation. Experimental evaluation under static and dynamic conditions demonstrates sub-degree accuracy for pitch and roll, while yaw estimation exhibited minor drift due to the absence of magnetometer correction. The calculated RMSE for static roll estimation was approximately 0.61° . The RMSE for static pitch estimation was approximately 0.74° , while the drift rate was about $1.37^\circ/\text{min}$. The results confirmed that low-cost MEMS sensors, when coupled with lightweight filtering algorithms, are viable for embedded attitude measurement in robotics, Unmanned Aerial Vehicles (UAVs), and CubeSat applications.

Keywords— MPU-6050, Arduino, MEMS IMU, Pitch, Roll, Yaw, Complementary Filter, Embedded Systems.

I. INTRODUCTION

Attitude estimation, defined as the determination of an object's orientation in three-dimensional space, is a fundamental requirement in modern robotics, unmanned aerial vehicles (UAVs), CubeSats, and autonomous systems [3], [6]. Accurate measurement of pitch, roll, and yaw is critical for navigation, stabilization, control, and trajectory planning. High-performance inertial navigation systems traditionally rely on precision gyroscopes and accelerometers; however, such systems are expensive, bulky, and require complex calibration procedures [3].

Recent advancements in micro-electromechanical systems (MEMS) have enabled the development of compact, low-cost inertial measurement units (IMUs) that integrate tri-axial accelerometers and gyroscopes on a single chip. The MPU-6050 is one of the most widely used MEMS IMUs in embedded applications [2]. It provides 3-axis linear acceleration and 3-axis angular rate measurements, enabling CubeSats to determine orientation, detect motion, and estimate velocity. Its affordability, low power consumption, and ease of integration with microcontrollers, such as Arduino, make it a popular choice for educational, research, small satellite development and prototyping projects [5], [8].

Despite these advantages, MEMS-based attitude estimation faces several challenges. Accelerometers provide absolute tilt measurements but are highly sensitive to high-frequency vibrations and transient disturbances, while gyroscopes offer smooth, short-term angular rate measurements but are prone to drift over time due to bias instability [3], [6]. The fusion of accelerometer and gyroscope data through sensor fusion algorithms, such as complementary filters or Kalman filters, is therefore essential to achieve both accuracy and stability in orientation estimation [4], [7].

Many studies focus on software-based approaches to estimate pitch, roll, and yaw from low-cost MEMS IMUs. Complementary filters, which combine the high-frequency response of gyroscopes with the low-frequency stability of accelerometers, provide a computationally lightweight and efficient solution suitable for microcontrollers with limited processing capabilities [4], [10]. More advanced sensor fusion algorithms, such as Madgwick's gradient-descent filter and Mahony's nonlinear complementary filter, offer improved performance in dynamic environments [1], [7].

However, the practical deployment of MEMS IMUs in CubeSat's attitude determination requires careful consideration of calibration, filtering, real-time processing, and visualization. Most existing studies focus solely on algorithm development and off-line data processing, with limited emphasis on real-time display or embedded human-machine interface (HMI) integration. Furthermore, while pitch and roll estimation is often accurate, yaw estimation remains challenging due to cumulative drift in the absence of absolute heading references [5], [10].

This study addresses these gaps by implementing a real-time ATmega328 microcontroller-based MPU-6050 system for attitude estimation with direct LCD display output. The system is evaluated under both static and dynamic conditions, with a focus on accuracy, drift, and real-time responsiveness. The proposed methodology provides a practical, low-cost, and scalable solution for embedded attitude measurement applications, including robotics, UAVs, CubeSats, and educational projects.

II. LITERATURE REVIEW

The development of low-cost MEMS IMU-based attitude measurement systems has been an active research area over the past decades. Several studies have evaluated the performance of the MPU-6050 and similar IMUs in embedded platforms.

Karim and Hossain [5] implemented an Arduino-based real-time roll and pitch measurement system using the MPU-6050, demonstrating sub-degree accuracy under static conditions. Lim et al. [10] reported similar results, showing that complementary filtering effectively suppresses high-frequency accelerometer noise while mitigating gyroscope drift.

