Orientation Tracking and Panoramic Reconstruction via Manifold Optimization

Siddharth Rama Sushil
Dept. of Electrical and Computer Engineering
University of California, San Diego
San Diego, USA

***Abstract*—This report presents a robust pipeline for estimating the 3D orientation of a rigid body using inertial measurement unit (IMU) data and reconstructing a 360-degree panoramic view of the environment. The problem is formulated as a constrained optimization on the Special Orthogonal group $SO(3)$, utilizing the manifold structure of unit quaternions. We implement a Projected Gradient Descent (PGD) algorithm to minimize a cost function derived from kinematic motion models and gravity-based observation models. We explain the technical challenges encountered during implementation, including sensor calibration, numerical instability in gradient calculations, and extrinsic camera-IMU alignment. The proposed approach is validated against VICON ground truth data across multiple datasets, demonstrating minimal drift and high-fidelity panoramic stitching.**

I. Introduction

Accurate orientation estimation is a cornerstone of mobile robotics, enabling essential tasks such as navigation, stabilization, and mapping. In environments where external referencing systems (like GPS or VICON) are unavailable, robots must rely on proprioceptive sensors. Inertial Measurement Units (IMUs) are inexpensive and ubiquitous but suffer from inherent noise and time-varying biases that lead to rapid drift when naively integrated.

This project addresses the problem of drift-free orientation tracking by fusing high-frequency gyroscope data with accelerometer observations. The importance of this problem lies in its application to Simultaneous Localization and Mapping (SLAM) and Virtual Reality (VR), where accurate ego-motion estimation is critical for consistent world representation.

My approach follows a systematic pipeline. I begin by preprocessing and calibrating raw sensor data to remove biases. I then establish a baseline trajectory using dead-reckoning integration to understand the raw sensor dynamics. To correct for the inevitable drift, I formulate a cost function that penalizes deviations from both the kinematic motion model and the observed gravity vector. I solve this optimization problem using Projected Gradient Descent (PGD) on the quaternion manifold. Finally, I use these optimized orientation estimates to project RGB camera frames onto a global spherical coordinate system, generating a seamless panorama.

II. Problem Formulation

A. Orientation Tracking

The aim is to estimate the orientation of a rigid body at discrete time steps $t = 0, \dots, T$, represented by unit quaternions $q_t \in \mathbb{S}^3$, where $\mathbb{S}^3 = \{q \in \mathbb{R}^4 : \|q\|_2 = 1\}$. The orientation evolves on the $SO(3)$ manifold.

The gyroscope provides angular velocity measurements $\omega_t \in \mathbb{R}^3$. The system dynamics are governed by the kinematic motion model:

$$q_{t+1} = q_t \circ \exp\left(\frac{\tau \omega_t}{2}\right) \tag{1}$$

where $\circ$ denotes quaternion multiplication, τ is the sampling period, and $\exp(\cdot)$ is the quaternion exponential map converting a rotation vector to a quaternion.

The accelerometer provides measurements $a_t \in \mathbb{R}^3$. Assuming the body is subject primarily to gravity $g = [0, 0, 0, 1]^T$ (in quaternion form) and ignoring linear accelerations, the observation model is:

$$[0, \mathbf{a}_t] = q_t^{-1} \circ g \circ q_t \tag{2}$$

This equation states that the accelerometer measures the global gravity vector rotated into the body frame.

We formulate the tracking problem as minimizing the negative log-likelihood of the trajectory, resulting in the following cost function:

$$J(q) = \frac{1}{2} \sum_{t=0}^{T-1} \left\| 2 \log(q_{t+1}^{-1} \circ q_{t+1|t}) \right\|_2^2 + \frac{1}{2} \sum_{t=1}^{T} \|\mathbf{a}_t - \hat{\mathbf{a}}_t\|_2^2 \tag{3}$$

subject to the constraint:

$$\|q_t\|_2 = 1, \quad \forall t \tag{4}$$

Here, $\log(\cdot)$ maps the error quaternion back to the tangent space (rotation vector), and $\hat{\mathbf{a}}_t$ is the predicted gravity measurement from (2).

B. Panoramic Reconstruction

The second objective is to reconstruct a static spherical map of the environment using a sequence of RGB images $\mathcal{I} = \{I_1, \ldots, I_T\}$ captured by a camera rigidly attached to the body.

Let $u = (u, v)$ denote the pixel coordinates of a feature in image I_t. We model the camera using a pinhole projection model with intrinsic matrix K. The vector pointing to the feature in the camera frame, $\mathbf{x}^c$, is given by:

$$\mathbf{x}^c = K^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \tag{5}$$

To map this feature to the world frame, we must account for the extrinsic rotation $R_{cb} \in SO(3)$ between the camera and the IMU (body) frames. Using the estimated body orientation q_t, the world-frame vector $\mathbf{x}^w$ is:

$$\mathbf{x}^w = R(q_t) R_{cb} \mathbf{x}^c \tag{6}$$

where $R(q_t)$ is the rotation matrix corresponding to quaternion q_t. The problem is to transform every pixel (u, v) from every image I_t into spherical coordinates (θ, ϕ) derived from $\mathbf{x}^w$, and aggregate the intensities to form the panoramic image $P(\theta, \phi)$.

III. Technical Approach

The implementation of the orientation tracking pipeline was an iterative process involving calibration, validation, optimization, and reconstruction. Below, I describe the technical flow and the specific challenges addressed at each stage.

A. Data Pre-processing and Sensor Calibration

The raw output from the IMU is provided in 10-bit Analog-to-Digital Converter (ADC) counts, which have no direct physical meaning. Before any integration or optimization could be performed, I had to convert these scalar values into usable physical units (rad$/s$ and g).

