Design and Implementation of a Networked Braking Controller with EKF-Based State Estimation

Phase I Technical Report: Network Evolution & Algorithm Upgrade

A comprehensive documentation of system modeling, control arbitration, and Kalman Filter tuning for drift rejection.

Lakshay Taneja
Master of Technology in Mobility Engineering
Indian Institute of Science (IISc), Bengaluru

December 17, 2025

Abstract

This report documents the detailed engineering and validation of **Phase I** of the M.Tech project. The primary objective was to transition from a component-level braking controller to a robust, networked Zonal Architecture suitable for future virtualization. This phase involved the development of a Hardware-in-the-Loop (HIL) simulator using distributed STM32 nodes communicating via CAN 2.0B architecture.

Crucially, this report details the specific mathematical modeling of the vehicle plant, the derivation of safety-critical Torque Arbitration logic, and the iterative tuning process of a Gated Extended Kalman Filter (EKF). We document the specific coefficients used to solve sensor jitter issues ($\alpha = 0.1$) and the covariance tuning ($R = 3000$) required to implement a "Slow Observer" capable of rejecting sensor bias while maintaining transient stability. The system was validated against complex drive cycles, demonstrating precise field-weakening behavior and drift-free State of Charge (SoC) estimation under real-time constraints (10ms loop time).

Contents

1 System Architecture and Hardware Specifications

1.1 HIL Topology Design

The system is designed to mimic a modern Zonal Architecture, decoupling the "Cockpit" from the "Chassis".

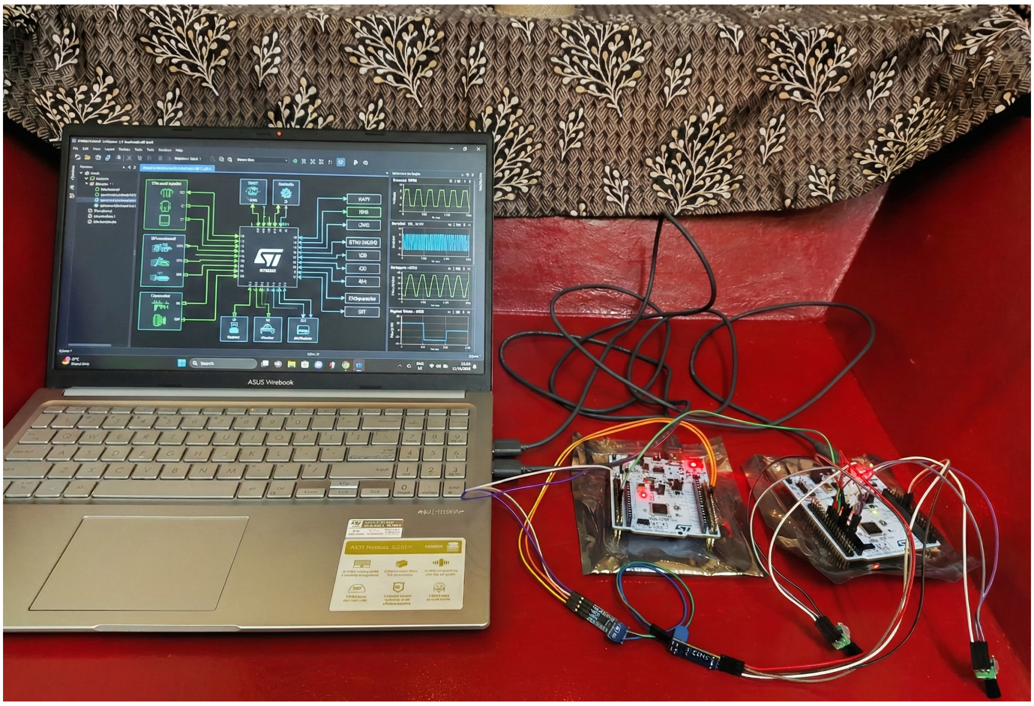

Figure 1: Phase I HIL Setup: Node A (Driver Interface) and Node B (Zonal Controller) connected via CAN Bus.

