Notes

October 22, 2025

Contents

1 State Space Models and Filtering Methods

1.1 Overview and Big Picture

Filtering methods estimate hidden states of dynamical systems from noisy measurements. The choice of filter depends on three key properties: **linearity of dynamics**, **process noise distribution**, and **measurement noise distribution**.

Key Insight: Process noise $\mathbf{w}_t$ and measurement noise $\mathbf{v}_t$ can have **different distributions** independently.

Linearity	Process Noise	Measurement Noise	Filter
Linear	Gaussian	Gaussian	Kalman (exact, optimal)
Nonlinear	Gaussian	Gaussian	EKF / UKF (approximate)
Any	Non-Gaussian	Gaussian	Particle / Hybrid
Any	Gaussian	Non-Gaussian	Particle / Robust Kalman
Any	Non-Gaussian	Non-Gaussian	Particle / Flow

Filtering Hierarchy:

- **Exact (Linear + Both Gaussian):** Kalman Filter
- **Approximate (Nonlinear + Both Gaussian):**
  - EKF: Linearization via Jacobian (1st-order)
  - UKF: Sigma point transform (3rd-order)
- **Sampling-based (At Least One Non-Gaussian):**
  - Standard Particle Filter: Stochastic sampling + resampling
  - Particle Flow Filters: Deterministic ODE evolution
  - Hybrid Methods: Rao-Blackwellized PF, Robust Kalman, Gaussian Sum

All methods solve the Bayesian filtering problem: compute $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ recursively.

1.2 General Framework

State space models (SSM) describe dynamical systems with hidden states observed through noisy measurements.

Evolution (Process Model):

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}) + \mathbf{w}_t, \quad \mathbf{w}_t \sim p_w(\cdot) \tag{1}$$

Measurement (Observation Model):

$$\mathbf{y}_t = h(\mathbf{x}_t) + \mathbf{v}_t, \quad \mathbf{v}_t \sim p_v(\cdot) \tag{2}$$

where $f(\cdot)$ is the evolution function, $h(\cdot)$ is the measurement function, $p_w(\cdot)$ is the **process noise distribution**, and $p_v(\cdot)$ is the **measurement noise distribution**. These two noise

distributions are **independent** and can be of different types (e.g., one Gaussian, one non-Gaussian).

Goal: Estimate hidden state $\mathbf{x}_t$ given measurements $\mathbf{y}_{1:t}$ by computing the posterior $p(\mathbf{x}_t|\mathbf{y}_{1:t})$.

1.3 Kalman Filter: Linear-Gaussian Case

For linear dynamics with **both** process and measurement noise Gaussian:

$$\mathbf{x}_t = \hat{\mathbf{F}}\mathbf{x}_{t-1} + \mathbf{w}_t, \quad \mathbf{w}_t \sim \mathcal{N}(0, \mathbf{Q}) \quad \text{(process noise)} \tag{3}$$

$$\mathbf{y}_t = \hat{\mathbf{H}}\mathbf{x}_t + \mathbf{v}_t, \quad \mathbf{v}_t \sim \mathcal{N}(0, \mathbf{R}) \quad \text{(measurement noise)} \tag{4}$$

where $\hat{\mathbf{F}}$ is the state transition matrix (evolution operator), $\hat{\mathbf{H}}$ is the measurement matrix, $\mathbf{Q}$ is the process noise covariance, and $\mathbf{R}$ is the measurement noise covariance.

The Kalman filter provides the **exact, closed-form** optimal solution when all conditions hold (linear + both Gaussian).

Notation:

- $\hat{\mathbf{x}}_t^-$ (also $\hat{\mathbf{x}}_{t|t-1}$, $\bar{\mathbf{x}}_t$, or $\mathbf{x}_t^f$) = *a priori* state estimate (prediction/forecast)
- $\hat{\mathbf{x}}_t^+$ (also $\hat{\mathbf{x}}_{t|t}$, $\hat{\mathbf{x}}_t$, or $\mathbf{x}_t^a$) = *a posteriori* state estimate (correction/analysis)
- $\mathbf{P}_t^-$ (also $\mathbf{P}_{t|t-1}$ or $\bar{\mathbf{P}}_t$) = *a priori* error covariance
- $\mathbf{P}_t^+$ (also $\mathbf{P}_{t|t}$ or $\mathbf{P}_t$) = *a posteriori* error covariance