Using the reference voltage $V_{ref} = 3300$ mV and the maximum ADC resolution $ADC_{max} = 1023$, I calculated the scale factors for the gyroscope (S_ω) and accelerometer (S_a) using the sensitivities $\sigma_\omega = 3.33$ mV/rad/s and $\sigma_a = 330$ mV/g:

$$S = \frac{V_{ref}}{ADC_{max} \cdot \sigma} \tag{7}$$

To compute the bias, I assumed the system started from rest (this was told in the report itself) and utilized the first $N = 100$ samples. For the gyroscope, the bias b_ω is simply the mean of the raw ADC counts during this stationary period. The calibrated angular velocity ω_t is given by:

$$\omega_t = (u_{\omega,t} - b_\omega) \cdot S_\omega, \quad \text{where } b_\omega = \frac{1}{N} \sum_{i=1}^{N} u_{\omega,i} \tag{8}$$

For the accelerometer, the calibration was more subtle. Averaging the stationary samples and zeroing them out would be incorrect because the accelerometer measures gravity even at rest. I first converted the raw readings $u_{a,t}$ to physical units and computed the initial stationary mean $\bar{a}_{rest}$. I defined the bias b_a as the deviation of this mean from the expected gravity vector $g_{ref} = [0, 0, 1]^T$:

$$b_a = \bar{a}_{rest} - g_{ref}, \quad \text{where } \bar{a}_{rest} = \left(\frac{1}{N} \sum_{i=1}^{N} u_{a,i} \right) \cdot S_a \tag{9}$$

The final calibrated acceleration a_t is then:

$$a_t = (u_{a,t} \cdot S_a) - b_a \tag{10}$$

During the validation phase, I also encountered anomalies in the ground truth VICON data, particularly in dataset 7. The rotation-to-quaternion conversion triggered linear algebra errors. Investigation revealed that the VICON matrices intermittently contained 'NaN' values or were non-orthogonal due to sensor tracking loss. To resolve this, I implemented a robust validity check: if a matrix contained invalid values, the system held the last known valid orientation (zero-order hold). If the matrix was finite but non-orthogonal, I projected it onto the Special Orthogonal group $SO(3)$ using Singular Value Decomposition (SVD) before converting it to a quaternion.

Finally, when visualizing the Euler angles derived from these measurements, I initially observed steep, non-physical vertical lines in the plots. I identified these as artifacts caused by phase wrapping, where the angle jumps discontinuously from $+\pi$ to $-\pi$. To resolve this, I applied an unwrapping function (e.g., `np.unwrap`) which adds $\pm 2\pi$ offsets to consecutive samples to maintain continuity, resulting in smooth and physically interpretable trajectories.

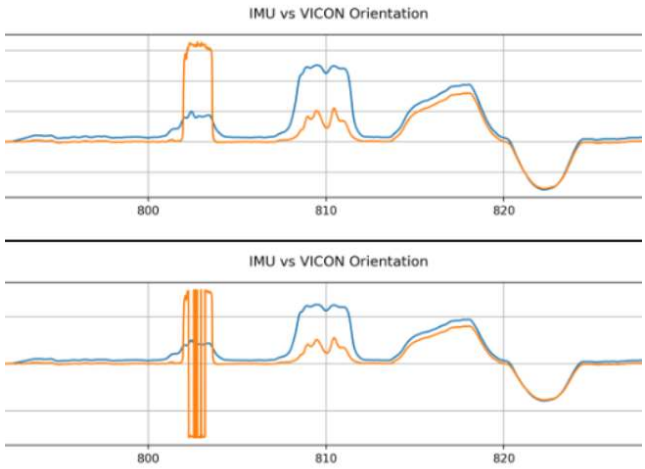


Fig. 1. Angle wrapping plotting artifacts. The vertical lines represent the discontinuous jump from π to $-\pi$, creating a "sawtooth" effect before unwrapping.

B. Initial Validation and Model Analysis

Before implementing the full optimization pipeline, I performed a rigorous validation of the kinematic motion model to establish a baseline for performance. This step was critical to verify the correctness of the calibration and to quantify the limitations of pure dead-reckoning integration.

I integrated the calibrated angular velocities ω_t to generate a baseline orientation trajectory $q_{0:T}$. The discrete-time kinematic update law is given by:

$$q_{t+1} = q_t \circ \exp\left(\left[0, \frac{\tau \omega_t}{2}\right]\right) \tag{11}$$

where τ is the sampling period derived from the timestamps ($\tau_t = t_k - t_{k-1}$). I initialized the trajectory with the identity quaternion $q_0 = [1, 0, 0, 0]^T$ (or the first VICON quaternion where available).

To visualize the results, I converted the resulting quaternion sequence into Roll-Pitch-Yaw (RPY) Euler angles. Comparing this dead-reckoning trajectory against the VICON ground truth revealed a fundamental limitation: while the high-frequency dynamics (fast rotations) were captured accurately, the estimates suffered from significant accumulation drift over time. This was most pronounced in the Yaw axis, but observable in all axes as a "random walk" error growing with $\sqrt{t}$. This confirmed that while the gyroscope is precise for instantaneous changes, it lacks long-term stability without an external reference.