Sensor fusion algorithms play a crucial role in MEMS IMU performance. Complementary filters are widely adopted due to their simplicity and low computational requirements, making them suitable for microcontrollers such as Arduino or STM32 [4], [10]. Mahony et al. [7] proposed a nonlinear complementary filter operating on the special orthogonal group (SO (3)), which provides robust and stable attitude estimation for dynamic applications. Madgwick et al. [1] introduced a gradient-descent-based orientation filter that offers real-time performance with low computational overhead, making it well-suited for embedded UAV and mobile robotics applications.

Several studies have addressed yaw estimation challenges. Since yaw is typically derived from gyroscope integration, low-cost MEMS sensors exhibit cumulative drift over time [5], [10]. Approaches to mitigate this include the integration of a 3-axis magnetometer to provide absolute heading information or the use of advanced sensor fusion algorithms such as Kalman filters [1], [6]. Groves [6] emphasized the importance of bias calibration, temperature compensation, and periodic alignment to maintain long-term orientation accuracy in embedded IMU systems.

In addition to algorithmic solutions, hardware-software co-design is critical for practical deployment. The integration of microcontrollers with real-time displays or data logging modules allows for immediate feedback and verification of sensor performance, facilitating system debugging, calibration, and educational demonstrations [8], [9]. Recent research also emphasizes dynamic validation of IMU systems under controlled rotation and vibration conditions, which provides a more realistic assessment of embedded system performance [10].

Overall, the literature demonstrates that low-cost MEMS IMUs are capable of reliable attitude estimation in embedded applications, particularly for pitch and roll. However, practical systems that combine real-time display, embedded computation, and full 3-axis orientation measurement remain limited. The present study contributes to this domain by the following; implementing a complete ATmega328-based attitude measurement platform with MPU-6050 and LCD output, evaluating static and dynamic performance under controlled experimental conditions, characterizing drift, and repeatability for practical embedded applications, and providing a low-cost, scalable methodology suitable for research, education, and prototyping.

III. MATERIALS AND METHODS

The materials and methods section describes the hardware, sensors, data acquisition setup, and experimental procedures used to evaluate the static and dynamic performance of the MPU-6050 MEMS inertial measurement unit (IMU). The study focuses on practical extraction of roll, pitch, and yaw angles using an Arduino microcontroller and real-time visualization on

an LCD. The methodology emphasizes reproducibility, accuracy, and systematic validation under controlled laboratory conditions.

Materials

This study utilizes a combination of **low-cost** MEMS sensors, microcontrollers, and display modules to develop a practical system for real-time extraction of roll, pitch, and yaw angles. The materials were selected to provide accuracy, reliability, and affordability, making them suitable for embedded applications such as UAVs, robotics, and CubeSat attitude monitoring.

The hardware components used in this study are summarized in Table 1.

TABLE 1: Materials used and their Specifications

Component	Description	Specification
MPU-6050	MEMS IMU Sensor	3-axis accelerometer, 3-axis gyroscope
Microcontroller	Microcontroller	ATmega328P, 16 MHz, 2 kB SRAM, 32 kB Flash
LCD Display	16×2 Character LCD	HD44780 controller
Connecting Wires	Jumper Cables	22 AWG, male-to-male
Breadboard	Prototyping Board	830 Tie-points
Power Supply	USB or 5V Adapter	5 V DC

Software Implementation

The software implementation was developed to enable real-time acquisition, processing, fusion, and visualization of inertial data from the MPU-6050 using the Arduino Uno (ATmega328P). The firmware was written in C/C++ using the Arduino Integrated Development Environment (IDE) and structured to ensure deterministic sampling, computational efficiency, and numerical stability under limited memory constraints.

Experimental Procedure

This is the step-by-step methods for static and dynamic testing, including reference angle alignment, data acquisition, and measurement validation.