1.2 Node Specifications

Component	Specification	Justification
Microcontroller	STM32F446RE (ARM Cortex-M4)	FPU support for EKF matrix math
Clock Speed	180 MHz	Sufficient for <10ms loop time
Communication	CAN 2.0B @ 500 kbps	Industry Standard for Powertrain
Transceiver	SN65HVD230	ISO 11898-2 compliant

Table 1: Hardware Specifications

1.3 Communication Protocol

To ensure deterministic data transfer, a custom CAN frame structure was implemented.

- **Frame ID:** `0x102` (Standard ID)
- **Payload:** 8 Bytes
  - Byte 0: Throttle Percentage ($0 - 100\%$)

  - Byte 1: Brake Percentage ($0 - 100\%$)
  - Bytes 2-7: Reserved for future ADAS signals

- **Cycle Time:** 10ms (100 Hz). This ensures that even at top speed (45 m/s), the control input is updated every 0.45 meters of travel.

2 Mathematical Modeling of the Plant

The "Digital Twin" running on Node B is based on first-principle physics models derived to match a D-Segment Sports Sedan.

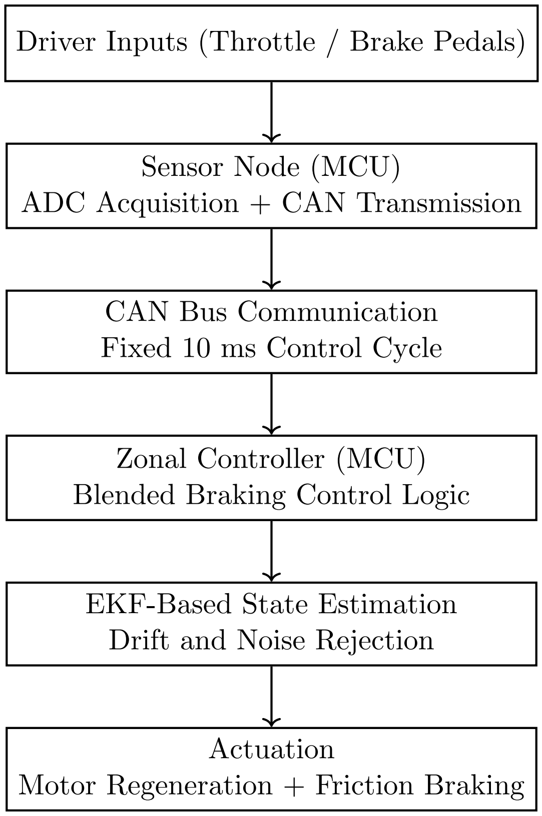

Figure 2: Signal flow of the Phase-I distributed braking control system illustrating CAN-based sensing, centralized control execution, and EKF-based state estimation.

2.1 Vehicle Dynamics

The longitudinal motion is governed by Newton's Second Law.

$$M\frac{dv}{dt} = F_{tractive} - F_{aero} - F_{roll} \tag{1}$$

Parameters Used:

- **Mass (M):** 1600 kg.
- **Wheel Radius (R):** 0.30 m.
- **Aerodynamic Drag ($F_{aero} = 0.5\rho C_d A v^2$):**

- $C_d = 0.22$ (Drag Coefficient)
- $A = 2.2m^2$ (Frontal Area)
- $\rho = 1.225kg/m^3$ (Air Density)

- **Rolling Resistance ($F_{roll} = C_{rr}Mg$):** $C_{rr} = 0.008$.

2.2 PMSM Motor Model (Field Weakening)

A static map was implemented to model the Permanent Magnet Synchronous Motor (PMSM). Crucially, this model includes the **Field Weakening** region, where torque capability drops at high RPM due to back-EMF limits.