Prediction Step:

$$\hat{\mathbf{x}}_t^- = \hat{\mathbf{F}}\hat{\mathbf{x}}_{t-1}^+ \tag{5}$$

$$\mathbf{P}_t^- = \hat{\mathbf{F}}\mathbf{P}_{t-1}^+\hat{\mathbf{F}}^T + \mathbf{Q} \tag{6}$$

Correction Step:

$$\hat{\mathbf{K}}_t = \mathbf{P}_t^-\mathbf{H}^T\left[\mathbf{H}\mathbf{P}_t^-\mathbf{H}^T + \mathbf{R}\right]^{-1} \quad \text{(Kalman gain)} \tag{7}$$

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

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

where $\mathbf{K}_t$ is the Kalman gain (optimal adaptive learning rate matrix) and $(\mathbf{y}_t - \mathbf{H}\hat{\mathbf{x}}_t^-)$ is the innovation (also called residual or measurement surprise).

The Kalman gain balances model prediction versus measurement based on their respective uncertainties: larger $\mathbf{R}$ (noisy sensors) reduces gain; larger $\mathbf{P}_t^-$ (uncertain prediction) increases gain.

Consider a linear-Gaussian state-space model at time t:

- State vector: $\mathbf{x}_t \in \mathbb{R}^n$, the hidden state.
- Observation vector: $\mathbf{z}_t \in \mathbb{R}^m$, the noisy measurement.

- Transition matrix: $F \in \mathbb{R}^{n \times n}$, linear dynamics operator.
- Observation matrix: $H \in \mathbb{R}^{m \times n}$, linear measurement operator.
- Process noise: $\mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, Q)$, with covariance $Q \in \mathbb{R}^{n \times n}$.
- Measurement noise: $\mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, R)$, with covariance $R \in \mathbb{R}^{m \times m}$.

The model equations are:

$$\mathbf{x}_t = F\mathbf{x}_{t-1} + \mathbf{v}_t, \tag{10}$$
$$\mathbf{z}_t = H\mathbf{x}_t + \mathbf{w}_t. \tag{11}$$

The predicted state (prior) given observations up to $t-1$ is:

- Mean: $\hat{\mathbf{x}}_{t|t-1} = F\hat{\mathbf{x}}_{t-1|t-1}$,
- Covariance: $P_{t|t-1} = FP_{t-1|t-1}F^\top + Q$.

Thus, the prior distribution is:

$$p(\mathbf{x}_t \mid \mathbf{z}_{1:t-1}) = \mathcal{N}(\hat{\mathbf{x}}_{t|t-1}, P_{t|t-1}).$$

The likelihood is:

$$p(\mathbf{z}_t \mid \mathbf{x}_t) = \mathcal{N}(H\mathbf{x}_t, R).$$

The posterior $p(\mathbf{x}_t \mid \mathbf{z}_{1:t}) \propto p(\mathbf{z}_t \mid \mathbf{x}_t)p(\mathbf{x}_t \mid \mathbf{z}_{1:t-1})$ is Gaussian. To derive the Kalman gain, consider the joint Gaussian distribution of $\mathbf{x}_t$ and $\mathbf{z}_t$ (conditional on $\mathbf{z}_{1:t-1}$):

$$\begin{bmatrix}\mathbf{x}_t \\ \mathbf{z}_t\end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix}\hat{\mathbf{x}}_{t|t-1} \\ H\hat{\mathbf{x}}_{t|t-1}\end{bmatrix}, \begin{bmatrix}P_{t|t-1} & P_{t|t-1}H^\top \\ HP_{t|t-1} & HP_{t|t-1}H^\top + R\end{bmatrix}\right).$$

For jointly Gaussian vectors $\mathbf{x}$ and $\mathbf{z}$ with means $\boldsymbol{\mu}_x, \boldsymbol{\mu}_z$, covariances Σ_{xx}, Σ_{zz}, and cross-covariance Σ_{xz}, the conditional distribution is:

$$\mathbf{x} \mid \mathbf{z} \sim \mathcal{N}\left(\boldsymbol{\mu}_x + \Sigma_{xz}\Sigma_{zz}^{-1}(\mathbf{z} - \boldsymbol{\mu}_z),\ \Sigma_{xx} - \Sigma_{xz}\Sigma_{zz}^{-1}\Sigma_{zx}\right).$$