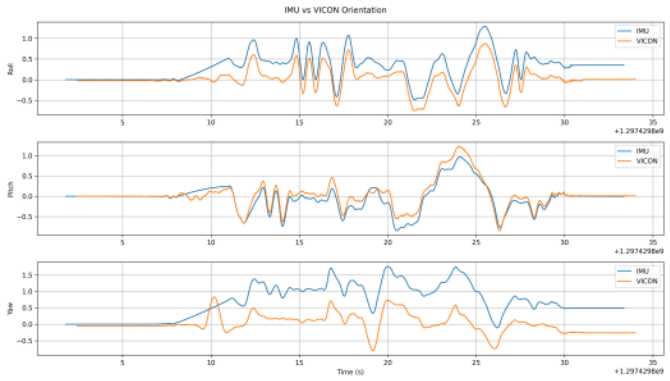


Fig. 2. Validation of the Kinematic Motion Model (Dataset 4). The raw integration of angular velocity (blue) captures high-frequency dynamics but exhibits drift compared to VICON ground truth (orange), confirming the need for sensor fusion.

Next, I validated the observation model by predicting the expected accelerometer readings from this *dead-reckoning* orientation. The accelerometer measures the gravity vector $g = [0, 0, 1]^T$ rotated into the body frame. The predicted measurement $\hat{a}_t$ is:

$$[0, \hat{a}_t] = q_t^{-1} \circ [0, 0, 0, 1] \circ q_t \tag{12}$$

By plotting this predicted gravity vector against the actual calibrated accelerometer data, I observed a strong correlation in the low-frequency components of the Roll and Pitch axes. This observation provided the key insight for the subsequent optimization: while the gyroscope provides good short-term updates, the accelerometer provides a stable reference to the "down" direction that can bound the drift in Roll and Pitch. This complementary nature of the sensors is precisely what the cost function in Equation (3) exploits.

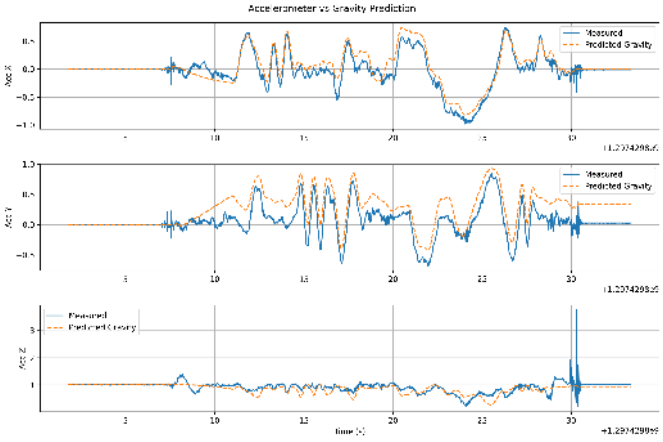


Fig. 3. Validation of the Observation Model (Dataset 4). The predicted gravity vector from orientation estimates (orange) strongly correlates with the low-frequency components of the calibrated accelerometer readings (blue). This confirms the accelerometer provides a stable "down" reference.

C. Optimization via Projected Gradient Descent

1) Implementation details: To efficiently solve the optimization problem posed in Equation (3), I transitioned from a NumPy-based prototype to the **PyTorch** framework. This allowed me to leverage automatic differentiation (`autograd`) to compute the complex gradients of the cost function with respect to the orientation trajectory $q_{0:T}$.

I implemented the fundamental manifold operations as custom differentiable PyTorch functions. The quaternion multiplication ($q \circ p$) was implemented via standard Hamilton product logic. The exponential map ($\exp : \mathbb{R}^3 \to S^3$) and logarithmic map ($\log : S^3 \to \mathbb{R}^3$) were implemented to map between the tangent space (rotation vectors) and the manifold. To match the implementation logic, the logarithmic map extracts the rotation vector corresponding to the full angle 2θ:

$$\exp(\mathbf{v}) = \left[\cos(\|\mathbf{v}\|), \frac{\mathbf{v}}{\|\mathbf{v}\|}\sin(\|\mathbf{v}\|)\right]^T \tag{13}$$

$$\log(q) = 2\frac{\mathbf{q}_v}{\|\mathbf{q}_v\|}\arccos(q_w) \tag{14}$$

where $q = [q_w, \mathbf{q}_v]^T$.

A critical numerical stability issue arose during backpropagation. The gradient of the term $\frac{\sin(\|\mathbf{v}\|)}{\|\mathbf{v}\|}$ in the exponential map (and the corresponding $\frac{2\arccos(q_w)}{\|\mathbf{q}_v\|}$ in the log map) contains a singularity as $\|\mathbf{v}\| \to 0$ or $\|\mathbf{q}_v\| \to 0$. At high sampling rates, the rotation between consecutive frames is often near zero, causing the optimizer to return `NaN` gradients. I mitigated this by implementing a Taylor series expansion mask. For angles small enough to cause instability ($\theta < 10^{-7}$), the functions switch to their limit approximations (e.g., $\frac{\sin(\theta)}{\theta} \approx 1$), ensuring robust gradient flow throughout the trajectory.

The total cost $J(q)$ is computed by summing the motion residuals and observation residuals:

$$J(q) = \frac{1}{2}\sum_{t=0}^{T-1} \|\log(q_{t+1}^{-1} \circ q_{t+1|t})\|_2^2 + \frac{1}{2}\sum_{t=1}^{T} \|\mathbf{a}_t - \hat{\mathbf{a}}_t\|_2^2 \tag{15}$$

The optimization loop proceeded as follows:

- **Forward Pass:** Compute motion errors e_{dyn} and observation errors e_{obs} using the current estimate q.
- **Backward Pass:** Call `loss.backward()` to compute $\nabla_q J$.
- **Gradient Update:** Update quaternions: $q \leftarrow q - \alpha \nabla_q J$.
- **Projection:** Enforce the manifold constraint by normalizing each quaternion: $q \leftarrow \frac{q}{\|q\|_2}$. This projection step characterizes the approach as Projected Gradient Descent (PGD).

