

Notes

November 12, 2025

Contents

1 State Space Models and Filtering Methods	2
1.1 Overview and Big Picture	2
1.2 General Framework	3
1.3 To add, discussion for noiseless case	3
1.4 Kalman Filter: Linear-Gaussian Case	3
1.5 Extended Kalman Filter (EKF)	5
1.6 Unscented Kalman Filter (UKF)	6
1.7 Particle Filter (Sequential Monte Carlo)	6
1.8 Particle Flow Filters (Advanced)	7
1.9 Hybrid and Special Methods	8
1.10 Summary and Selection Guide	8
2 Theoretical intuition	10
2.1 One-dimensional problems	10
3 Literature review for part one	19

1 State Space Models and Filtering Methods

1.1 Overview and Big Picture

Given the most general time series,

Evolution (Process Model):

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

Measurement (Observation Model):

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

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. where $p(x_t | y_{1:t})$ is the filtering distribution or posterior distribution of the hidden state at time t given all observations up to time t.

1.2 General Framework

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

Evolution (Process Model):

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}) + \mathbf{w}_t, \quad \mathbf{w}_t \sim p_w(\cdot) \quad (3)$$

Measurement (Observation Model):

$$\mathbf{y}_t = h(\mathbf{x}_t) + \mathbf{v}_t, \quad \mathbf{v}_t \sim p_v(\cdot) \quad (4)$$

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 To add, discussion for noiseless case

1.4 Kalman Filter: Linear-Gaussian Case

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

$$\mathbf{x}_t = \hat{\mathbf{F}}\mathbf{x}_{t-1} + \mathbf{w}_t, \quad \mathbf{w}_t \sim \mathcal{N}(0, \mathbf{Q}) \quad (\text{process noise}) \quad (5)$$

$$\mathbf{y}_t = \hat{\mathbf{H}}\mathbf{x}_t + \mathbf{v}_t, \quad \mathbf{v}_t \sim \mathcal{N}(0, \mathbf{R}) \quad (\text{measurement noise}) \quad (6)$$

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 (mean of the predicted distribution)
- $\hat{\mathbf{x}}_t^+$ (also $\hat{\mathbf{x}}_{t|t}$, $\hat{\mathbf{x}}_t$, or \mathbf{x}_t^a) = *a posteriori* state estimate (mean of the predicted distribution)
- \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:

$$\hat{\mathbf{x}}_t^- = \hat{\mathbf{F}}\hat{\mathbf{x}}_{t-1}^+ \quad (7)$$

$$\mathbf{P}_t^- = \hat{\mathbf{F}}\mathbf{P}_{t-1}^+\hat{\mathbf{F}}^T + \mathbf{Q} \quad (8)$$

Correction Step:

$$\hat{\mathbf{K}}_t = \mathbf{P}_t^- \mathbf{H}^T [\mathbf{H} \mathbf{P}_t^- \mathbf{H}^T + \mathbf{R}]^{-1} \quad (\text{Kalman gain}) \quad (9)$$

$$\hat{\mathbf{x}}_t^+ = \hat{\mathbf{x}}_t^- + \mathbf{K}_t (\mathbf{y}_t - \mathbf{H} \hat{\mathbf{x}}_t^-) \quad (10)$$

$$\hat{\mathbf{P}}_t^+ = (\mathbf{I} - \mathbf{K}_t \mathbf{H}) \mathbf{P}_t^- \quad (11)$$

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:

$$\mathbf{x}_t = F \mathbf{x}_{t-1} + \mathbf{v}_t, \quad (12)$$

$$\mathbf{z}_t = H \mathbf{x}_t + \mathbf{w}_t. \quad (13)$$

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: $P_{t|t-1} = F P_{t-1|t-1} F^\top + Q$.

Thus, the prior distribution is:

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

The likelihood is:

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

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

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

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:

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

Substitute:

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

The posterior mean is:

$$\hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t|t-1} + P_{t|t-1}H^\top(HP_{t|t-1}H^\top + R)^{-1}(\mathbf{z}_t - H\hat{\mathbf{x}}_{t|t-1}). \quad (14)$$

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:

$$\begin{aligned} P_{t|t} &= P_{t|t-1} - P_{t|t-1}H^\top(HP_{t|t-1}H^\top + R)^{-1}HP_{t|t-1} \\ &= (I - K_t H)P_{t|t-1}, \end{aligned} \quad (15)$$

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.