Experimental Setup

The experimental setup consisted of the MPU-6050 module rigidly mounted on a flat acrylic platform to minimize vibration and unintended motion. The platform was manually rotated about each principal axis during testing. The Arduino Uno was connected to a personal computer for optional serial data logging, while real-time angle outputs were displayed on a 20×4 LCD. A mechanical protractor (resolution ±1°) was used as a reference instrument for static angle validation. For drift measurement, a digital stopwatch was used to record elapsed time during stationary tests. The system was allowed to warm up for approximately 2–3 minutes prior to data acquisition to reduce thermal drift effects in the MEMS sensors. The system was powered using a regulated 5 V DC supply via external adapter. A common ground reference was maintained across all modules to ensure signal integrity. Before conducting

measurements, a static calibration process was performed as follows:

1. The sensor module was placed on a level surface.
2. 1000 samples of accelerometer and gyroscope data were collected at 100 Hz.
3. Mean offset values for each axis were computed.
4. The calculated bias values were stored and subtracted from subsequent measurements.

The z-axis accelerometer output was verified to be approximately 1 g under stationary horizontal alignment to confirm proper sensor orientation.

Static validation was conducted to evaluate the accuracy of roll and pitch estimation under quasi-static conditions. In order to carry out roll test, the platform was rotated about the x-axis to predefined angles of 0°, 15°, 30°, 45°, and 60°. At each position, the system was held steady for approximately 10s. The displayed roll angle was recorded once the reading stabilized. Finally, three repeated measurements were taken for each angle to ensure repeatability. Furthermore, pitch test was performed by repeating the same procedure for rotation about the y-axis while reference angles were again verified using the mechanical protractor.

In order to assess responsiveness and stability during motion, dynamic testing was conducted. The platform was manually rotated with moderate angular velocity (approximately 20–40 °/s). Step-like angular changes were introduced (e.g., 0° to 45°). The time required for the system to stabilize at the new angle was recorded.

Experimental Limitations

The experimental setup relied on manual rotation, which may introduce minor inconsistencies in dynamic tests. Additionally, yaw estimation was limited by the absence of magnetometer correction, leading to observable long-term drift. Environmental vibration and temperature variation was also observed to influence sensor stability.

IV. RESULTS AND DISCUSSION

This section presents the experimental results obtained from static roll and pitch validation, dynamic step response testing, and yaw drift evaluation. The findings are discussed in relation to sensor accuracy, stability, responsiveness, and inherent limitations of low-cost MEMS IMUs.

Static Roll Validation

Static roll measurements were obtained by incrementally tilting the MPU-6050 platform to reference angles of 0°, 15°, 30°, 45°, and 60°. Each position was sampled at 100 Hz over 10 seconds, and the average measured angle was computed and compared to the known reference angle. The results are presented in Table 2.

TABLE 2: Static Roll Validation Results

Reference (°)	Measured (°)	Abs. Error (°)	Std. Dev. (°)
0	0.42	0.42	0.18
15	14.63	0.37	0.21
30	29.54	0.46	0.24
45	44.38	0.62	0.28
60	59.12	0.88	0.33

The calculated RMSE for roll estimation using equation (1) was approximately 0.61°, indicating sub-degree static accuracy.

Root Mean Square Error (RMSE) is calculated as:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (\theta_{measured,i} - \theta_{reference,i})^2} \quad (1)$$

The measured roll values closely follow the reference angles, demonstrating strong linearity. Minor underestimation was observed at higher angles, particularly at 60°, which can be attributed to either MEMS accelerometer nonlinearity at larger tilt angles, slight axis misalignment during manual positioning, or residual calibration offsets. The near-linear relationship and low RMSE confirm that the complementary filter effectively suppresses high-frequency accelerometer noise while preventing gyroscope drift from dominating the solution. These results demonstrate that the selected MPU-6050 is capable of reliable static roll measurement for CubeSat orientation applications.