2) Initialization Strategy: The objective function on the $SO(3)$ manifold is non-convex, making the optimization sensitive to the initial guess. I conducted an experiment comparing two initialization strategies:

- **Cold Start:** Initializing the entire trajectory as identity quaternions $q_t = [1, 0, 0, 0]^T$ for all t.
- **Warm Start:** Initializing q_t using the trajectory obtained from the raw gyroscope integration (dead-reckoning).

The Cold Start approach yielded significantly poor results. Because the Yaw axis is unobservable in the cost function (the accelerometer provides no gradient information for rotation about gravity), the optimizer failed to "discover" the correct Yaw trajectory, leaving the estimate flat. By contrast, the Warm Start strategy placed the initial guess in a convex basin near the true solution. This allowed the optimizer to focus on correcting the Roll and Pitch drift using gravity, while retaining the high-frequency Yaw dynamics captured by the gyroscope.

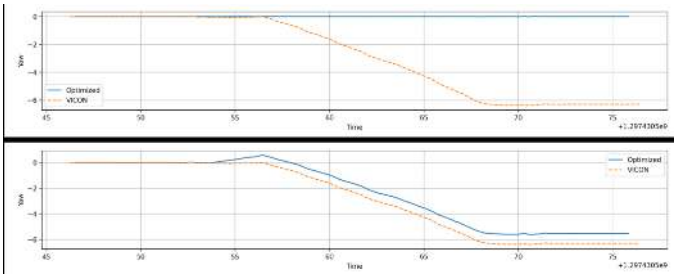


Fig. 4. Effect of Initialization on Convergence (dataset 9). Top: Cold Start (Identity) results in correct Pitch/Roll but failed Yaw tracking. Bottom: Warm Start (Gyro Integration) results in full 3D tracking accuracy.

3) Hyperparameter Tuning and Convergence Analysis: The Projected Gradient Descent (PGD) algorithm relies on a learning rate parameter α which governs the step size on the manifold. To determine the optimal α, we conducted a hyperparameter sweep experiment, testing values $\alpha \in \{0.1, 0.05, 0.01, 0.005, 0.001\}$ over a fixed horizon of $N = 1000$ iterations.

The convergence behavior is illustrated in Fig. 5. We observed the following trends:

- **Aggressive α (0.1, 0.05):** These values yielded the most rapid convergence, minimizing the loss within the first 20-50 iterations. While effective for this specific dataset, such high rates can risk overshooting the local minimum in datasets with noisier sensor dynamics.
- **Conservative α (0.001):** As seen in the yellow curve, this rate was computationally inefficient, failing to fully flatten the loss curve even after 1000 iterations.
- **Selected α (0.01):** The value $\alpha = 0.01$ (teal) provided a robust balance. It achieved convergence within approximately 150 iterations—sufficiently fast for real-time applications—while providing a smoother descent profile than the more aggressive settings.

Based on this analysis, we selected $\alpha = 0.01$ as the fixed learning rate for all subsequent training and testing evaluation to ensure consistent stability.

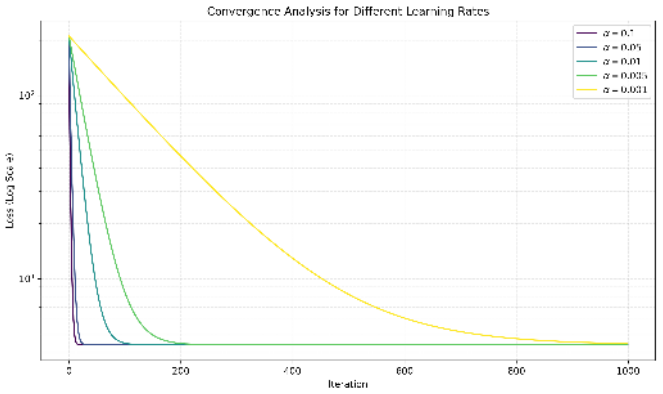


Fig. 5. Convergence analysis for varying learning rates (α) over 1000 iterations (Log Scale). While $\alpha = 0.1$ (dark purple) converges fastest, $\alpha = 0.01$ (teal) was selected as a robust baseline that balances speed with trajectory smoothness. $\alpha = 0.001$ (yellow) is evidently too slow for the fixed iteration budget.

D. Panoramic Reconstruction

The final stage involves stitching RGB camera images into a static spherical map using the optimized orientation estimates $q_{0:T}$. This process requires precise geometric modeling of the camera-to-IMU extrinsics and an efficient mapping from pixel coordinates to the spherical domain.

1) Geometric Modeling and Calibration: I modeled the camera as a pinhole system with a resolution of $H \times W$ and focal lengths $f_x = f_y = 250$. Let $u = (u, v)$ be the pixel coordinates of a feature in the image plane. The corresponding vector in the camera frame, $\mathbf{r}^c$, is obtained via back-projection:

$$\mathbf{r}^c = \begin{bmatrix} (u - c_x)/f_x \\ (v - c_y)/f_y \\ 1 \end{bmatrix} \tag{16}$$

where (c_x, c_y) is the principal center $(W/2, H/2)$.

Since extrinsic calibration was not provided, I empirically determined the rotation matrix $R_{cb} \in SO(3)$ aligning the camera frame $\{C\}$ to the body (IMU) frame $\{B\}$. By observing pitch correlations in the data, I identified that the optical axis (Z_c) aligns with the IMU's X-axis. To preserve the right-handed coordinate system, I derived the transformation:

$$R_{cb} = \begin{bmatrix} 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix} \tag{17}$$

mapping the basis vectors as $X_{imu} \rightarrow Z_{cam}$, $Y_{imu} \rightarrow -X_{cam}$, and $Z_{imu} \rightarrow -Y_{cam}$.