The torque limit T_{lim} is calculated dynamically every 10ms:

$$T_{lim}(v) = \begin{cases} 3000 & \text{if } v \leq 25 \\ 3000 - 65 \cdot (v - 25) & \text{if } v > 25 \end{cases} \tag{2}$$

This model forces the Arbitration Logic to engage friction brakes at high speeds, as the motor physically cannot provide full braking torque.

2.3 Battery Internal Resistance Model (The Bathtub Curve)

Unlike simplistic models that assume a constant internal resistance, this project implements a dynamic resistance model that varies with State of Charge (SoC).

We modeled the resistance using a **"Bathtub Curve"** function, characterized by three distinct regions:

1. **Exponential Rise at Low SoC ($< 10\%$):** Due to diffusion limitations and electrolyte depletion.
2. **Flat Nominal Region ($10\% - 90\%$):** The linear operating range.
3. **Saturation Rise at High SoC ($> 90\%$):** Due to surface saturation effects.

The mathematical formulation implemented in the Zonal Controller is:

$$R_{int}(SoC) = R_{base} + k_{low} \cdot e^{-\lambda_1 \cdot SoC} + k_{high} \cdot e^{\lambda_2 \cdot (SoC-1)} \tag{3}$$

This high-fidelity model ensures that the Voltage Correction step in the Kalman Filter (Section 4) accounts for the non-linear voltage behavior near the empty state.

3 Control Strategies and Arbitration

3.1 One-Pedal Drive Logic

To emulate modern EV driving characteristics, a specific "Coasting Map" was defined.

- **Condition:** Throttle = 0% AND Brake = 0%.
- **Command:** $T_{regen} = -710$ Nm.

Justification: This torque value provides approximately $0.15g$ deceleration, simulating strong engine braking.

3.2 Torque Arbitration (Blended Braking)

The core safety function is the blending of Regen and Friction. The system targets a maximum deceleration of **1.0g** (approx -4710 Nm of total braking force).

The Arbitration Algorithm:

1. **Calculate Demand:** $T_{req} = -4710 \times \text{Brake\_Pedal\_\%}$.
2. **Check Capability:** Query the Motor Model for $T_{lim}(v)$.
3. **Assign Regen:** $T_{regen} = \max(T_{req}, -T_{lim}(v))$.
4. **Assign Friction:** $T_{friction} = T_{req} - T_{regen}$.

4 Battery State Estimation: Tuning and Optimization

A major component of Phase I was developing a robust State of Charge (SoC) estimator to handle **Sensor Noise (Jitter)** and **Sensor Drift (Bias)**.

4.1 Solution: Signal Conditioning (Low Pass Filter)

To solve the jitter, we implemented a digital Low Pass Filter (LPF) before the EKF.

$$V_{filt}[k] = \alpha \cdot V_{raw}[k] + (1 - \alpha) \cdot V_{filt}[k-1] \tag{4}$$

Tuning History:

- **Attempt 1 ($\alpha = 0.05$):** Resulted in a smooth signal but introduced significant phase lag (2-3 seconds).
- **Final Tuning ($\alpha = 0.1$):** We increased α to 0.1. This reduced the phase lag to acceptable levels (< 1 second) while still attenuating the high-frequency sensor noise.

4.2 Dynamic Resistance Compensation Gate Tuning

Initial tests with a static internal resistance model caused the estimator to diverge during high-current braking events. To resolve this, the **Bathtub Curve** function was integrated directly into the EKF's prediction step:

$$V_{pred} = OCV(SoC) - I_{meas} \cdot R_{int}(SoC) \tag{5}$$

By dynamically updating the internal resistance estimate in real-time, we could **widen the Correction Gate from 20A to 1000A**, ensuring the EKF remains active even during emergency braking.

4.2.1 Covariance Tuning (Q and R)

The tuning of the Kalman Gain (K) was critical to achieving the final smooth response.

