Complementary Filtering for IMU-Based Orientation Estimation with Vicon Ground Truth

Shakthibala Sivagami Balamurugan
Robotics Engineering
Worcester Polytechnic Institute
Email: sbalamurugan@wpi.edu

***Abstract*—This project implements a per-axis complementary filter that mitigates drift errors from both the gyroscope and accelerometer. The algorithm was applied to a dataset comprising 6-DoF IMU data and ground-truth Vicon rotations. Gyroscope integration provides high-frequency, drift-prone orientation estimates, while accelerometer-based tilt estimation offers low-frequency, drift-free corrections. The results demonstrate that a fusion factor of $\alpha = 0.98$ yields a smooth and accurate orientation trajectory, with the gyroscope capturing short-term dynamics and the accelerometer providing long-term drift correction.**

I. Introduction

In many robotics and navigation systems, accurate estimation of 3D orientation from IMU data is essential. Gyroscopes provide good short-term angular velocity measurements but drift over time due to integration errors. Accelerometers provide an absolute measure of the gravity vector but are noisy and affected by linear acceleration. A complementary filter can be used to fuse the strengths of both sensors, blending fast dynamics from the gyroscope with drift-free tilt from the accelerometer.

II. Dataset Description

The dataset consists of MATLAB `.mat` files organized into two subfolders: `Phase1/Data/Train/IMU` and `Phase1/Data/Train/Vicon`. Each IMU file corresponds to a Vicon file with the same numeric suffix, ensuring synchronization between the two datasets. For example, the file `imuRaw1.mat` directly matches with `viconRot1.mat`, meaning they represent the same recording session from both sensors. Each IMU `.mat` file contains two essential variables: `vals` and `ts`. The variable `vals` $\in R^{6\times N}$ represents the raw IMU sensor readings, arranged in the following order:

$$\texttt{vals} = a_x a_y a_z \omega_z \omega_x \omega_y^T.$$

Here, a_x, a_y, and a_z correspond to the raw accelerometer measurements along the x, y, and z axes, respectively, while ω_x, ω_y, and ω_z represent the raw gyroscope readings about the corresponding axes. The second variable, `ts` $\in R^N$, contains the timestamps in seconds, defining the sampling instances for each recorded IMU measurement.

The Vicon `.mat` files, on the other hand, provide ground-truth orientation data for the same recording session. These files contain two key variables: `rots` $\in R^{3\times 3\times N}$, which represent rotation matrices following the Z–Y–X Euler angle convention, and `ts` $\in R^N$, which holds the Vicon timestamps synchronized to its data stream. Together, these IMU and Vicon datasets allow precise comparison between estimated orientation and ground-truth orientation for further analysis and sensor fusion.

III. Bias Estimation and Calibration

Calibration is essential to convert raw IMU counts into physical units. For the accelerometer, scale factors and biases are stored in `IMUParams.mat`. The first row contains the scale vector

$$\mathbf{s} = [s_x,\, s_y,\, s_z],$$

and the second row contains the bias vector

$$\mathbf{b}_a = [b_{a,x},\, b_{a,y},\, b_{a,z}].$$

The calibrated accelerations are computed as:

$$\tilde{a}_x = \frac{a_x + b_{a,x}}{s_x}, \quad \tilde{a}_y = \frac{a_y + b_{a,y}}{s_y}, \quad \tilde{a}_z = \frac{a_z + b_{a,z}}{s_z},$$

ensuring that accelerations are expressed in m/s^2.

For the gyroscope, the bias b_g is estimated under the assumption that the IMU is stationary during the initial samples. The bias is obtained by averaging the first n measurements:

$$b_g = \frac{1}{n}\sum_{i=1}^{n} \omega_i.$$

The calibrated gyroscope readings are then converted into angular velocities (in rad/s) using the scale relationship:

$$\tilde{\omega} = \frac{3300}{1023} \cdot \frac{\pi}{180} \cdot 0.3 \cdot (\omega - b_g)\,.$$

This calibration ensures that both accelerometer and gyroscope data are physically meaningful for orientation estimation.

