Kalman Filter's Variants and Particle Filter

Approaches to Non-linear and Non-Gaussian Estimation Problems

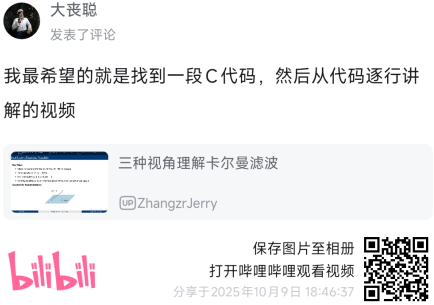
Zirui Zhang

Cheng Kar-Shun Robotics Institute
The Hong Kong University of Science and Technology

October 9, 2025

Kalman Filter in 3 Ways

The previous video has been well received. In response to fans' requests, we updated the Kalman filter visualization and unscented Kalman filter.



Kalman Filter Algorithm Steps

1. Prediction

$$\hat{x}_{k|k-1} = \mathbb{E}[x_k|z_{1:k-1}]$$
$$P_{k|k-1} = \mathrm{cov}[x_k - \hat{x}_{k|k-1}|z_{1:k-1}]$$

2. Update

$$\hat{x}_{k|k} = \mathbb{E}[x_k|z_{1:k}]$$
$$P_{k|k} = \mathrm{cov}[x_k - \hat{x}_{k|k}|z_{1:k}]$$

Nonlinear System Model

The Extended Kalman Filter (EKF) linearizes nonlinear system models using first-order Taylor series expansion around the current estimate, then applies standard Kalman Filter equations.

$$x_k = f(x_{k-1}, u_{k-1}, w_{k-1}), \qquad w_k \sim \mathcal{N}(0, Q_k)$$
$$z_k = h(x_k, v_k), \qquad v_k \sim \mathcal{N}(0, R_k)$$

Jacobian Matrices

$$F_{k-1} = \left.\frac{\partial f}{\partial x}\right|_{\hat{x}_{k-1|k-1}, u_{k-1}}$$
$$H_k = \left.\frac{\partial h}{\partial x}\right|_{\hat{x}_{k|k-1}}$$

EKF Algorithm Steps

1. Prediction Step

$$\hat{x}_{k|k-1} = f(\hat{x}_{k-1|k-1}, u_{k-1}, 0)$$
$$P_{k|k-1} = F_{k-1} P_{k-1|k-1} F_{k-1}^T + Q_k$$

2. Update Step

$$K_k = P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + R_k)^{-1}$$
$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(z_k - h(\hat{x}_{k|k-1}, 0))$$
$$P_{k|k} = (I - K_k H_k) P_{k|k-1}$$

Pros and Cons of the EKF

Pros

- Intuitive concept, relatively low computational cost
- Performs well in many applications

Cons

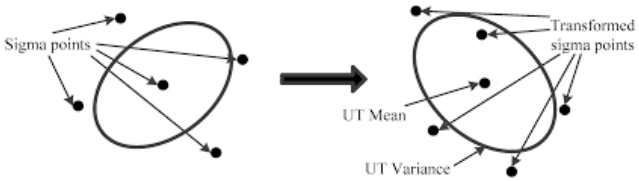
- Only suitable for **mildly nonlinear** systems
- Linearization errors can be large for strong nonlinearities
- Requires calculation of Jacobian matrices, which can be complex
- Can diverge due to accumulation of linearization errors

UKF - Different Approach

The Unscented Kalman Filter takes a different approach: Instead of approximating the nonlinear function, approximate the probability distribution.

Unscented Transform Steps

1. **Select Sigma Points**: Choose $2n+1$ points (n is state dimension)
2. **Propagate Sigma Points**: Pass each through nonlinear function f or h
3. **Recalculate Statistics**: Compute new mean and covariance



Courtesy: Zhenhui Zhang

Sigma Point Selection

For n-dimensional state vector $\hat{x}$ and covariance P:

$$\mathcal{X}^{(0)} = \hat{x}$$

$$\mathcal{X}^{(i)} = \hat{x} + \left(\sqrt{(n+\lambda)P}\right)_i, \qquad i = 1, \ldots, n$$

$$\mathcal{X}^{(i)} = \hat{x} - \left(\sqrt{(n+\lambda)P}\right)_{i-n}, \qquad i = n+1, \ldots, 2n$$

where $\lambda = \alpha^2(n+\kappa) - n$ is scaling parameter.

