NoisyTruth

Unscented Kalman Filter (UKF) — From Scratch Implementation

1 Introduction

Real-world systems are never perfectly observable. Sensors are noisy, models are imperfect, and uncertainty is unavoidable. This repository implements the **Unscented Kalman Filter (UKF)** from scratch to address the problem of reliable state estimation in nonlinear systems under noise.

2 Why Noise Exists in Sensor Models

In practice, no sensor measures the true state directly. Every measurement is corrupted by noise due to:

- Electronic noise (thermal, quantization)
- Environmental disturbances
- Sensor resolution limits
- Calibration errors
- Time delays and synchronization issues

A generic sensor measurement model is:

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{v}_k \tag{1}$$

where:

- $\mathbf{x}_k$ is the true system state
- $h(\cdot)$ is the (possibly nonlinear) measurement function
- $\mathbf{v}_k \sim \mathcal{N}(0, \mathbf{R})$ is the measurement noise

Similarly, system dynamics are modeled as:

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) + \mathbf{w}_{k-1} \tag{2}$$

where $\mathbf{w}_{k-1} \sim \mathcal{N}(0, \mathbf{Q})$ is process noise.

2.1 Why Sensor Readings Alone Are Not Enough

- Sensors provide instantaneous but noisy data
- Models provide smooth but imperfect predictions
- Neither alone is reliable

A Kalman filter provides a principled way to optimally combine both.

3 Kalman Filter Overview

The **Kalman Filter** is an optimal recursive estimator for linear systems with Gaussian noise.

3.1 Linear System Model

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_{k-1} + \mathbf{w}_{k-1} \tag{3}$$

$$\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k \tag{4}$$

4 Classical Kalman Filter Algorithm

4.1 Prediction Step

State prediction

$$\hat{\mathbf{x}}_k^- = \mathbf{A}\hat{\mathbf{x}}_{k-1} \tag{5}$$

Covariance prediction

$$\mathbf{P}_k^- = \mathbf{A}\mathbf{P}_{k-1}\mathbf{A}^T + \mathbf{Q} \tag{6}$$

4.2 Update Step

Kalman Gain

$$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{H}^T \left(\mathbf{H}\mathbf{P}_k^-\mathbf{H}^T + \mathbf{R}\right)^{-1} \tag{7}$$

State update

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_k^-) \tag{8}$$

Covariance update

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H})\mathbf{P}_k^- \tag{9}$$

5 Limitations of the Classical Kalman Filter

Most real-world systems are nonlinear:

- Robot motion models
- IMU and inertial sensors
- Radar and LiDAR measurements

Linear Kalman filters fail under strong nonlinearities, leading to biased estimates or divergence.

6 Extended Kalman Filter (EKF)

The **Extended Kalman Filter** addresses nonlinearity by linearizing the system using first-order Taylor expansion.

6.1 Jacobian Matrices

$$\mathbf{A}_k = \frac{\partial f}{\partial \mathbf{x}}, \quad \mathbf{H}_k = \frac{\partial h}{\partial \mathbf{x}} \tag{10}$$

6.2 Limitations of EKF

- Jacobians are difficult to derive
- Linearization errors accumulate
- Poor performance for highly nonlinear systems
- Can diverge easily

7 Unscented Kalman Filter (UKF)

The **Unscented Kalman Filter** avoids linearization entirely.

7.1 Key Idea: Unscented Transform

Instead of approximating nonlinear functions, UKF approximates the probability distribution by propagating carefully chosen sample points (sigma points).

8 Sigma Points

For a state dimension n, UKF generates $2n + 1$ sigma points.

$$\chi_0 = \mathbf{x} \tag{11}$$

$$\chi_i = \mathbf{x} + \sqrt{(n+\lambda)\mathbf{P}}_i \tag{12}$$

$$\chi_{i+n} = \mathbf{x} - \sqrt{(n+\lambda)\mathbf{P}}_i \tag{13}$$

where:

$$\lambda = \alpha^2(n+\kappa) - n \tag{14}$$

9 UKF Prediction Step

1. Generate sigma points
2. Propagate through system model:
$$\chi_i^- = f(\chi_i) \tag{15}$$
3. Compute predicted mean:
$$\hat{\mathbf{x}}^- = \sum_i W_i^m \chi_i^- \tag{16}$$
4. Compute predicted covariance:
$$\mathbf{P}^- = \sum_i W_i^c (\chi_i^- - \hat{\mathbf{x}}^-)(\chi_i^- - \hat{\mathbf{x}}^-)^T + \mathbf{Q} \tag{17}$$

10 UKF Measurement Update

1. Transform sigma points into measurement space:
$$\mathbf{z}_i = h(\chi_i^-) \tag{18}$$
2. Predicted measurement mean:
$$\hat{\mathbf{z}} = \sum_i W_i^m \mathbf{z}_i \tag{19}$$
3. Measurement covariance:
$$\mathbf{S} = \sum_i W_i^c (\mathbf{z}_i - \hat{\mathbf{z}})(\mathbf{z}_i - \hat{\mathbf{z}})^T + \mathbf{R} \tag{20}$$
4. Cross covariance:
$$\mathbf{P}_{xz} = \sum_i W_i^c (\chi_i^- - \hat{\mathbf{x}}^-)(\mathbf{z}_i - \hat{\mathbf{z}})^T \tag{21}$$
5. Kalman Gain:
$$\mathbf{K} = \mathbf{P}_{xz}\mathbf{S}^{-1} \tag{22}$$
6. State and covariance update:
$$\hat{\mathbf{x}} = \hat{\mathbf{x}}^- + \mathbf{K}(\mathbf{z} - \hat{\mathbf{z}}) \tag{23}$$
$$\mathbf{P} = \mathbf{P}^- - \mathbf{K}\mathbf{S}\mathbf{K}^T \tag{24}$$

11 EKF vs UKF Comparison

Feature	EKF	UKF
Linearization	Jacobians	None
Accuracy	First-order	Higher-order
Stability	Can diverge	More robust
Implementation	Complex	Cleaner

12 Applications

- Robotics and autonomous vehicles
- Sensor fusion
- Tracking and navigation
- Control systems
- Research and Development projects

13 Conclusion

This implementation provides a clear and faithful realization of the Unscented Kalman Filter by directly translating its mathematical formulation into readable Python code. The filter maintains a probabilistic state using a mean and covariance, generates sigma points via the Unscented Transform, propagates them through an arbitrary nonlinear system model, and reconstructs the predicted state without any linearization or Jacobians. The prediction and update steps strictly follow UKF theory, with explicit computation of weighted means, covariances, cross-covariances, and Kalman gain. Important features of the code include support for nonlinear dynamics through user-defined models, partial state measurements, configurable UKF tuning parameters, and thread-safe execution using locks. Overall, the design prioritizes correctness, transparency, and educational value, making the code suitable both for practical use and for developing a deep understanding of nonlinear state estimation.