IV. Orientation Estimation

A. Gyroscope-Only Orientation

Gyroscope-based orientation is estimated by integrating angular velocity over time. In the per-axis complementary filter implementation, however, gyroscope integration was simplified to Euler angles. For example, the roll update is computed as

$$roll_{k+1} = roll_k + \omega_x[k+1]\,\Delta t_k,$$

with analogous expressions for pitch and yaw. This small-angle approximation is computationally simple but can introduce errors for large rotations or coupled axis motions. As shown in Fig.2, The gyroscope-only trajectory gradually diverges from the ground truth due to drift..

B. Accelerometer-Only Orientation

The accelerometer provides the direction of gravity, allowing estimation of roll and pitch. By aligning the measured acceleration vector with the known gravity vector

$$g = [0,\ 0,\ -1]^T,$$

tilt orientation is computed. Since yaw is unobservable from accelerometer data, it is adopted from the gyroscope estimate. The accelerometer-based orientation , Fig.3 tracks tilt well under static conditions but becomes highly noisy and unreliable during dynamic motion.

C. Complementary Filter Fusion

The complementary filter fuses both estimates by applying low-pass characteristics to accelerometer tilt and high-pass characteristics to gyroscope integration. In this implementation, fusion is performed in a per-axis fashion using Euler angles. The update rule is:

$$\theta_{fused}[k+1] = \alpha \cdot \theta_{gyro}[k+1] + (1-\alpha) \cdot \theta_{acc}[k+1],$$

where $\theta \in \{roll, pitch, yaw\}$, θ_{gyro} is the orientation propagated by gyroscope integration, and θ_{acc} is the accelerometer tilt estimate. Here, α is the fusion factor that weights the gyroscope contribution. For example, $\alpha = 0.98$ gives 98% weight to the gyroscope (capturing short-term dynamics) and 2% weight to the accelerometer (providing long-term drift correction).

V. Software Time Synchronization

Since the IMU and Vicon operate at different sampling rates, software synchronization was necessary. For each IMU timestamp $t_{IMU}[i]$, the nearest Vicon timestamp was selected as:

$$j = \arg\min_k \left| t_{Vicon}[k] - t_{IMU}[i] \right|.$$

This was efficiently implemented using binary search with `np.searchsorted`, ensuring computational efficiency even for large datasets. The matched indices aligned the Vicon orientation sequence with the IMU timeline, enabling direct comparison. The Vicon system output, Fig.5 is used as the reference for evaluating all IMU-based orientation estimates. This synchronization ensured that orientation estimates derived from IMU data were evaluated against the temporally closest Vicon ground-truth measurements.

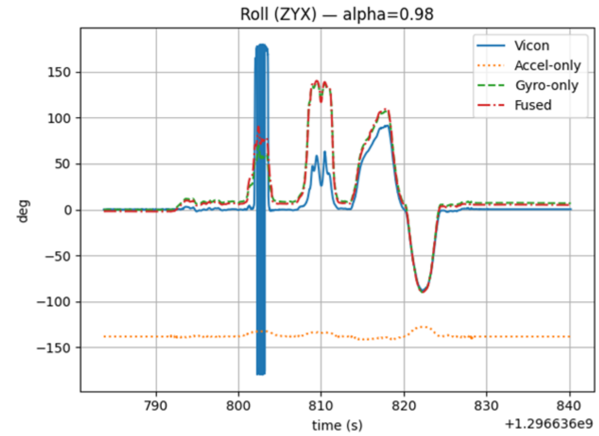

Fig. 1. Benchmarking the results

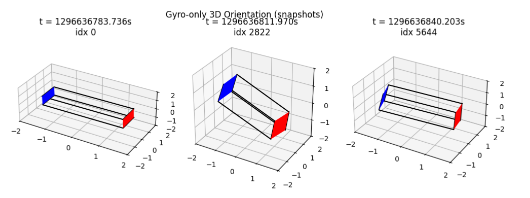

Fig. 2. 3D Orientation of Standalone Gyroscope

VI. Gimbal lock

This is a common phenomenon when working with Euler angles. As shown in Fig. 1, a discontinuity is observed in the roll trajectory, which arises due to gimbal lock in the Z–Y–X Euler representation. Gimbal lock occurs when the pitch angle approaches ±90°, causing two rotational axes to align and leading to a loss of one degree of freedom. In such a configuration, multiple physical orientations may map to the same Euler angle representation, and small smooth changes in the underlying orientation can result in abrupt jumps in the Euler angle plots. It is important to note that this discontinuity is not a failure of the complementary filter itself, nor an error in the underlying rotation estimate, but rather an artifact of the Euler angle parameterization.