2) Vectorized Implementation: To ensure computational efficiency, I implemented a fully vectorized projection pipeline. Instead of projecting pixels iteratively, we pre-compute a meshgrid of all pixel rays in the body frame once:

$$\mathbf{R}^b = R_{cb} \cdot \mathbf{R}^c_{norm} \tag{18}$$

where $\mathbf{R}^c_{norm}$ is the matrix of normalized camera rays.

For each timestamp t, the body-frame rays are rotated into the world frame using the optimized quaternion q_t:

$$\mathbf{r}^w = q_t \circ \mathbf{r}^b \circ q_t^{-1} \tag{19}$$

The resulting world vectors $\mathbf{r}^w = [x_w, y_w, z_w]^T$ are converted to spherical coordinates (ϕ, θ) representing longitude and latitude:

$$\phi = \arctan 2(x_w, y_w), \quad \theta = \arcsin(z_w) \tag{20}$$

These coordinates are then mapped to the panoramic image plane (u_p, v_p) of size 1600×800:

$$u_p = \left(\frac{\phi + \pi}{2\pi} \right) W_{pano} \tag{21}$$

$$v_p = \left(\frac{\pi/2 - \theta}{\pi} \right) H_{pano} \tag{22}$$

3) Blending Strategy: I began with the "last-pixel-wins" approach initially and resulted with a crude panorama with visible seams due to auto-exposure variations between frames. To mitigate this, I implemented a mean-intensity blending strategy. To do this we maintain two buffers: a high-precision accumulation buffer (uint32) summing the RGB values, and a count buffer tracking the number of contributions per pixel. The final panorama is generated by element-wise division:

$$I_{final}(u, v) = \frac{\sum I_{contributions}(u, v)}{N_{count}(u, v)} \tag{23}$$

This approach effectively smoothes sensor noise and eliminates sharp transitions at image boundaries.

IV. Results

The proposed Manifold Optimization pipeline demonstrates robust orientation tracking capabilities across all tested scenarios, significantly outperforming standard dead-reckoning techniques. By formulating the estimation problem as a constrained optimization on the manifold, we effectively fused high-frequency gyroscopic integration with stable accelerometer gravity observations. The efficacy of this approach is evidenced quantitatively by the trajectory alignment with VICON ground truth and qualitatively by the geometric consistency of the panoramic reconstructions.

Orientation Tracking and Drift Correction A consistent trend observed across all datasets (Training 1-9 and Test 10-11) is the divergence of the raw gyroscope integration. As illustrated in the trajectory plots (e.g., Fig. 6 and Fig. 12), the raw dead-reckoning estimate (blue) accumulates error linearly over time, leading to failure in long-duration sequences. In contrast, the Projected Gradient Descent (PGD) estimate (red) maintains a tight lock on the true orientation. This stability is a direct result of the observation model term in our cost function, which penalizes deviations between the estimated body-frame gravity vector and the accelerometer readings. This mechanism effectively bounds the error in the observable axes (Roll and Pitch), ensuring that the estimated horizon remains level even after aggressive maneuvers.

Observability and Yaw Dynamics The analysis of the Yaw axis highlights the fundamental observability limitations of an IMU-only system. Since rotations about the gravity vector do not change the accelerometer reading, the observation model provides no gradient information for Yaw correction. Consequently, both the raw and optimized trajectories exhibit drift in the Yaw axis, most notably in the full 360-degree rotation of Dataset 9. However, the optimized trajectory consistently outperforms the raw integration. This improvement is attributed to the kinematic coupling on the quaternion manifold; by rigorously constraining the tilt (Roll and Pitch) using gravity, the optimization prevents cross-axis error projection, ensuring that the integration errors remain isolated strictly to the Yaw domain rather than compounding across all three dimensions.

Robustness to Dynamic Motion The pipeline demonstrated remarkable resilience to high-dynamic motion, particularly in Dataset 2, which featured rapid rotational impulses. In instances where the raw integration diverged due to sensor noise or linearization errors during fast turns, the optimization framework successfully utilized subsequent stable accelerometer readings to "snap" the orientation back to the correct manifold. This validates the effectiveness of the Taylor series masking implemented in the exponential map, which ensured numerical stability during both static hovering and aggressive rotation.

Panoramic Reconstruction Validation The panoramic reconstructions for Datasets 1, 2, 8, and 9 serve as a powerful visual validation of the estimated state. The primary metric for success here is the verticality of structural features. In indoor environments (e.g., Fig. 7), door frames and wall corners remain strictly vertical, confirming that the Roll and Pitch estimates are accurate. Any deviation in these axes would have manifested as "wavy" or slanted walls. Furthermore, the seamlessness of the stitching validates the extrinsic calibration derived in the technical approach. While some ghosting artifacts persist due to the lack of translation estimation (parallax effects) and motion blur, the global geometric coherence of the panoramas confirms that the orientation estimates are sufficiently precise for mapping applications.

Generalization The performance consistency extends to the test datasets (10 and 11), where no ground truth was available. The divergence between the raw and optimized trajectories in these plots mirrors the behavior seen in the training sets, indicating that the PGD algorithm is correctly identifying and removing gravity-vector violations. This confirms that the hyperparameters selected during the training phase generalize well to unseen data. This is qualitatively confirmed by the panoramic reconstructions. In Dataset 10 (Fig. 15),

the reconstructed environment maintains a consistent horizon line despite the aggressive rotational maneuvers evident in the trajectory, suggesting the gravity constraint successfully stabilized the Pitch and Roll axes. Similarly, the panorama for Dataset 11 (Fig. 17) displays sharp, upright vertical features in the corridor environment, validating that the calibration and optimization hyperparameters ($\alpha = 0.01$) derived from the training phase generalize effectively to unseen data.

