Investigating MEMS Accelerometer Calibration Techniques

Student Author: William Ward

Faculty Mentor: Dr. William Slaton

University of Central Arkansas

Department of Physics and Astronomy

Abstract

Microelectromechanical systems (MEMS) accelerometers are inertial sensors that measure linear acceleration: data that can be used to track a robot's relative displacement. However, displacement estimates derived from an accelerometer's raw measurements drift significantly from ground truth values due to errors in acceleration measurements that accumulate over time. This study investigates methods of calibrating a low-cost MEMS accelerometer using least-squares fitting to remove systematic error from acceleration signals and reduce the effect of displacement estimation drifting.

Background

- An **accelerometer** is a sensor that measures three-dimensional linear acceleration.
- Ideally, engineers could use an accelerometer to precisely track an object's linear displacement, $s(t)$, over time by performing double integration on the measured acceleration, $a(t)$ [2]:

$$s(t) = \int_{t_0}^{t} \left(\int_{t_0}^{t} a(t)dt \right) dt \qquad (1)$$

- However, accelerometers measurement **always** contain small errors, making accelerometer-reliant position tracking systems highly volatile [3].
- For example, Table 1 shows how displacement drift accumulates over time for different acceleration errors.

Table 1: Displacement Drift Over Time $\left[s(t) = \frac{1}{2}at^2\right]$ for Different Acceleration Errors

	$t = 10$ seconds	$t = 30$ seconds	$t = 60$ seconds
$a = 0.001$ m/s²	0.05 meters	0.45 meters	1.8 meters
$a = 0.01$ m/s²	0.5 meters	4.5 meters	18 meters
$a = 0.1$ m/s²	5 meters	45 meters	180 meters

- A common solution to this problem is to combine accelerometers with other sensors, such as a GPS to create a more robust position tracking (**odometry**) system [5, 10].
- In robotics applications, micro electromechanical systems (MEMS) accelerometers are often used because they are small, lightweight, cheap, fast, and able to easily integrate with other electronic devices [3, 4]
- **Project Goal:** Calibrate a MEMS accelerometer (using simple methods) to reduce errors in acceleration and reduce drifting in displacement calculations.

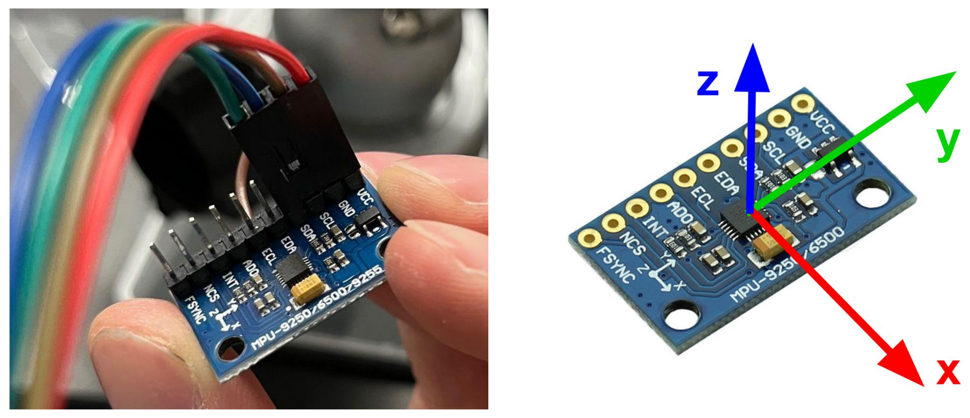

Figure 1: MEMS Inertial Measurement Unit (IMU)

Acceleration Error Modeling

- One approach to calibrating an accelerometer is to create a mathematical model that quantifies sources of error in the sensor.
- In each model, the true acceleration, a, is manipulated by different error matrices to derive the measured values, a'.
- In Model 1, the measured values equals the true values plus a vector, b, of constant offset biases [2].
- In Model 2, the true values are also each multiplied by scale factor constants, S, to add an additional layer of complexity to the model [14].
- In Model 3, off-diagonal matrix components are included to account for any misalignments (non-orthogonality) in the sensor axes [14, 15, 16, 17].

$$\text{Model 1:} \quad \begin{bmatrix} a'_x \\ a'_y \\ a'_z \end{bmatrix} = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} + \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} \qquad (2)$$

$$\text{Model 2:} \quad \begin{bmatrix} a'_x \\ a'_y \\ a'_z \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & S_z \end{bmatrix} \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} + \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} \qquad (3)$$

$$\text{Model 3:} \quad \begin{bmatrix} a'_x \\ a'_y \\ a'_z \end{bmatrix} = \begin{bmatrix} S_{xx} & S_{xy} & S_{xz} \\ S_{yx} & S_{yy} & S_{yz} \\ S_{zx} & S_{zy} & S_{zz} \end{bmatrix} \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} + \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} \qquad (4)$$

- After optimizing the unknown parameters in each model, they can be used to correct new measurements into true values.