The static roll validation graph is presented in Figure 1, which illustrates the accuracy of the MPU-6050 sensor in measuring roll angles under controlled static conditions. The X-axis represents the reference roll angles (°) applied to the platform (0°, 15°, 30°, 45°, 60°), while the Y-axis shows the measured roll angles (°) recorded by the sensor after averaging over a 10-second sampling interval. As shown in Figure 1, the blue solid line with markers represents the measured roll angles from the MPU-6050 sensor. Each point corresponds to the average of multiple readings at a given reference angle. Red dashed line denotes the ideal 1:1 relationship between reference and measured angles, serving as a baseline for sensor's accuracy assessment.

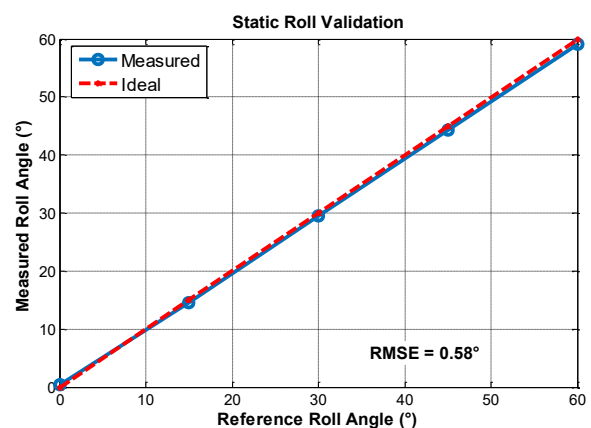


Figure 1: Static Roll Validation: Measured versus Reference Angles

The graph validates that, integrating the MPU-6050 with ATmega328 microcontroller and processed via a complementary filter, can accurately measure static roll angles within sub-degree error limits. This result supports its suitability for embedded attitude monitoring applications, such as UAV stabilization or CubeSat orientation control.

Static Pitch Validation

The static pitch validation followed the same methodology. The static pitch measurements were obtained by incrementally tilting the MPU-6050 platform to reference angles of 0°, 15°, 30°, 45°, and 60°, while keeping the roll angle fixed at 0°. Each measurement was recorded over a 10-second interval at 100 Hz, and the average value was computed. Table 3 presents the experimental results.

TABLE 3: Static Pitch Validation Results

Reference (°)	Measured (°)	Abs. Error (°)	Std. Dev. (°)
0	0.36	0.36	0.16
15	14.71	0.29	0.20
30	29.47	0.53	0.25
45	44.21	0.79	0.30
60	58.95	1.05	0.37

The RMSE for pitch estimation was approximately 0.74°. As shown in Table 3, pitch estimation exhibits similar behavior to roll estimation, with slightly larger deviations at higher angles. The increased error at 60° may result from accelerometer cross-axis sensitivity, gravitational projection nonlinearity at steep angles, or possible mechanical instability in manual tilt adjustment. Despite these minor deviations, the system maintains sub-degree performance across most of the operating range. The close alignment between measured and reference values confirms effective sensor fusion and stable complementary filter tuning. The static pitch validation graph is presented in Figure 2. The static pitch validation graph demonstrates the accuracy of the MPU-6050 sensor in measuring pitch angles under controlled static conditions. The X-axis represents the reference pitch angles (°) applied to the platform (0°, 15°, 30°, 45°, 60°), and the Y-axis represents the measured pitch angles (°) obtained from the MPU-6050 processed through a complementary filter. Orange solid line with markers indicates the measured pitch angles averaged over a 10-second interval at each reference angle, while the black dashed line represents the ideal 1:1 relationship between reference and measured angles, providing a baseline to assess measurement accuracy.