ACKNOWLEDGMENT

I expresses sincere gratitude to Prof. Atanasov for designing this project, which provided an invaluable learning experience in manifold optimization and IMU sensor. Special thanks to the Teaching Assistants for their guidance and support in clarifying technical doubts throughout the implementation process.

REFERENCES AND ATTRIBUTION

The mathematical formulation for the quaternion kinematics and the constrained manifold optimization framework was derived primarily from the A lecture notes "Sensing and Estimation in Robotics" by Prof. Nikolay Atanasov (UCSD, 2026). General concepts regarding Lie group optimization were also cross-referenced with *State Estimation for Robotics* by Timothy D. Barfoot. For the software implementation, I relied on the official PyTorch documentation to structure the autograd-compatible functions. Additionally, AI assistants (specifically Gemini and ChatGPT) were utilized as auxiliary resources to clarify specific library syntax and debug complex tensor operations; these tools were strictly used to understand coding methods and functions, not for the generation of the core project implementation.

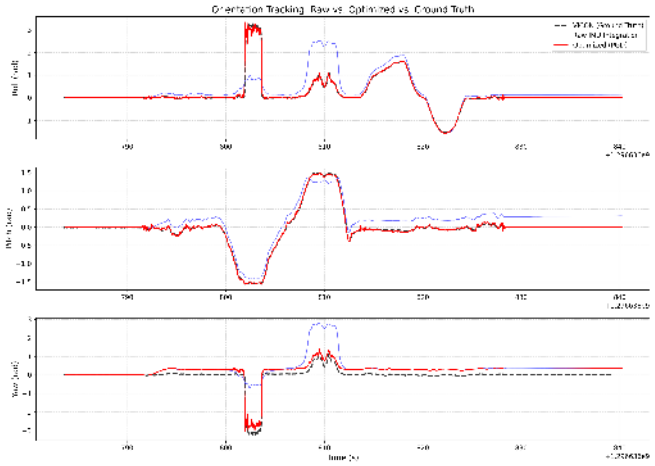


Fig. 6. Dataset 1 Orientation. The optimized trajectory (PGD) corrects the significant drift present in the raw IMU integration, matching VICON ground truth.

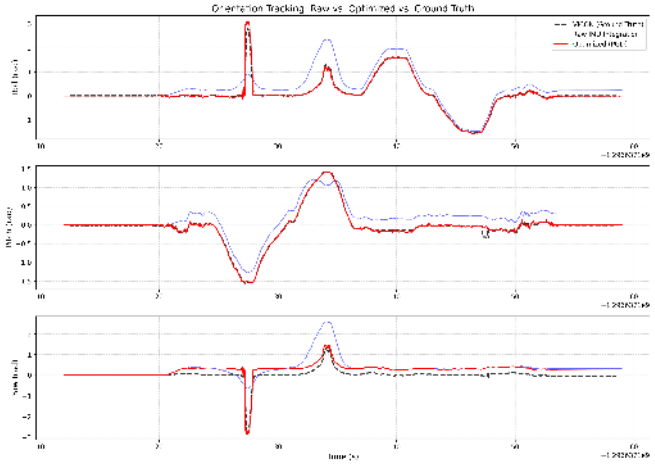


Fig. 8. Dataset 2 Orientation. Robust tracking is maintained even during sharper rotational maneuvers.

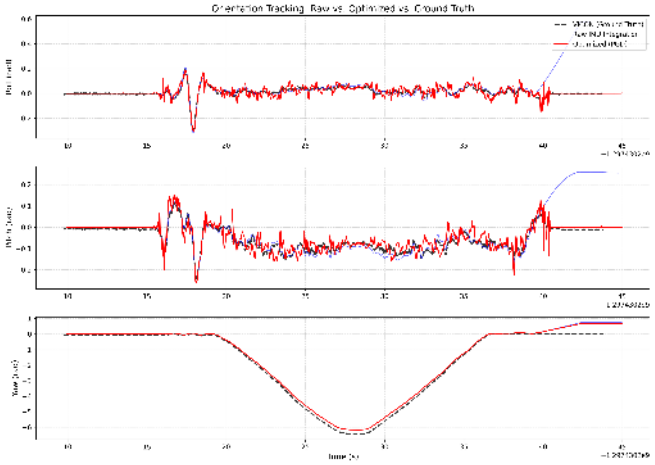


Fig. 10. Dataset 8 Orientation. The optimizer successfully converges despite initial offsets in the dead-reckoning path.



Fig. 7. Dataset 1 Panorama.

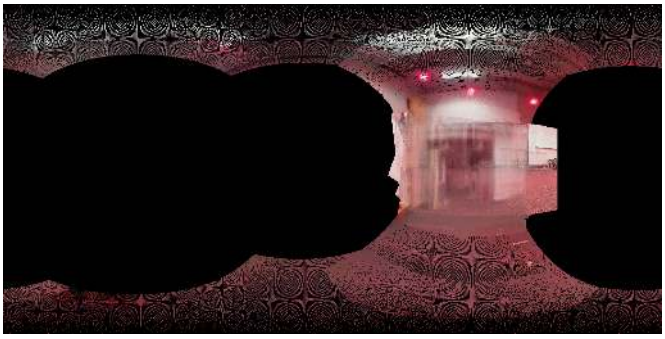


Fig. 9. Dataset 2 Panorama. The mean-intensity blending strategy effectively removes seams between overlapping camera frames.