Displacement Error Modeling

- Unfortunately, accelerometer error models are not perfect. After integrating calibrated acceleration over time to calculate displacement, some drifting will still remain.
- To further improve displacement calculations, the drift error can be fitted to a polynomial function of time [2, 13]:

$$\mathbf{d}' = \mathbf{d} + \frac{1}{2}\mathbf{q}_2 t^2 + \mathbf{q}_1 t + \mathbf{q}_0 \qquad (5)$$

- After optimizing the model parameters, the model can be used to correct calculated displacements into true values.

Least-Squares Optimization

- The unknown parameters $\boldsymbol{b}$, $\boldsymbol{S}$, and $\boldsymbol{q}$ in Equations 2, 3, 4, and 5 are optimized using least-squares fitting.
- In least squares fitting, a model is fit to a set of data points by minimizing the sum of the square differences between the data coordinates and the trend function [11, 18].
- Data coordinates (a **measured value** paired with an **expected true value**) are generated by recording acceleration data while the sensor is at rest and rotated to **known** orientations relative to gravity.

Experimental Data Collection

- Data measured by an **MPU-9250 accelerometer** is sent to a **Raspberry Pi** for recording and processing.
- The sensor is mounted inside a 3D-printed calibration cube [23] for easy 90° rotation of the sensor during calibration.

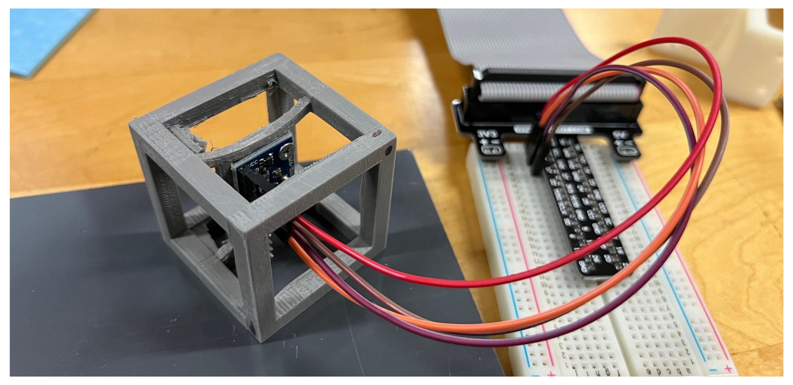

Figure 2: Sensor in calibration cube wired to Raspberry Pi

- Data is recorded and analyzed using Python scripts.
- A **six-position** method is used to collect data and optimize the parameters in the acceleration error models [14].
- While sitting at rest on a level surface, acceleration data is recorded for 30 seconds (at 180 Hz).
- Then, the sensor is rotated to a new orientation, and data is collected again. This repeats for all six orientations.

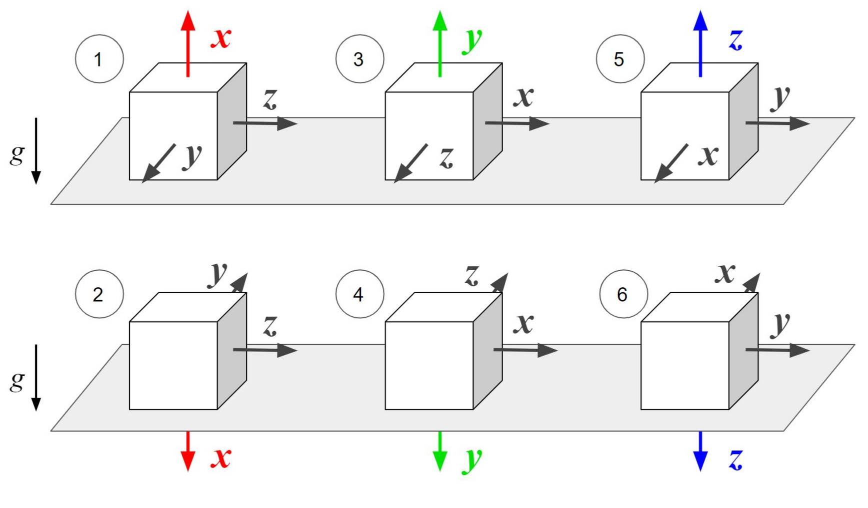

Figure 3: Six-position data collection method [14]

- A new set of acceleration data collected for 60 seconds at rest is calibrated using the optimized acceleration models.
- Then, the calibrated and uncalibrated acceleration data are both integrated to evaluate each model's performance.
- The displacement data of the acceleration model that performs the best is fit to the displacement model (5).
- Finally, an additional set of static test data is calibrated using the optimized displacement error model.