1.5 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:

$$\mathbf{F}_t = \left. \frac{\partial f}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_{t-1}^+}, \quad \mathbf{H}_t = \left. \frac{\partial h}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_t^-} \quad (16)$$

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

Prediction:

$$\hat{\mathbf{x}}_t^- = f(\hat{\mathbf{x}}_{t-1}^+) \quad (17)$$

$$\mathbf{P}_t^- = \mathbf{F}_t \mathbf{P}_{t-1}^+ \mathbf{F}_t^T + \mathbf{Q} \quad (18)$$

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.

1.6 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:

$$\mathcal{X}^{(0)} = \hat{\mathbf{x}} \quad (19)$$

$$\mathcal{X}^{(i)} = \hat{\mathbf{x}} + \left(\sqrt{(n + \lambda)\mathbf{P}} \right)_i, \quad i = 1, \dots, n \quad (20)$$

$$\mathcal{X}^{(i)} = \hat{\mathbf{x}} - \left(\sqrt{(n + \lambda)\mathbf{P}} \right)_{i-n}, \quad i = n + 1, \dots, 2n \quad (21)$$

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:

$$\mathcal{Y}^{(i)} = h(\mathcal{X}^{(i)}) \quad (22)$$

Reconstruct: Compute weighted mean and covariance of transformed points:

$$\hat{\mathbf{y}} = \sum_{i=0}^{2n} W^{(i)} \mathcal{Y}^{(i)} \quad (23)$$

$$\mathbf{P}_y = \sum_{i=0}^{2n} W^{(i)} (\mathcal{Y}^{(i)} - \hat{\mathbf{y}})(\mathcal{Y}^{(i)} - \hat{\mathbf{y}})^T \quad (24)$$

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.

1.7 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):

$$p(\mathbf{x}_t | \mathbf{y}_{1:t}) \approx \sum_{i=1}^N w_t^{(i)} \delta(\mathbf{x}_t - \mathbf{x}_t^{(i)}) \quad (25)$$

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 $i = 1, \dots, N$:

$$\mathbf{x}_t^{(i)} = f(\mathbf{x}_{t-1}^{(i)}) + \mathbf{w}_t^{(i)}, \quad \mathbf{w}_t^{(i)} \sim p_w \quad (26)$$

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

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

$$w_t^{(i)} \propto w_{t-1}^{(i)} \cdot p(\mathbf{y}_t | \mathbf{x}_t^{(i)}), \quad \sum_i w_t^{(i)} = 1 \quad (27)$$

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:

$$\hat{\mathbf{x}}_t = \sum_{i=1}^N w_t^{(i)} \mathbf{x}_t^{(i)} \quad (28)$$

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

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

1.8 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:

$$\frac{d\mathbf{x}^{(i)}}{d\lambda} = \mathbf{u}(\mathbf{x}^{(i)}, \lambda) \quad (29)$$

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

Example (Log-Homotopy):

$$p_\lambda(\mathbf{x}) \propto p(\mathbf{x}_t | \mathbf{y}_{1:t-1}) \cdot p(\mathbf{y}_t | \mathbf{x})^\lambda \quad (30)$$

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.

1.9 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

1.10 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 \rightarrow **Kalman Filter** (exact, optimal)
- Nonlinear f or h + both Gaussian:
 - Mild nonlinearity \rightarrow **EKF** (fast, needs Jacobian)
 - Strong nonlinearity \rightarrow **UKF** (better accuracy, derivative-free)
- At least one non-Gaussian noise:
 - Standard case or low-D ($n < 10$) \rightarrow **Particle Filter**
 - Special structure (part linear-Gaussian) \rightarrow **Rao-Blackwellized PF**
 - Occasional outliers only \rightarrow **Robust Kalman**
 - Mixture of Gaussians \rightarrow **Gaussian Sum Filter**
 - High-D or degeneracy issues \rightarrow **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.

2 Theoretical intuition

In this section, I want to lay out clear theoretical foundation for the problem at hand. Let us start with the most general case for a time series,

Evolution (Process Model):

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}) + \mathbf{w}_t, \quad \mathbf{w}_t \sim p_w(\cdot) \quad (31)$$

Measurement (Observation Model):

$$\mathbf{y}_t = h(\mathbf{x}_t) + \mathbf{v}_t, \quad \mathbf{v}_t \sim p_v(\cdot) \quad (32)$$

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).

