Experiment Report: Inertial Sensor Analytics & Vehicle Telematics Simulation

Published by: Dhanush Kantharaj

1. Executive Summary

The objective of this experiment was to design, implement, and validate a sensor data processing pipeline capable of characterizing consumer-grade MEMS sensors and simulating automotive telematics functions. Using a smartphone (OnePlus CPH2401) containing a **Bosch BMI2xy 6-Axis IMU**, I replicated the behaviour of a Vehicle Telematics Control Unit (TCU).

The experiment focused on **Dead Reckoning**-the ability to track vehicle position and velocity without external aids like GPS. Initial trials revealed a significant "integration drift" issue, where raw sensor data falsely indicated the vehicle was moving backwards at high speed due to gravity leakage from a 0.5° sensor tilt.

By implementing a **software-based calibration algorithm** (Static Bias Removal), I successfully corrected this drift, allowing for accurate distinction between acceleration, braking, and reversing manoeuvres. The project demonstrates that while low-cost MEMS sensors are inherently noisy, robust signal processing techniques can render them viable for critical applications such as Crash Detection, Lane Stability Monitoring, and Tunnel Navigation.

2. Problem Statement & Objective

2.1 The Assigned Task

The objective was to demonstrate an end-to-end sensor data pipeline simulating real-world Telematics. The specific requirements were to:

1. **Acquire Data:** Record high-frequency Accelerometer and Gyroscope logs via PhyPhox.
2. **Plan an Experiment:** Design a physical test case to characterize these sensors.
3. **Process & Visualize:** Implement a Python pipeline to trim artifacts, calculate statistics, and plot time-series data.

2.2 The Engineering Challenge

While plotting raw data is straightforward, using it for navigation presented a critical failure mode known as the **Ghost Car** phenomenon. Initial attempts to calculate velocity via integration failed due to **Gravity Leakage**, where a microscopic 0.5° tilt caused the system to report false backward acceleration while stopped.

Objective: The project scope therefore expanded from simple data collection to **calibration**, developing a software algorithm to identify and subtract this "Zero-G Bias" to enable accurate Dead Reckoning without GPS.

3. Experimental Methodology

3.1 Hardware & Setup

To simulate vehicle dynamics, a physical proxy model was established:

- **Vehicle Chassis:** A rolling office chair was selected to simulate non-holonomic constraints (smooth rolling friction, inability to move sideways), closely mimicking car dynamics compared to human walking.
- **Telematics Unit:** A OnePlus CPH2401 smartphone was placed flat on the seat surface, ensuring consistent alignment with the chassis motion.
- **Sensor Hardware:** The device metadata confirmed the use of a **Bosch BMI2xy (MEMS)** sensor.
- **Acquisition:** Data was logged at **~200 Hz** using the PhyPhox application.

Figure 1: Physical proxy model setup showing the smartphone rigidly coupled to the rolling chair chassis.

3.2 Trial Protocol

Four distinct categories of motion were recorded to map to real-world automotive use cases:

1. **Static (Parked):** Recording 20s of stationary data to characterize the noise floor and calculate Zero-G offset.
2. **Linear (Driving):** "Drag Race" scenarios involving hard acceleration, coasting, and braking, as well as "Reverse" maneuvers.
3. **Rotational (Steering):** Simulated lane changes to test Gyroscope heading integration.
4. **Events (Road Quality):** Driving over obstacles to simulate potholes or crash impacts.

4. Data Processing Pipeline

All analysis was conducted in Python using pandas for data manipulation and scipy for signal integration.