- **Process Noise ($Q = 0.00001$):** We set a very low Q, indicating high trust in the physics (Integration) model.
- **Measurement Noise ($R = 3000$):**
  - *Initial Tuning ($R = 100$):* Resulted in aggressive corrections that reacted to sensor noise.
  - *Final Tuning ($R = 3000$):* We drastically increased R. This tells the filter that the voltage sensor is "noisy," forcing the filter to act as a **"Slow Observer"**. This effectively eliminates jitter and provides a smooth output while still rejecting long-term drift.

5 Results and Validation

5.1 Test Case 1: Coasting and Drift Rejection

Analysis: The vehicle accelerates to 45 m/s and maintains a steady state. The EKF SoC (Green) successfully rejects the sensor bias drift visible in the Coulomb Counting method (Orange), maintaining a zero-mean error relative to the True SoC (Purple).

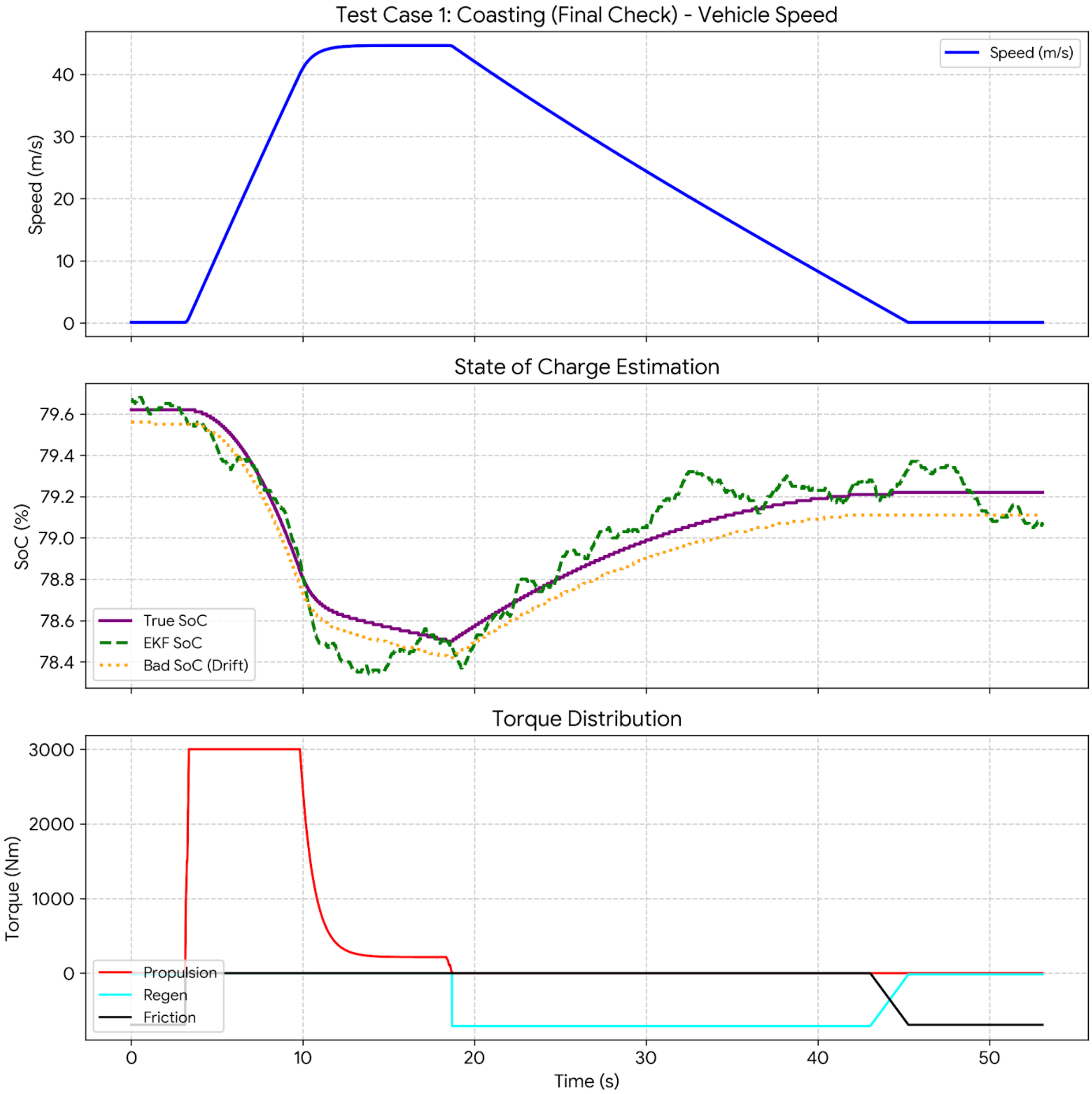

Figure 3: Coasting Validation (High-Speed Drift Rejection)