VII. Results and Analysis

The complementary filter was tested on multiple sequences. Gyroscope-only integration produced smooth trajectories but gradually diverged from Vicon ground truth due to drift. Accelerometer-only estimation accurately tracked tilt under static conditions but was highly noisy and failed during dynamic motion. The complementary filter successfully fused both signals, producing stable orientation estimates with minimal drift. Euler angle plots (yaw, pitch, roll) confirmed that the fused orientation tracked Vicon measurements closely while remaining smoother and more robust than either method alone. The fused estimate, Fig.4 closely follows the ground truth, achieving both smoothness and robustness.

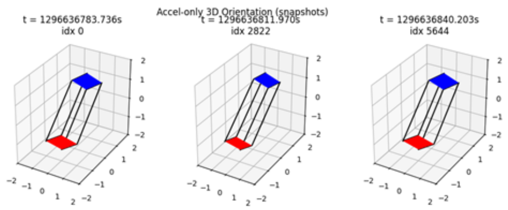

Fig. 3. 3D Orientation of Standalone Accelerometer

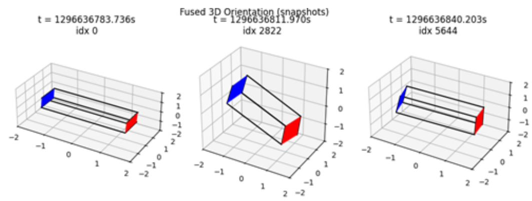

Fig. 4. 3D Orientation using Complementary filter

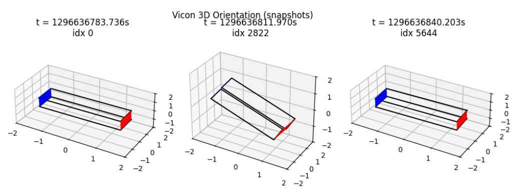

Fig. 5. 3D Orientation using Vicon Motion Capture System

VIII. Conclusion

This study demonstrated the effectiveness of a per-axis complementary filter for orientation estimation using IMU data. By blending gyroscope integration with accelerometer tilt estimates, the filter produced orientation trajectories that were smooth, stable, and closely aligned with Vicon ground truth. A fusion factor of $\alpha = 0.98$ provided the best balance between responsiveness to fast motion and correction of long-term drift. Future work may incorporate magnetometer data or extend the framework to a full manifold-based or Kalman filtering approach for enhanced robustness in real-world robotics applications.

IX. AI Usage

A. Time synchronization

Prompt:I have two dict which contains 'values' and 'timestamps' as key, I want to map them by synchronizing their timestaps in software, Remember not in hardware, Give me a set of ideas arranging them according to their time complexity.

Answer:Got nearest neighbor method as the best fit.

B. Validation of the result

Prompt:I have attached three images, One only with standalone gyroscope, one with stanalone accelerometer, one with complementary filter.

Answer:The graphs are right and the complementary filter tries to align with Vicon data.

X. Supplementary Material

The videos of the complementary filter along with its comparison against ground truth is given: bit.ly/3HWZjXb

References

[1] Wikipedia, "Slerp," [Online]. [Accessed: Aug. 22, 2025].
[2] Stack Overflow, "How to load mat files in Python and access columns individually," [Online]. [Accessed: Aug. 22, 2025].
[3] PRD UMG Teaching, "Sensor Fusion and IMU Orientation," YouTube playlist, [Online]. [Accessed: Aug. 22, 2025].
[4] N. J. Sanket, "Attitude Estimation from IMU," [Online]. [Accessed: Aug. 23, 2025].
[5] Phil's Lab, "IMU Orientation Estimation Tutorial, YouTube video, [Online]. [Accessed: Aug. 23, 2025].
[6] OpenAI, "ChatGPT (GPT5)," [Online]. [Accessed: Aug. 23, 2025].