

Robust State Estimation via Extended Kalman Filtering for Autonomous Navigation

AV Perception Portfolio Team

January 24, 2026

Abstract

This report details the implementation and evaluation of an Extended Kalman Filter (EKF) for fusing Global Positioning System (GPS) and Inertial Measurement Unit (IMU) data. The objective is to estimate the 2D pose and kinematic state of an autonomous vehicle in the CARLA simulation environment. By addressing the non-linearities in vehicle motion and the stochastic nature of sensor noise, the proposed estimator achieves smooth trajectory tracking with significant noise rejection.

1 Introduction

State estimation is the foundation of autonomous mobility. While GNSS provides absolute localization, it is plagued by low sampling rates (typically 10 Hz) and measurement variance ($\sigma > 2m$). Conversely, proprioceptive sensors like IMUs offer high-bandwidth dynamics but suffer from unbounded integration drift. Determining the optimal posterior estimate of the vehicle state $\hat{\mathbf{x}}_k$ given all prior measurements $Z_{1:k}$ requires a sensor fusion framework. We employ the Extended Kalman Filter (EKF), which linearizes the system dynamics around the current estimate to propagate the first two moments of the belief state (mean μ and covariance Σ).

2 Methodology

2.1 State Space Formulation

We define the state vector $\mathbf{x} \in \mathbb{R}^6$ comprising the vehicle's planar position, body-frame velocities, yaw, and yaw rate:

$$\mathbf{x} = [p_x \ p_y \ v_x \ v_y \ \psi \ \omega]^T \quad (1)$$

where v_x represents the longitudinal velocity and v_y the lateral velocity (sideslip).

2.2 Process Model (Prediction)

The system evolves according to a non-linear Constant Turn Rate and Velocity (CTRV) model $\mathbf{x}_{k+1} = f(\mathbf{x}_k, \Delta t) + \mathbf{w}_k$. The transition function $f(\cdot)$ is defined as:

$$\begin{aligned} p_{x,k+1} &= p_{x,k} + (v_{x,k} \cos \psi_k - v_{y,k} \sin \psi_k) \Delta t \\ p_{y,k+1} &= p_{y,k} + (v_{x,k} \sin \psi_k + v_{y,k} \cos \psi_k) \Delta t \\ v_{x,k+1} &= v_{x,k} \quad (\text{Constant Body Velocity assumption}) \\ v_{y,k+1} &= v_{y,k} \\ \psi_{k+1} &= \psi_k + \omega_k \Delta t \\ \omega_{k+1} &= \omega_k \end{aligned} \quad (2)$$

Note: This model simplifies the dynamics by ignoring acceleration inputs, effectively treating velocity as a random walk driven by process noise $\mathbf{w}_k \sim \mathcal{N}(0, \mathbf{Q})$.

2.3 Measurement Model (Correction)

The sensor observations $\mathbf{z}_k \in \mathbb{R}^4$ map linearly to the state variables, though the EKF framework allows for non-linear observers $h(\mathbf{x})$.

$$\mathbf{z}_k = \begin{bmatrix} z_{GPS,x} \\ z_{GPS,y} \\ z_{IMU,\psi} \\ z_{IMU,\omega} \end{bmatrix} = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k \quad (3)$$

where \mathbf{H} is the observation matrix selecting $[p_x, p_y, \psi, \omega]$ and $\mathbf{v}_k \sim \mathcal{N}(0, \mathbf{R})$ is the measurement noise.

3 Implementation Details

The algorithm is implemented in Python utilizing the `carla` API. Major components include:

- **Jacobian Computation:** The system Jacobian $\mathbf{F} = \frac{\partial f}{\partial \mathbf{x}}$ is computed analytically at each step to linearize the uncertainty propagation.
- **Sensor Modeling:** Artificial Gaussian noise ($\sigma_{GPS} = 2.0m$) is injected into the ground truth signals to validate the filter's noise rejection capabilities.
- **Handling Asynchronous Data:** The filter operates on a ‘predict-update’ cycle triggered by simulator ticks, simplifying the time synchronization of heterogeneous sensors.

4 Results and Discussion

4.1 Performance Evaluation

Figure 1 illustrates the Estimator’s performance over a 1000-step trial.

- **Noise Rejection:** The fused estimate (Blue) effectively suppresses the high-frequency jitter of the raw GPS signal (Green).
- **Tracking Accuracy:** The EKF maintains a coherent trajectory even when measurement noise induces instantaneous errors of > 3 meters.

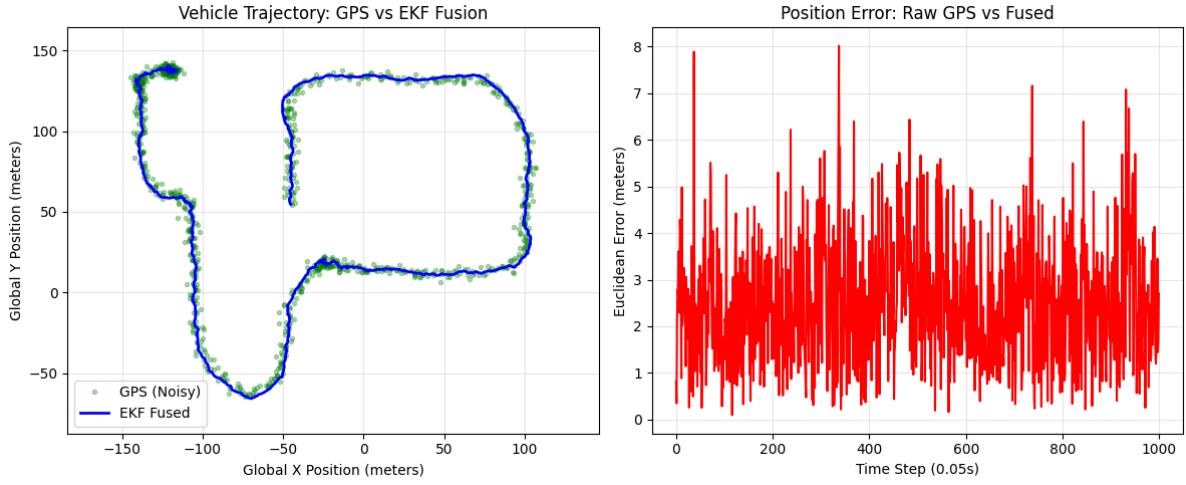


Figure 1: Trajectory Analysis: Comparing Raw GPS Observations (Noisy) vs. EKF Fused Estimate (Filtered).

4.2 Critical Analysis & Future Work

While the current implementation succeeds in basic state estimation, a rigorous evaluation highlights several limitations:

- 1. Absence of Accelerometer Integration:** The current prediction step uses a Constant Velocity model. Ideally, IMU accelerations (a_x, a_y) should be used as control inputs \mathbf{u}_k to drive the kinematic updates. This would reduce lag during rapid acceleration/braking events.
- 2. Sensor Bias:** Low-cost IMUs suffer from bias (b_ω, b_a). A production-grade estimator would augment the state vector to estimate and compensate for these biases online ($\mathbf{x}' = [\mathbf{x}, \mathbf{b}]^T$).
- 3. Observability:** Fusing absolute compass orientation prevents long-term yaw drift. However, in GPS-denied environments, the system would rely entirely on integration, necessitating the bias correction mentioned above.

5 Conclusion

The Week 1 project successfully established a sensor fusion pipeline in CARLA. The EKF implementation demonstrates the fundamental utility of Bayesian estimation in autonomous systems: transforming noisy, disparate sensor feeds into a unified, reliable state estimate.