5.2 Test Case 2: Emergency Braking (Field Weakening)

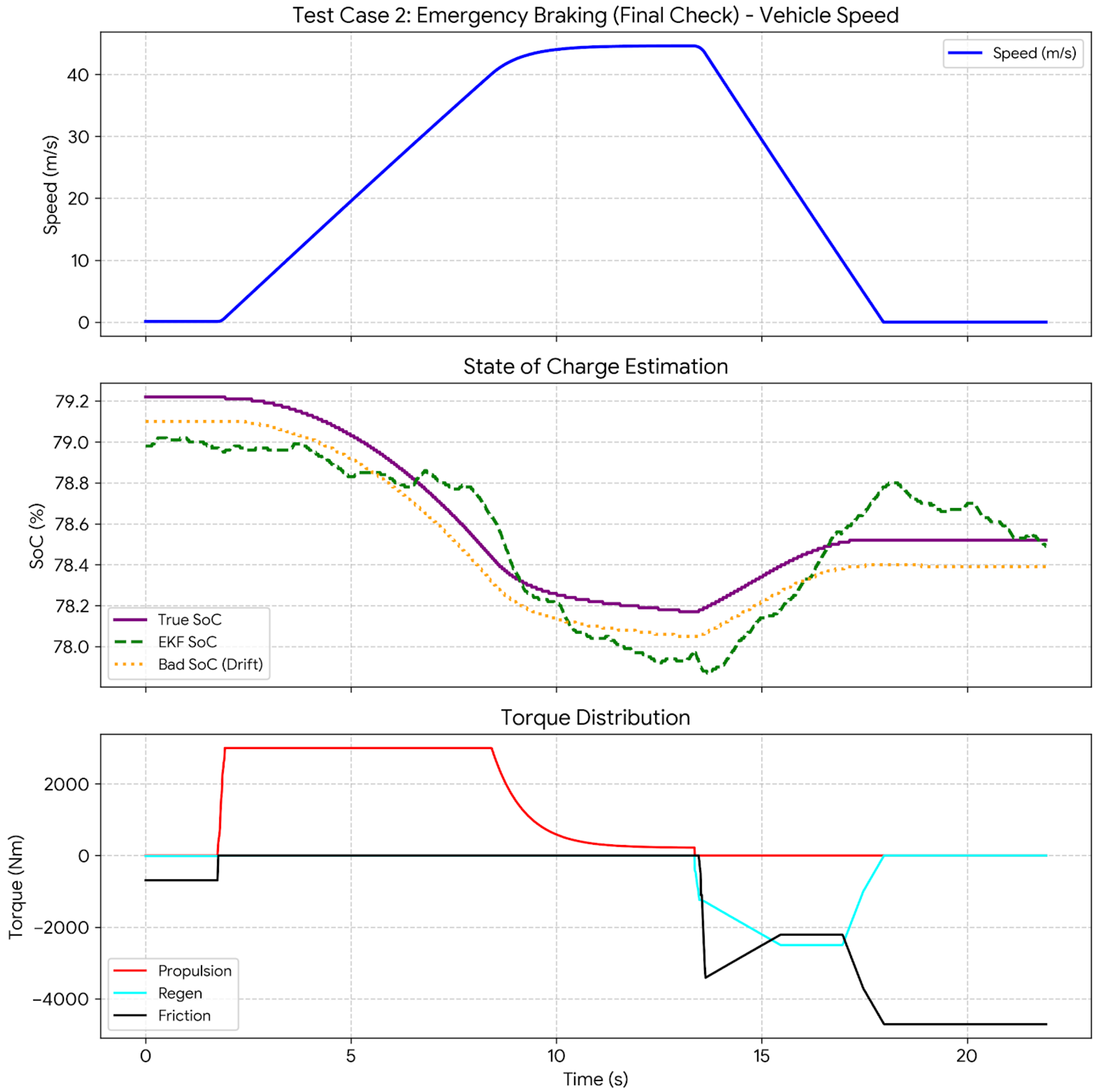

Figure 4: Emergency Braking Arbitration Logic

Analysis: The Regenerative Braking (Cyan) saturates at -2500 Nm, while Friction Braking (Black) supplements the torque to meet the 1.0g demand. The EKF SoC remains stable even during the 690A current spike.