Substitute:

- $\boldsymbol{\mu}_x = \hat{\mathbf{x}}_{t|t-1}$,
- $\boldsymbol{\mu}_z = H\hat{\mathbf{x}}_{t|t-1}$,
- $\Sigma_{xx} = P_{t|t-1}$,
- $\Sigma_{zz} = HP_{t|t-1}H^\top + R$,
- $\Sigma_{xz} = P_{t|t-1}H^\top$, $\Sigma_{zx} = HP_{t|t-1}$.

The posterior mean is:

$$\hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t|t-1} + P_{t|t-1}H^\top(HP_{t|t-1}H^\top + R)^{-1}(\mathbf{z}_t - H\hat{\mathbf{x}}_{t|t-1}). \tag{12}$$

Define the Kalman gain as:

$$K_t = P_{t|t-1}H^\top (HP_{t|t-1}H^\top + R)^{-1}.$$

The innovation (residual) is $\tilde{\mathbf{y}}_t = \mathbf{z}_t - H\hat{\mathbf{x}}_{t|t-1}$, so the update becomes:

$$\hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t|t-1} + K_t\tilde{\mathbf{y}}_t.$$

The posterior covariance is:

$$\begin{aligned} P_{t|t} &= P_{t|t-1} - P_{t|t-1}H^\top (HP_{t|t-1}H^\top + R)^{-1}HP_{t|t-1} \\ &= (I - K_tH)P_{t|t-1}, \end{aligned} \tag{13}$$

where I is the identity matrix. This Joseph stabilized form ensures numerical stability.

Intuition

The Kalman gain K_t balances the trust between the prior prediction $\hat{\mathbf{x}}_{t|t-1}$ (with uncertainty $P_{t|t-1}$) and the measurement $\mathbf{z}_t$ (with noise R). A large R (noisy measurement) reduces K_t, relying more on the prior; a large $P_{t|t-1}$ (uncertain prediction) increases K_t, favoring the measurement. This minimizes the trace of $P_{t|t}$, ensuring the minimum mean-squared error (MMSE) estimate.

1.4 Extended Kalman Filter (EKF)

When: Nonlinear f, h with **both** process and measurement noise Gaussian.

Strategy: Linearize via first-order Taylor expansion around current estimate.

Jacobians:

$$\mathbf{F}_t = \left.\frac{\partial f}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}^+_{t-1}}, \quad \mathbf{H}_t = \left.\frac{\partial h}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}^-_t} \tag{14}$$

Apply Kalman filter equations using time-varying $\mathbf{F}_t, \mathbf{H}_t$.

Prediction:

$$\hat{\mathbf{x}}^-_t = f(\hat{\mathbf{x}}^+_{t-1}) \tag{15}$$

$$\mathbf{P}^-_t = \mathbf{F}_t\mathbf{P}^+_{t-1}\mathbf{F}^T_t + \mathbf{Q} \tag{16}$$

Correction: Same as Kalman filter but with $\mathbf{H}_t$ and predicted measurement $\hat{\mathbf{y}}^-_t = h(\hat{\mathbf{x}}^-_t)$.

Limitations: Only first-order accurate; can diverge for strong nonlinearity; requires Jacobian computation (may be expensive or analytically intractable); assumes Gaussian noises.

1.5 Unscented Kalman Filter (UKF)

When: Nonlinear f, h with **both** noises Gaussian; better for strong nonlinearity than EKF.

Strategy: Propagate uncertainty through nonlinear functions using **sigma points** (also called unscented points) - deterministically chosen samples that capture mean and covariance.

Sigma Points: For state with mean $\hat{\mathbf{x}}$ and covariance $\mathbf{P}$, generate $2n+1$ points:

$$\mathcal{X}^{(0)} = \hat{\mathbf{x}} \tag{17}$$

$$\mathcal{X}^{(i)} = \hat{\mathbf{x}} + \left(\sqrt{(n+\lambda)\mathbf{P}}\right)_i, \quad i = 1, \ldots, n \tag{18}$$

$$\mathcal{X}^{(i)} = \hat{\mathbf{x}} - \left(\sqrt{(n+\lambda)\mathbf{P}}\right)_{i-n}, \quad i = n+1, \ldots, 2n \tag{19}$$

where $\mathcal{X}^{(i)}$ denotes the i-th sigma point, n is the state dimension, and λ is a scaling parameter.