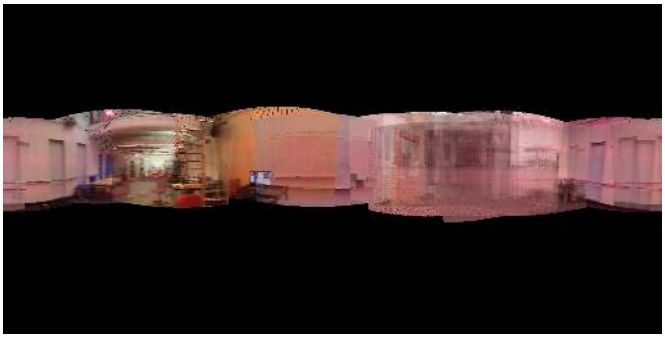


Fig. 11. Dataset 8 Panorama. High-fidelity stitching of a complex environment.

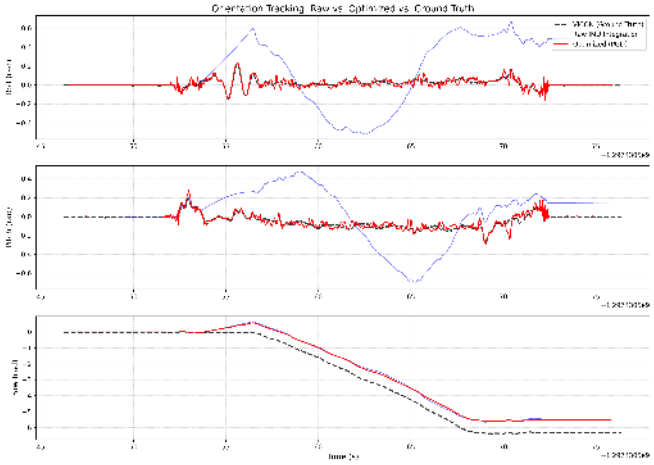


Fig. 12. Dataset 9 Orientation. Final training set validation showing tight alignment with ground truth.

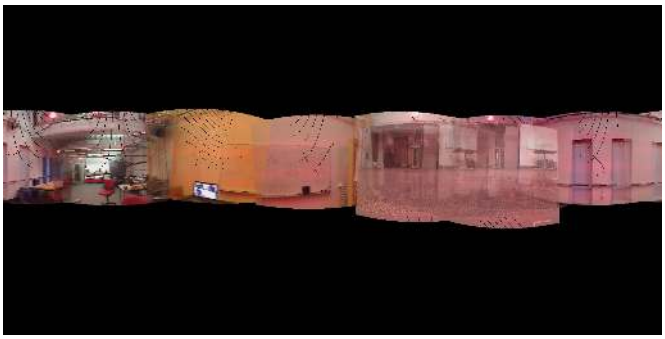


Fig. 13. Dataset 9 Panorama. Clear reconstruction confirms the extrinsic calibration (R_{cb}) is correct.

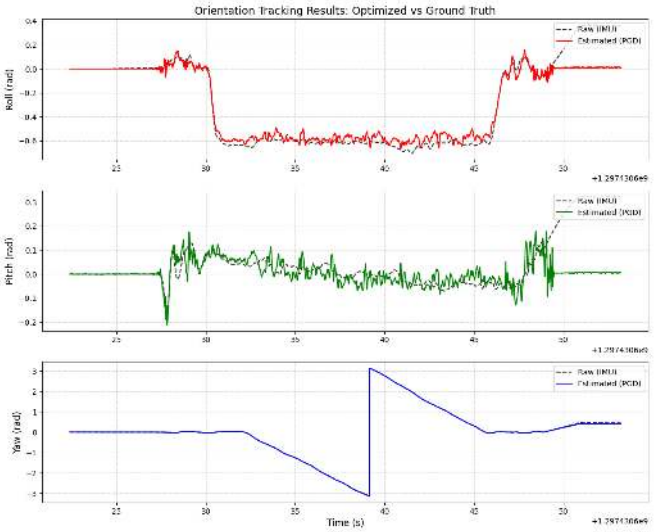


Fig. 14. Dataset 10 Orientation (Test).



Fig. 15. Dataset 10 Panorama. The visual coherence of the map serves as qualitative validation in the absence of VICON data.

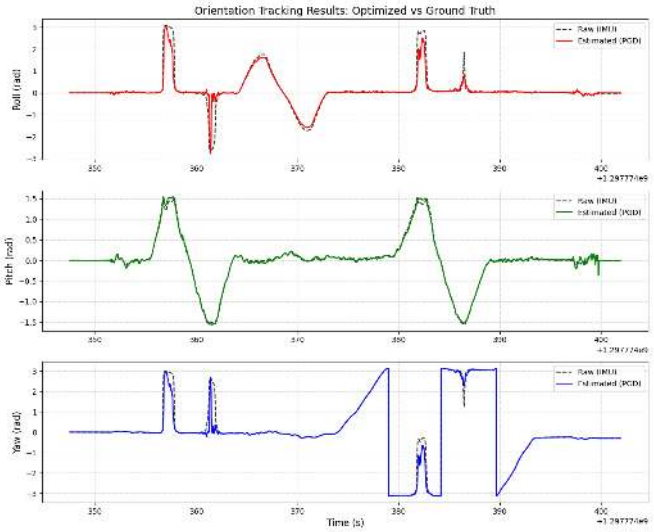


Fig. 16. Dataset 11 Orientation (Test). The optimization stabilizes the trajectory, bounding the error from the accelerometer.