Results

- Table 2 shows the results of optimizing the acceleration models and applying them to a new set of static test data (**i.e., the sensor is at rest and not moving**).
- Acceleration Model 3 performed the best. With no calibration, drift was in the 1000s of meters, but after calibrating Model 3, this reduced to below 100 meters.
- Table 3 shows the results of optimizing Model 3 and the displacement model and applying them to a new set of data.
- Calibrating the displacement data, in addition to calibrating the acceleration data reduced the displacement drift error to a magnitude of less than 10 meters.
- Figure 4 shows a graph of the final calibrated displacement drift over time.

Table 2: Calculated Displacement of Static Test Data (after 60 seconds)

Calibration Method Used	x (m)	y (m)	z (m)
No Calibration	1731	1089	-4002
Model 1: Bias	115	254	413
Model 2: Add Scale Factors	115	254	25
Model 3: Add Misalignments	-45	66	25

Table 3: Calculated Displacement of Static Test Data (after 60 seconds)

Calibration Method Used	x (m)	y (m)	z (m)
No Calibration	1760	1061	-4014
Model 3 (Acceleration Only)	-35	96	25
Model 3 and Displacement Model	-1	-1	-7

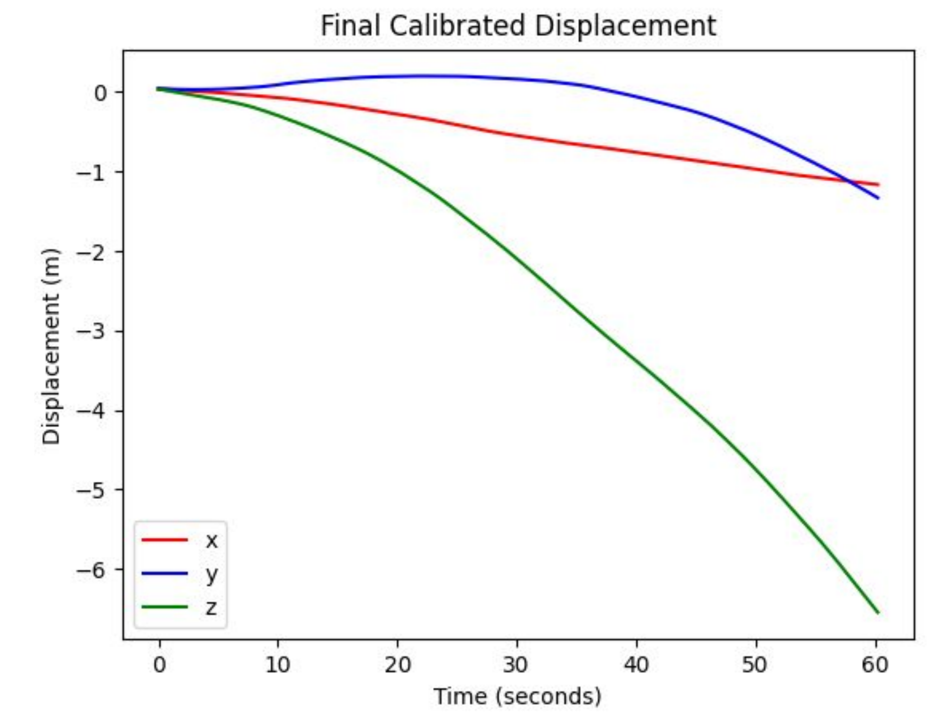

Figure 4: Drift in displacement using Model 3 and Displacement Model

Conclusion

In this project, mathematical modeling was used to calibrate an accelerometer's output data and reduce the effect of displacement drifting. While the results of calibration is still unsuitable for long term odometry, the calibration significantly improved the sensor's performance. Due to inherent flaws in the sensor, no matter how much the data accuracy improves, small amounts of error in acceleration will always exist and accumulate over time in displacement calculations. Nevertheless, when only needed to track displacement over a short period, a calibrated accelerometer can be highly useful for odometry, especially in scenarios where other sensors are unavailable.

In the future, the calibration methods discussed in this project (accelerometer Model 3 and the displacement error model) should be tested on multiple IMU sensors to investigate the consistency in results. Additionally, the methods should be tested on a moving robot to simulate real-world conditions in which odometry estimations are needed. Finally, while these calibration methods are simple and quick to implement, more advanced calibration methods could be researched and tested to investigate whether there is a more effective way of using accelerometers for displacement tracking.

Acknowledgements & References

Department of Physics and Astronomy
University Scholars Program
Capstone Mentor: Dr. William Slaton
Capstone Second Reader: Dr. Lin Zhang