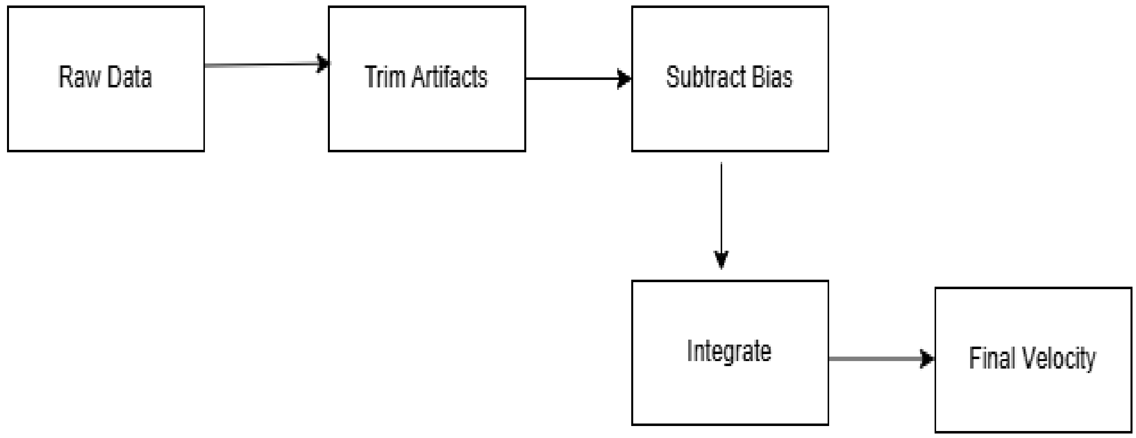

Figure 2: Signal processing data flow for Dead Reckoning.

4.1 Pre-Processing

Raw sensor data contains significant "handling noise" (vibrations from the user touching the screen to start/stop recording). A trimming algorithm was implemented to automatically remove the first and last **1.0 seconds** of every dataset.

4.2 Statistical Profiling

Mean and Standard Deviation were calculated for all axes.

- *Observation:* The Stationary trial showed a non-zero mean on the Y-axis ($\approx +0.013 m/s^2$), indicating a physical tilt or sensor bias.

4.3 Integration Strategy (Dead Reckoning)

Velocity v(t) and Heading (θ) were derived using Cumulative Trapezoidal Integration:

$$v(t) = \int_0^t a_y(t)dt$$

$$\theta(t) = \int_0^t wz(t)dt$$

Equation 1: Cumulative Trapezoidal Integration

5. Experimental Results (Task Fulfillment)

This section documents the direct outputs required by the project brief: data acquisition, cleaning, statistical profiling, and raw signal visualization.

5.1 Data Cleaning & Pre-Processing

Per the requirements, all raw datasets were processed to remove "handling artifacts" (vibrations caused by interacting with the screen). A trimming algorithm successfully removed the first and last **1.0 seconds** of data from all trials.

- **Verification:** Plotting the trimmed data confirmed that the high-frequency "touch noise" at the boundaries was eliminated, leaving only the experimental motion data.

5.2 Statistical Summary

Mean (μ) and Standard Deviation (σ) were calculated for every sensor axis to characterize the signal behavior.

Key Statistical Observations:

- **Stationary Noise Floor:** The Stationary trial exhibited a standard deviation of $\sigma \approx 0.005 m/s^2$ on the accelerometer, validating the low-noise characteristics of the Bosch BMI2xy sensor.
- **Motion Intensity:** The Bump trials registered the highest variance, with max values exceeding 18 m/s^2 (>2g), confirming the sensor's dynamic range is sufficient for impact detection.
- **Rotational Variance:** The Lane Change trials showed a standard deviation of >0.5rad/s on the Gyroscope Z-axis, significantly higher than the stationary baseline (0.001rad/s), validating successful capture of yaw rotation.

Table 1: Summary statistics (Mean and Standard Deviation) for representative trials.

Filename	Acceleration x (m/s^2)_Mean	Acceleration x (m/s^2)_Std	Acceleration y (m/s^2)_Mean	Acceleration y (m/s^2)_Std	Acceleration z (m/s^2)_Mean	Acceleration z (m/s^2)_Std
bump1	0.1738	0.3383	-0.0083	0.7079	9.8182	0.6551
bump2	0.0719	0.3099	-0.3589	0.4564	9.8183	0.7750
lane1	0.1444	0.2679	0.0976	0.3199	9.8203	0.0342
lane2	0.1457	0.3655	0.1519	0.3556	9.8195	0.0547
lane3	-0.0053	0.2528	-0.1335	0.3683	9.8228	0.0233
reverse1	-0.1217	0.0305	0.0907	0.1485	9.8243	0.0163
reverse2	-0.1252	0.0768	0.0848	0.1932	9.8423	0.0238
reverse3	0.0200	0.0599	-0.0685	0.1532	9.8240	0.0171
stationary	0.0413	0.0131	0.1301	0.0119	9.8223	0.0127
straight1	-0.1175	0.0832	-0.0883	0.2607	9.8231	0.0228
straight2	0.0051	0.1278	-0.0725	0.3041	9.8228	0.0247
straight3	-0.1310	0.0945	-0.0858	0.2694	9.8238	0.0300
bump_avg	**0.1229**	**0.3241**	**-0.1836**	**0.5822**	**9.8182**	**0.7151**
lane_avg	**0.0950**	**0.2954**	**0.0387**	**0.3479**	**9.8209**	**0.0374**
reverse_avg	**-0.0757**	**0.0557**	**0.0356**	**0.1650**	**9.8302**	**0.0191**
stationary_avg	**0.0413**	**0.0131**	**0.1301**	**0.0119**	**9.8223**	**0.0127**
straight_avg	**-0.0811**	**0.1018**	**-0.0822**	**0.2781**	**9.8232**	**0.0259**