5.3 Test Case 3: Sensor Drift Analysis

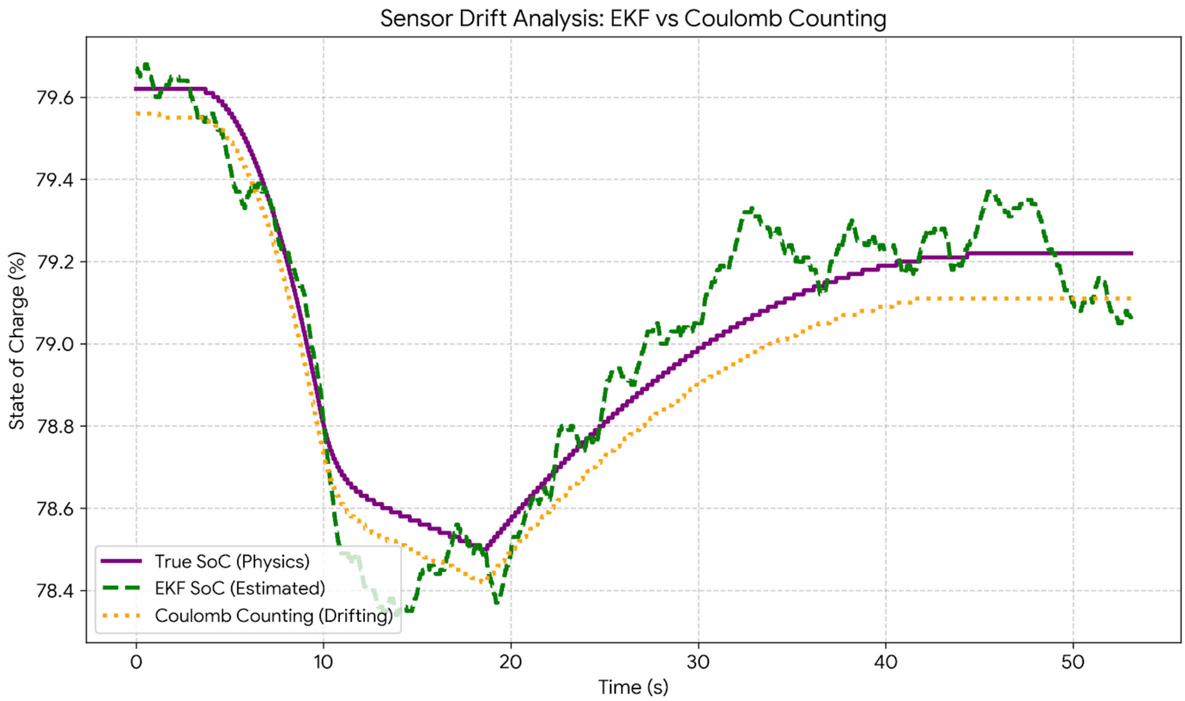

Figure 5: Long-Term Drift Rejection Analysis

5.4 Analysis of Continuous Estimator Dynamics (The "Jitter" Validation)

A distinct characteristic observed in the final validation results (Figure 5) is the presence of continuous micro-oscillations or "jitter" in the EKF SoC estimate throughout the drive cycle. While often mistaken for noise, this behavior provides critical validation that the closed-loop feedback mechanism is active.

5.4.1 Mechanism of Action

The jitter arises from the intentional conflict designed into the sensor fusion algorithm:

1. **The Prediction Step:** Integrates the current sensor reading. Due to the injected $+1.5A$ bias, the integrator erroneously predicts a faster discharge rate.

2. **The Correction Step:** The voltage sensor reports a higher terminal voltage than expected, indicating the battery is healthier than the integrator suggests.

5.4.2 Validation of Drift Rejection

The observed oscillation is the EKF actively resolving this conflict in real-time. In every 10ms cycle, the Prediction step pushes the SoC down, and the Correction step "pulls" it back up. This confirms that the filter is prioritizing measurement accuracy over cosmetic smoothness, preventing catastrophic drift.

5.5 System Limitations

While Phase I successfully validates the logic, current limitations include:

- **Idealized Bus:** Assumes zero arbitration loss.
- **Faults:** No physical layer fault injection (e.g., short to ground).
- **Aging:** No SOH degradation model.

6 Conclusion and Future Scope

6.1 Conclusion