Transform: Pass each sigma point through actual nonlinear function:

$$\mathcal{Y}^{(i)} = h(\mathcal{X}^{(i)}) \tag{20}$$

Reconstruct: Compute weighted mean and covariance of transformed points:

$$\hat{\mathbf{y}} = \sum_{i=0}^{2n} W^{(i)} \mathcal{Y}^{(i)} \tag{21}$$

$$\mathbf{P}_y = \sum_{i=0}^{2n} W^{(i)} (\mathcal{Y}^{(i)} - \hat{\mathbf{y}})(\mathcal{Y}^{(i)} - \hat{\mathbf{y}})^T \tag{22}$$

where $W^{(i)}$ are predetermined weights.

Advantages: No Jacobian needed (derivative-free); captures up to 3rd-order accuracy (vs. EKF's 1st-order); same $O(n^3)$ complexity as EKF.

Limitations: Still assumes both noises are Gaussian.

1.6 Particle Filter (Sequential Monte Carlo)

When: Arbitrary nonlinearity and/or **at least one** non-Gaussian noise; multimodal distributions.

Strategy: Represent posterior distribution using N weighted particles (samples):

$$p(\mathbf{x}_t | \mathbf{y}_{1:t}) \approx \sum_{i=1}^{N} w_t^{(i)} \delta(\mathbf{x}_t - \mathbf{x}_t^{(i)}) \tag{23}$$

where $\mathbf{x}_t^{(i)}$ is the i-th particle (sample state), $w_t^{(i)}$ is its weight, and $\delta(\cdot)$ is the Dirac delta function.

Algorithm (SIS: Sequential Importance Sampling with Resampling):

1. **Prediction (stochastic):** For each particle $i = 1, \ldots, N$:

$$\mathbf{x}_t^{(i)} = f(\mathbf{x}_{t-1}^{(i)}) + \mathbf{w}_t^{(i)}, \quad \mathbf{w}_t^{(i)} \sim p_w \tag{24}$$

Particles evolve by random sampling from the process noise distribution p_w (can be non-Gaussian).

2. **Update Weights:** Compute likelihood and normalize:

$$w_t^{(i)} \propto w_{t-1}^{(i)} \cdot p(\mathbf{y}_t | \mathbf{x}_t^{(i)}), \quad \sum_i w_t^{(i)} = 1 \tag{25}$$

where $p(\mathbf{y}_t|\mathbf{x}_t^{(i)})$ is the measurement likelihood (depends on p_v, can be non-Gaussian).

3. **Resampling (stochastic):** Resample particles with replacement according to weights to avoid degeneracy (also called particle depletion or weight collapse). High-weight particles get duplicated; low-weight particles die out.

4. **Estimate:** Compute weighted average:

$$\hat{\mathbf{x}}_t = \sum_{i=1}^{N} w_t^{(i)} \mathbf{x}_t^{(i)} \tag{26}$$

Advantages: Handles arbitrary nonlinearity; handles non-Gaussian noise in **either or both** process and measurement; can represent multimodal distributions; asymptotically exact as $N \to \infty$.

Limitations: Computationally expensive ($O(Nn^2)$); curse of dimensionality for high-dimensional states ($n > 10$); particle degeneracy issues despite resampling.

1.7 Particle Flow Filters (Advanced)

When: Similar to particle filters but with **deterministic** particle evolution to avoid degeneracy.

Key Difference: Standard particle filters use **stochastic** sampling and resampling (Monte Carlo). Particle flow filters move particles **deterministically** via continuous-time flow.

Strategy: Move particles via ODE from prior to posterior, rather than random sampling.

Homotopy Flow: Define parameter $\lambda \in [0, 1]$ where $\lambda = 0$ is prior $p(\mathbf{x}_t|\mathbf{y}_{1:t-1})$ and $\lambda = 1$ is posterior $p(\mathbf{x}_t|\mathbf{y}_{1:t})$.