Weights

$$W_m^{(0)} = \frac{\lambda}{n+\lambda}$$

$$W_c^{(0)} = \frac{\lambda}{n+\lambda} + (1 - \alpha^2 + \beta)$$

$$W_m^{(i)} = W_c^{(i)} = \frac{1}{2(n+\lambda)}, \quad i = 1, \ldots, 2n$$

UKF Prediction Step

Prediction Process

- Generate Sigma points $\mathcal{X}_{k-1}^{(i)}$ from $(\hat{x}_{k-1|k-1}, P_{k-1|k-1})$
- Propagate through process model: $\mathcal{X}_{k|k-1}^{*(i)} = f(\mathcal{X}_{k-1}^{(i)}, u_{k-1})$

Compute Predicted Statistics

$$\hat{x}_{k|k-1} = \sum_{i=0}^{2n} W_m^{(i)} \mathcal{X}_{k|k-1}^{*(i)}$$

$$P_{k|k-1} = \sum_{i=0}^{2n} W_c^{(i)} \left(\mathcal{X}_{k|k-1}^{*(i)} - \hat{x}_{k|k-1}\right)\left(\mathcal{X}_{k|k-1}^{*(i)} - \hat{x}_{k|k-1}\right)^T + Q_k$$

UKF Update Step

Update Process

- Generate Sigma points $\mathcal{X}_{k|k-1}^{(i)}$ from $(\hat{x}_{k|k-1}, P_{k|k-1})$
- Propagate through observation model: $\mathcal{Z}_k^{(i)} = h(\mathcal{X}_{k|k-1}^{(i)})$

Observation Statistics

$$\hat{z}_{k|k-1} = \sum_{i=0}^{2n} W_m^{(i)} \mathcal{Z}_k^{(i)}$$

$$P_{zz} = \sum_{i=0}^{2n} W_c^{(i)} \left(\mathcal{Z}_k^{(i)} - \hat{z}_{k|k-1}\right)\left(\mathcal{Z}_k^{(i)} - \hat{z}_{k|k-1}\right)^T + R_k$$

$$P_{xz} = \sum_{i=0}^{2n} W_c^{(i)} \left(\mathcal{X}_{k|k-1}^{(i)} - \hat{x}_{k|k-1}\right)\left(\mathcal{Z}_k^{(i)} - \hat{z}_{k|k-1}\right)^T$$

UKF Update Step (cont.)

Kalman Gain and State Update

$$K_k = P_{xz} P_{zz}^{-1}$$
$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(z_k - \hat{z}_{k|k-1})$$
$$P_{k|k} = P_{k|k-1} - K_k P_{zz} K_k^T$$

Pros and Cons of the UKF

Pros

- No need to calculate Jacobian matrices - simpler implementation
- Higher approximation accuracy for nonlinear systems
- Generally more stable and less prone to divergence

Cons

- Slightly higher computational cost than EKF
- Parameters (α, β, κ) need tuning

Particles Approximation

Particle Filter uses random particles for sampling, unlike UKF's deterministic sigma points, enabling it to handle nonlinear, non-Gaussian systems via importance sampling and resampling:

$$p(x_k|z_{1:k}) \approx \sum_{i=1}^{N} w_k^{(i)} \delta(x_k - x_k^{(i)})$$

Where $\{x_k^{(i)}, w_k^{(i)}\}_{i=1}^{N}$ is the set of weighted particles, and $\delta(\cdot)$ is the Dirac delta function.

Particle Filter Algorithm Steps

1. Initialization

- Sample N particles $\{x_0^{(i)}\}_{i=1}^N$ from prior $p(x_0)$
- Initial weights $w_0^{(i)} = 1/N$

2. Recursive Process (each time step k)

- **Importance Sampling**: $x_k^{(i)} \sim p(x_k|x_{k-1}^{(i)})$
- **Weight Update**: $w_k^{(i)} = w_{k-1}^{(i)} \cdot p(z_k|x_k^{(i)})$
- **Weight Normalization**: $\tilde{w}_k^{(i)} = \frac{w_k^{(i)}}{\sum_{j=1}^N w_k^{(j)}}$
- **Resampling**: Replicate high-weight particles, eliminate low-weight ones

3. State Estimation

$$\hat{x}_k = \sum_{i=1}^N \tilde{w}_k^{(i)} x_k^{(i)}$$

Pros and Cons of the Particle Filter

Pros

- Can handle **highly nonlinear and non-Gaussian** systems
- Solid theoretical foundation (Bayesian estimation + Monte Carlo)
- Suitable for complex multi-modal distributions
- Relatively intuitive implementation

Cons

- **High computational cost**: Requires many particles
- **Particle degeneracy problem**: Few particles have significant weights
- **Sample impoverishment**: Loss of particle diversity after resampling

Comparison of Filter Methods

Feature	Kalman Filter	EKF	UKF	Particle Filter
Theoretical Basis	Optimal Linear Estimator	Local Linearization + KF	Unscented Transform + KF	Monte Carlo + Bayesian
Complexity	$O(n^3)$	$O(n^3)$	$O(n^3)$	$O(N)$, $N \gg n$
Applicable Systems	**Linear, Gaussian**	Mildly Nonlinear, Gaussian	Moderate to Highly Nonlinear, Gaussian	**Arbitrarily Complex Nonlinear, Non-Gaussian**
Memory	Low	Low	Low	High
Key Challenge	Limited to Linear Systems	Linearization Errors, Divergence	Parameter Tuning	**Particle Degeneracy**, Computational Cost

Visualization

Kalman Filter Visualization

- Interactive demonstration available at:
- `https://zhangzrjerry.github.io/html/kf.html`

Key Takeaways

- KF: Linear systems
- EKF: Mild nonlinearities, computational constraints
- UKF: Strong nonlinearities, no Jacobian preference
- PF: Highly nonlinear, non-Gaussian, complex distributions

Thank you for listening !

Zirui Zhang