Phase I has successfully established a high-fidelity, networked Zonal Architecture simulator. By rigorously modeling the vehicle plant and employing advanced signal conditioning ($\alpha = 0.1$) and covariance tuning ($R = 3000$), we have achieved a stable and accurate control system. The system effectively blends braking torque and estimates battery state, satisfying all "Network & Algorithm Upgrade" objectives.

6.2 Future Scope: Phase II (Migration to Hybrid Architectures)

As per the M.Tech research roadmap, the Zonal Architecture developed in Phase I will serve as the foundational **Digital Twin** for a **Hybrid Powertrain Control Unit**. Phase II will focus on:

1. **Platform Migration:** Porting the validated "Torque Arbitration" logic to a heterogeneous SoC (NXP i.MX8 or similar) running a **Type-1 Hypervisor**.
2. **Mixed-Criticality Integration:**
   - **RTOS Domain (Safety):** Will host the safety-critical **Energy Management Strategy (EMS)** for splitting torque between the **Internal Combustion Engine (ICE)** and the Electric Motor.
   - **Linux Domain (Performance):** Will host the "Fuel Efficiency & Diagnostics" dashboard.
3. **Why Hybrid?** The noise, vibration, and scheduling complexity of a Hybrid Powertrain necessitates the strict **Freedom From Interference (FFI)** provided by this Virtualized Zonal Architecture.

System Element	Phase I Implementation	Phase II Extension (Target)
Compute Platform	STM32F446RE (Cortex-M4)	NXP i.MX8 / RPi4 (Cortex-A + M)
OS Architecture	Single-Core Bare Metal	Dual-OS (Linux + FreeRTOS)
Control Focus	EV Blended Braking	Hybrid Torque Split (EMS)
Isolation	Physical (Separate PCBs)	Virtual (Type-1 Hypervisor)
Safety Goal	100Hz Control Loop	Freedom From Interference (FFI)

Table 2: Roadmap: Migration from Phase I Networked Control to Phase II Hybrid Zonal Architecture.