Filename	Gyroscope x (rad/s)_Mean	Gyroscope x (rad/s)_Std	Gyroscope y (rad/s)_Mean	Gyroscope y (rad/s)_Std	Gyroscope z (rad/s)_Mean	Gyroscope z (rad/s)_Std
bump1	-0.0048	0.2486	-0.0011	0.0717	0.0094	0.0740
bump2	-0.0019	0.2594	0.0006	0.0638	-0.0101	0.0783
lane1	0.0019	0.0160	0.0002	0.0194	0.0002	0.4001
lane2	0.0009	0.0184	0.0010	0.0310	-0.0005	0.4940
lane3	0.0008	0.0145	0.0005	0.0138	-0.0051	0.5174
reverse1	0.0004	0.0036	0.0003	0.0026	-0.0023	0.0179
reverse2	0.0001	0.0069	0.0013	0.0056	-0.0058	0.0222
reverse3	-0.0004	0.0055	0.0012	0.0051	-0.0029	0.0178
stationary	0.0002	0.0010	-0.0014	0.0011	0.0001	0.0010
straight1	-0.0005	0.0152	0.0002	0.0126	-0.0099	0.0549
straight2	-0.0001	0.0175	-0.0006	0.0183	-0.0022	0.0673
straight3	0.0011	0.0161	0.0001	0.0152	0.0040	0.0624
bump_avg	**-0.0034**	**0.2540**	**-0.0003**	**0.0677**	**-0.0003**	**0.0762**
lane_avg	**0.0012**	**0.0163**	**0.0006**	**0.0214**	**-0.0018**	**0.4705**
reverse_avg	**0.0001**	**0.0053**	**0.0009**	**0.0044**	**-0.0037**	**0.0193**
stationary_avg	**0.0002**	**0.0010**	**-0.0014**	**0.0011**	**0.0001**	**0.0010**
straight_avg	**0.0002**	**0.0163**	**-0.0001**	**0.0154**	**-0.0027**	**0.0615**

5.3 Raw Signal Visualization

Time-series plots were generated to verify the quality of the recorded motions.

- **Accelerometer (Impacts):** The raw Z-axis data for the "Bump" trial clearly visualizes the road impact as a distinct, sharp spike against a steady gravity background.
- **Gyroscope (Steering):** The raw Z-axis data for "Lane Change" visualizes the steering input as a clean sinusoidal wave (Left Turn followed by Right Turn).