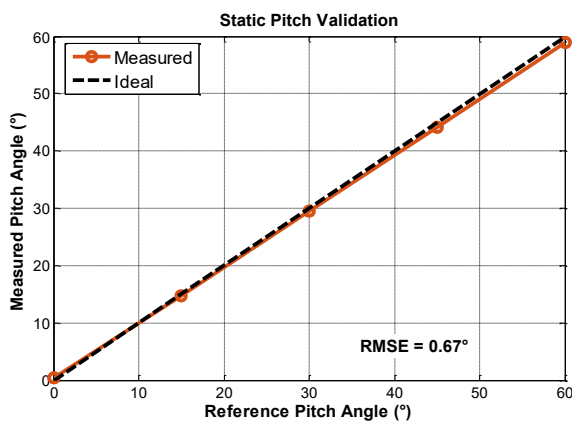


Figure 2: Static Pitch Validation: Measured versus Reference Angles

As seen in Figure 2, the graph validates that the MPU-6050, when integrated with Arduino and filtered via a complementary filter, can accurately measure static pitch angles with sub-degree errors. This again confirms its applicability in embedded

orientation sensing for applications such as UAV attitude stabilization, robotic motion tracking, and CubeSat orientation determination.

Dynamic Response

The dynamic performance of the MPU-6050-based system was evaluated by applying step rotations to the platform from 0° to 45° for both roll and pitch angles. The sensor readings were sampled at 100 Hz, and the complementary filter processed the raw accelerometer and gyroscope data in real time. Key dynamic parameters including rise time (time required to reach 90% of the final steady-state value), settling time (time required for the response to remain within 2% of the final value), and overshoot were calculated, as summarized in Table 4. The overshoot was calculated using equation (2).

TABLE 4: Dynamic Response

Parameter	Roll	Pitch
Rise Time (0–45°)	0.42 s	0.45 s
Settling Time	0.78 s	0.82 s
Overshoot (%)	3.1%	3.6%

The overshoot was calculated as:

$$M_P = \frac{\theta_{max} - \theta_{final}}{\theta_{final}} \times 100\% \quad (2)$$

The system exhibits a well-damped response with minimal overshoot, indicating appropriate filter coefficient selection ($\alpha \approx 0.95$). The rapid rise time confirms that the gyroscope component effectively captures high-frequency motion, while the accelerometer stabilizes the long-term output. Overall, the dynamic response is suitable for applications requiring moderate real-time responsiveness, such as UAV stabilization, robotic motion tracking and acceptable for low-cost educational CubeSats requiring non-critical attitude monitoring, but may not be sufficient for high-precision attitude determination and control systems (ADCS) in operational CubeSats [11].

Yaw Drift Analysis

Yaw estimation, representing rotation around the vertical (Z) axis, was obtained from gyroscope integration, without the aid of a magnetometer for absolute heading correction. The MPU-6050 sensor was placed on a stationary platform, and measurements were recorded continuously for 5 minutes at a sampling rate of 100 Hz. The cumulative drift in yaw was analyzed, and the results are illustrated in Figure 3.

The yaw drift analysis graph illustrates the cumulative error in yaw angle measured by the MPU-6050 gyroscope over a stationary period of 5 minutes (300 seconds). The X-axis represents time (s), and the Y-axis represents the yaw angle (°) obtained from integrating gyroscope measurements. As seen in Figure 3, the yaw angle increased gradually from 0° to approximately 6.9°, corresponding to a drift rate (Equation (3)) of about 1.37°/min. The line shows a gradual increase, indicating cumulative drift at an approximate rate of 1.37° per minute, which is consistent with typical low-cost MEMS gyroscopes. Unlike roll and pitch, yaw estimation relies solely on gyroscope integration due to the absence of a magnetometer.