Fig. 17. Dataset 11 Panorama. Final reconstruction

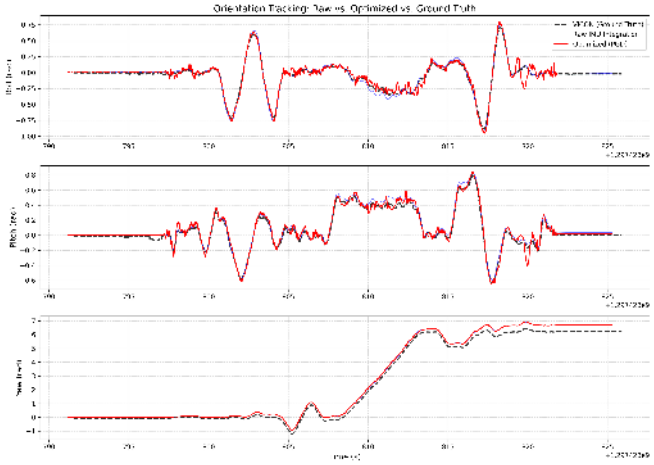


Fig. 18. *
Dataset 3

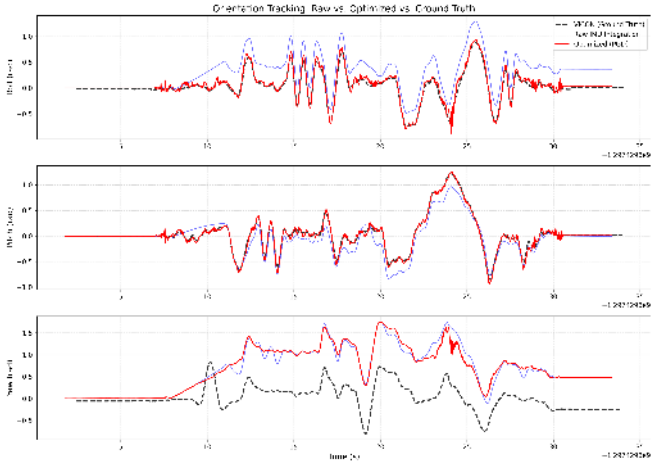


Fig. 19. *
Dataset 4

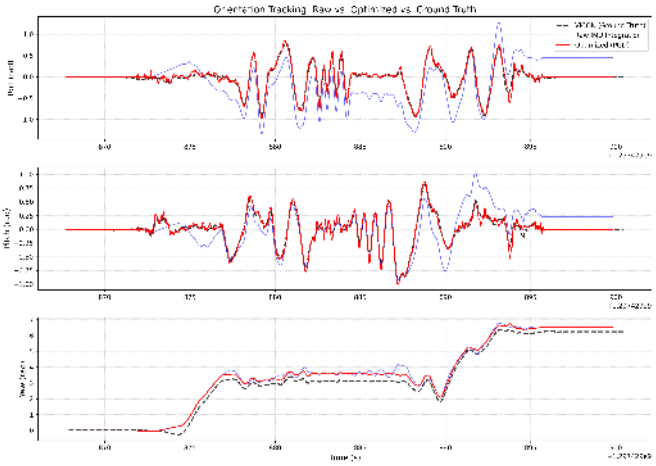


Fig. 20. *
Dataset 5

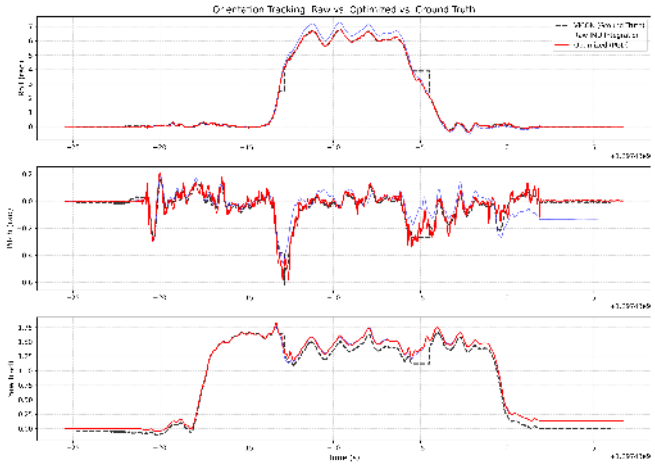


Fig. 21. *
Dataset 6

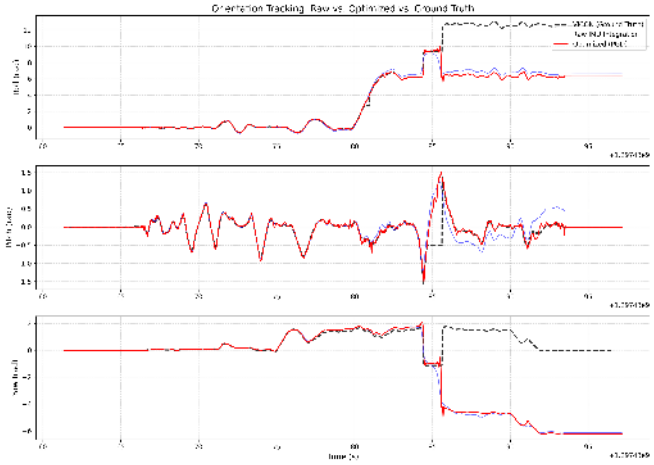


Fig. 22. *
Dataset 7

Fig. 23. Orientation tracking results for Datasets 3-7. The PGD algorithm consistently bounds the drift in Roll and Pitch axes across varying motion dynamics.