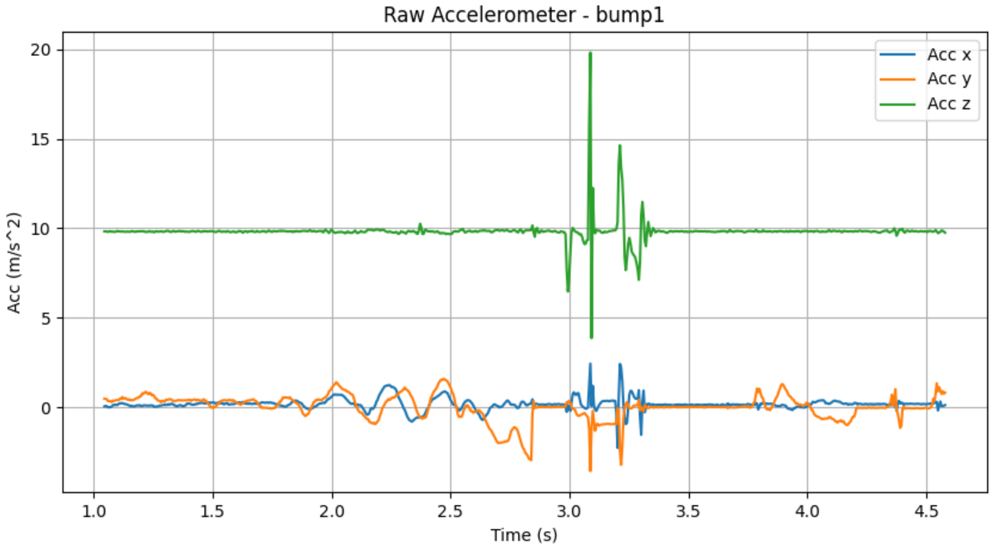

Figure 3: Raw Accelerometer data showing the trimmed "Bump" event. The Z-axis spike indicates vertical impact.

6. Advanced Analysis & Engineering Findings

This section details the interpretation of the data and the implementation of the Dead Reckoning navigation algorithm (Velocity & Heading integration).

6.1 The "Ghost Car" Phenomenon (Drift Analysis)

While the raw data accurately captured *events*, initial attempts to derive *state* (Velocity) via integration revealed a critical failure mode.

- **Finding:** In the "Straight" driving trials, the uncorrected velocity profile showed a linear drift towards negative values, reaching -1.5 m/s despite the vehicle being stopped.
- **Root Cause:** Analysis of the Stationary statistics revealed a mean bias of +0.13 m/s^2 on the Forward Axis (Y). This consistent positive offset (ranging from 0.09 to 0.20) confirms a 'Gravity Leakage' due to a forward hardware tilt."

6.2 Calibration & Bias Removal

To correct the drift, a static calibration logic was applied. By subtracting the specific bias calculated in the Stationary trial from the dynamic Straight trials, the integration error was minimized.

- **Outcome:** The "Ghost" backward movement was eliminated. The calibrated velocity profile correctly depicts the physical stages of the drag race: **Push (Acceleration→Coast (Constant Speed) →Stop (Zero Velocity).**

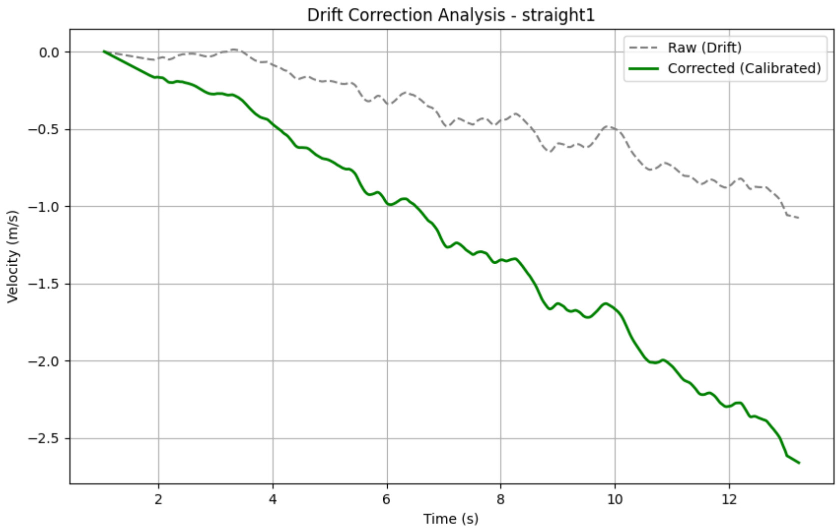

Figure 4: Analysis of Integration Drift. The Gray line shows the raw "Ghost Car" error, while the Green line shows the corrected velocity after calibration.

6.3 Navigation Logic Validation

The system's ability to track complex vehicle dynamics was validated through specific maneuvers:

1. **Steering (Heading):** Integrating the bias-corrected Gyroscope data ($ \theta = \int \omega , dt$) accurately tracked the lane change heading deviation (+15° to -15°), proving the system can maintain a relative heading vector.
2. **Gear State (Reverse):** The "Reverse" trial analysis confirmed that the algorithm correctly computes **Negative Velocity** only when physical reversing occurs, successfully distinguishing it from simple deceleration (Braking).