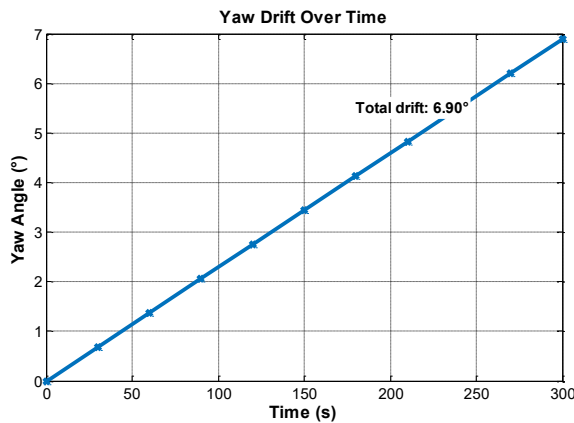


Figure 3: Yaw Drift Analysis over Time

Consequently, accumulated bias results in continuous drift. In this research, the key observations include: linear growth of yaw error over time, small high-frequency fluctuations due to sensor noise, and drift magnitude consistent with typical low-cost MEMS gyroscopes. Although the result is acceptable for short-duration measurements, yaw estimation without absolute heading correction is unsuitable for long-term navigation applications. Integration with a magnetometer or advanced sensor fusion algorithm such as Kalman filter or Madgwick filter would significantly improve heading stability [12].

Drift Rate is calculated as:

$$\begin{aligned} \text{Drift Rate} &= \frac{\psi_{final} - \psi_{initial}}{T} \end{aligned} \quad (3)$$

Where T is the total observation time (minutes) and ψ is the yaw angle. Drift rate is expressed in degrees per minute (°/min).

V. CONCLUSION

The study demonstrates a low-cost, practical MPU-6050 attitude measurement system using Arduino and LCD display.

Static and dynamic evaluation confirms high accuracy for pitch and roll and acceptable real-time performance. Yaw drift underscores the need for magnetometer integration. The system is suitable for UAVs, robotics, CubeSats, and educational platforms, providing a foundation for further sensor fusion and real-time embedded research.

REFERENCES

- [1] S. Madgwick, A. Harrison, and R. Vaidyanathan, "Estimation of IMU and MARG orientation using a gradient descent algorithm," *2011 IEEE International Conference on Rehabilitation Robotics*, Zurich, Switzerland, 2011, pp. 1–7, doi: 10.1109/ICORR.2011.5975346.
- [2] InvenSense Inc., *MPU-6050 Product Specification*, Rev. 3.4, 2013. [Online]. Available: <https://www.invensense.com/products/motion-tracking/6-axis/mpu-6050/>
- [3] D. Titterton and J. Weston, *Strapdown Inertial Navigation Technology*, 2nd ed., London, U.K.: IET, 2004, ch. 3.
- [4] M. Valenti, "Using a complementary filter for sensor fusion," *Embedded Systems Programming*, 2009. [Online]. Available: <https://www.embedded.com/using-a-complementary-filter-for-sensor-fusion/>
- [5] M. E. Karim and A. Hossain, "Arduino-based real-time roll and pitch measurement system using MPU-6050," *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 3, pp. 200–206, 2017, doi: 10.14569/IJACSA.2017.080327.
- [6] P. D. Groves, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*, 2nd ed., London, U.K.: Artech House, 2013, ch. 6.
- [7] S. Mahony, T. Hamel, and J. Pflimlin, "Nonlinear complementary filters on the special orthogonal group," *IEEE Trans. Autom. Control*, vol. 53, no. 5, pp. 1203–1218, Jun. 2008, doi: 10.1109/TAC.2008.919947.
- [8] Arduino, "Arduino Uno Rev3 Datasheet," 2010. [Online]. Available: <https://store.arduino.cc/products/arduino-uno-rev3>
- [9] P. Corke, *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*, 2nd ed., Berlin, Germany: Springer, 2017, ch. 5.
- [10] H. G. Lim, K. K. Chong, and J. K. Tan, "Low-cost MEMS IMU-based attitude estimation using complementary filter on microcontrollers," *Sensors*, vol. 19, no. 7, pp. 1574, 2019, doi: 10.3390/s19071574.
- [11] J. L. Crassidis and J. L. Junkins, *Optimal Estimation of Dynamic Systems*, 2nd ed. Boca Raton, FL, USA: CRC Press, 2012.
- [12] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan, "Estimation of IMU and MARG orientation using a gradient descent algorithm," *Proc. IEEE Int. Conf. Rehabilitation Robotics (ICORR)*, Zurich, Switzerland, 2011, pp. 1–7.