Particles evolve via:

$$\frac{d\mathbf{x}^{(i)}}{d\lambda} = \mathbf{u}(\mathbf{x}^{(i)}, \lambda) \tag{27}$$

where $\mathbf{u}$ is a drift function designed to morph the prior into the posterior.

Example (Log-Homotopy):

$$p_\lambda(\mathbf{x}) \propto p(\mathbf{x}_t|\mathbf{y}_{1:t-1}) \cdot p(\mathbf{y}_t|\mathbf{x})^\lambda \tag{28}$$

As λ increases from 0 to 1, likelihood is gradually incorporated.

Common Methods:

- Daum-Huang Filter: Original exact ODE formulation
- Feedback Particle Filter: Control-theoretic approach
- Stein Variational Gradient Descent (SVGD): ML/variational inference perspective

Advantages: No particle degeneracy; no resampling needed; better for high-dimensional problems; smoother convergence; connects to optimal transport theory; handles non-Gaussian noises.

Limitations: Higher computational cost ($O(N^2n^2)$ per step); requires gradients $\nabla_{\mathbf{x}} \log p(\mathbf{y}_t|\mathbf{x})$; more complex implementation.

1.8 Hybrid and Special Methods

When noise types differ or have special structure, specialized methods can be more efficient than full particle filtering.

Rao-Blackwellized (Marginalized) Particle Filter: If state decomposes as $\mathbf{x}_t = [\mathbf{x}_t^{(1)}, \mathbf{x}_t^{(2)}]^T$ where part (2) is conditionally linear-Gaussian given part (1):

- Use particle filter for $\mathbf{x}_t^{(1)}$ (non-Gaussian part)
- Use Kalman filter for $\mathbf{x}_t^{(2)}|\mathbf{x}_t^{(1)}$ (Gaussian part)
- Much more efficient than full particle filtering

Robust Kalman Filter: For mostly Gaussian measurement noise with occasional outliers:

- Run standard Kalman filter
- Detect outliers via chi-squared test on innovation
- Reject or downweight suspicious measurements

Gaussian Sum Filter: For noise that is a mixture of Gaussians: $p(\mathbf{w}) = \sum_{k=1}^{K} \alpha_k \mathcal{N}(\mathbf{w}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$

- Run K parallel Kalman filters (one per mixture component)
- Combine outputs via weighted sum

1.9 Summary and Selection Guide

Filter	f, h	**Proc. Noise**	**Meas. Noise**	**Method**	**Complexity**	**Accuracy**
Kalman	Linear	Gaussian	Gaussian	Exact	$O(n^3)$	Optimal
EKF	Nonlinear	Gaussian	Gaussian	1st-order lin.	$O(n^3)$	Good (mild)
UKF	Nonlinear	Gaussian	Gaussian	Sigma points	$O(n^3)$	Better (strong)
Particle	Arbitrary	Arbitrary	Arbitrary	Stochastic MC	$O(Nn^2)$	Asymptotic
Flow	Arbitrary	Arbitrary	Arbitrary	Deterministic ODE	$O(N^2n^2)$	Asymptotic
Hybrid	Mixed	Mixed	Mixed	Combined	Varies	Problem-specific

Decision Tree:

- Linear f, h + both Gaussian → **Kalman Filter** (exact, optimal)
- Nonlinear f or h + both Gaussian:
  - Mild nonlinearity → **EKF** (fast, needs Jacobian)
  - Strong nonlinearity → **UKF** (better accuracy, derivative-free)
- At least one non-Gaussian noise:
  - Standard case or low-D ($n < 10$) → **Particle Filter**

- Special structure (part linear-Gaussian) → **Rao-Blackwellized PF**
- Occasional outliers only → **Robust Kalman**
- Mixture of Gaussians → **Gaussian Sum Filter**
- High-D or degeneracy issues → **Particle Flow**

Core Insight: All methods solve the same Bayesian filtering problem $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ using different computational strategies. The Kalman gain concept (optimal weighting of prediction vs. measurement based on uncertainty) underlies all approaches, appearing explicitly in Kalman/EKF/UKF and implicitly in particle methods. Process and measurement noises are independent and can have different distributions.