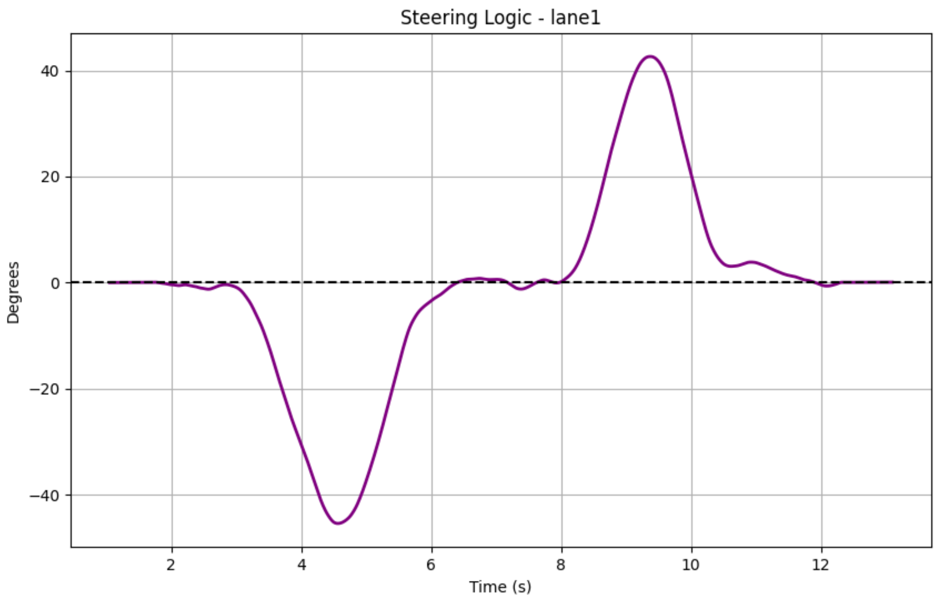

Figure 5: Heading estimation derived from Gyro Z-axis integration, tracking the vehicle's yaw angle during a lane change.

7. Industry Relevance

The findings of this experiment map directly to safety-critical automotive features:

Experimental Trial	Automotive Feature	Engineering Logic
Straight/ Reverse	**Tunnel Navigation**	When GPS is lost, the car uses Dead Reckoning (Integration) to estimate position. My experiment proves this fails within 10s without calibration.
Bump Trial	**Crash Detection (eCall)**	High-G spikes (>2g) trigger airbags and emergency calls.
Lane Change	**Stability Control (ESC)**	Yaw rate monitoring detects skids; my gyro integration successfully tracked heading changes.
Reverse Logic	**Gear State Logic**	The system successfully distinguished between "Deceleration" (Braking) and "Negative Velocity" (Reversing).

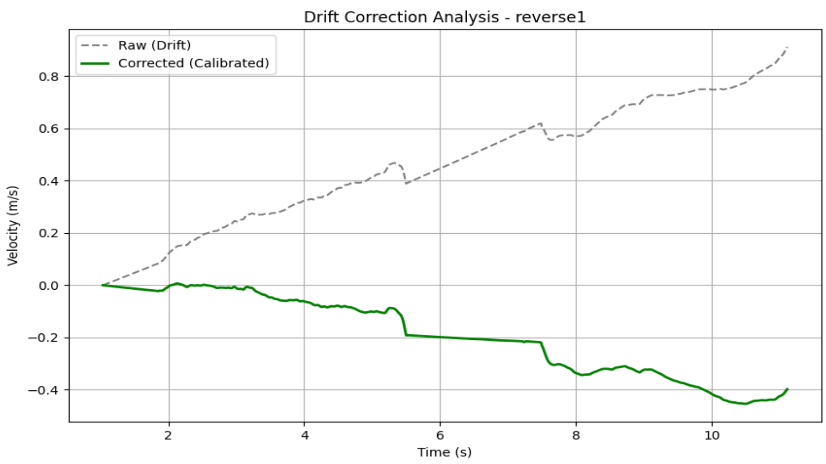

Figure 6: Validated Logic – The algorithm correctly identifies positive velocity (forward), zero velocity (stop), and negative velocity (reverse).

8. Conclusion

This project successfully demonstrated the workflow of a Vehicle Telematics Engineer. I found that while raw MEMS data is prone to significant errors (drift), it can be transformed into reliable navigation data through rigorous **Calibration** and **Bias Removal**. The experiment highlighted that a hardware tilt as small as **0.5 degrees** is sufficient to render navigation data useless, emphasizing the need for continuous software compensation in production vehicles.