2.1 One-dimensional problems

Suppose the observation is perfect, we ask given the distribution of \mathbf{x}_{t-1} , $\rho_{t-1}(x)$, how do we compute the distribution at t , namely ρ_t . Let us use one dimension as a demonstration,

$$\rho_t(x_t) = \int dw_t \ p_w(w_t) \int dx_{t-1} \ \rho_t(x_{t-1}) \delta(x_t - f(x_{t-1}) + w_t) \quad (33)$$

This is very difficult to compute for arbitrary function $f(\cdot)$ in high dimension, even without observation noise, due to the difficulty of convergence in high dimension. If somehow we are luck enough to deal with a system where we can expand $f(x_{t-1})$.

$$\begin{aligned} \rho_t(x_t) &= \int dw_t \ p_w(w_t) \int dx_{t-1} \ \rho_t(x_{t-1}) \delta(x_t - f(x_{t-1}) + w_t) \\ &= \int dw_t \ p_w(w_t) \int dx_{t-1} \ \rho_t(x_{t-1}) \delta(x_t - (1 + f_t)x_{t-1} + w_t) \\ &= \int dx_{t-1} \ \rho_t(x_{t-1}) p_w((1 + f_t)x_{t-1} - x_t) \end{aligned} \quad (34)$$

where $f_{t-1} = \partial_x f(x)|_{x_{t-1}}$. We can make the problem even easier, such that f becomes a constant. The problem is straightforward but difficult. However if we have imperfect observation, things become much more involved.

$$x_t = f x_{t-1} + w_t, \quad w_t \sim p_w(\cdot) \quad (35)$$

$$y_t = h x_t + v_t, \quad v_t \sim p_v(\cdot) \quad (36)$$

now the goal, is to, given the information about y_1, y_2, \dots, y_t , determine the probability $P(x_t|y_{1:t})$. The core idea of filtering method is,

$$\begin{aligned} p(x_t|y_{1:t-1}) &= \int dx_{t-1} \ p(x_t|x_{t-1}) p(x_{t-1}|y_{1:t-1}) \\ p(x_t|y_{1:t}) &= p(y_t|x_t) p(x_t|y_{1:t-1}) \end{aligned} \quad (37)$$

Now, this looks like not much more complicated than the case of perfect observation, namely the same number of integrals for certain discretization scheme.

Derivation of 1D Kalman Filter

State Evolution:

$$x_t = ax_{t-1} + w_t, \quad w_t \sim \mathcal{N}(0, Q) \quad (38)$$

Observation:

$$y_t = hx_t + v_t, \quad v_t \sim \mathcal{N}(0, R) \quad (39)$$

Initial Condition:

$$p(x_0) = \mathcal{N}(x_0 | m_0, P_0) \quad (40)$$

Goal Compute the filtering distribution $p(x_t | y_{1:t}) = \mathcal{N}(x_t | m_t, P_t)$ recursively.

Key Fact The filtering distribution remains Gaussian at all times. We only need to track the mean m_t and variance P_t .

Predict Step

Goal: Compute $p(x_t | y_{1:t-1})$ from $p(x_{t-1} | y_{1:t-1})$.

Given $p(x_{t-1} | y_{1:t-1}) = \mathcal{N}(x_{t-1} | m_{t-1}, P_{t-1})$, we have:

$$p(x_t | y_{1:t-1}) = \int dx_{t-1} p(x_t | x_{t-1}) p(x_{t-1} | y_{1:t-1}) \quad (41)$$

From the state evolution model:

$$p(x_t | x_{t-1}) = \mathcal{N}(x_t | ax_{t-1}, Q) \quad (42)$$

Computing the predicted mean:

$$m_t^- = \mathbb{E}[x_t | y_{1:t-1}] \quad (43)$$

$$= \mathbb{E}[\mathbb{E}[x_t | x_{t-1}, y_{1:t-1}] | y_{1:t-1}] \quad (44)$$

$$= \mathbb{E}[ax_{t-1} | y_{1:t-1}] \quad (45)$$

$$= am_{t-1} \quad (46)$$

Computing the predicted variance:

$$P_t^- = \text{Var}[x_t | y_{1:t-1}] \quad (47)$$

$$= \mathbb{E}[\text{Var}[x_t | x_{t-1}, y_{1:t-1}] | y_{1:t-1}] + \text{Var}[\mathbb{E}[x_t | x_{t-1}, y_{1:t-1}] | y_{1:t-1}] \quad (48)$$

$$= \mathbb{E}[Q | y_{1:t-1}] + \text{Var}[ax_{t-1} | y_{1:t-1}] \quad (49)$$

$$= Q + a^2 P_{t-1} \quad (50)$$

Result:

$$p(x_t | y_{1:t-1}) = \mathcal{N}(x_t | m_t^-, P_t^-) \quad (51)$$

where

$$\boxed{m_t^- = am_{t-1}, \quad P_t^- = a^2 P_{t-1} + Q} \quad (52)$$

Update Step

Goal: Incorporate observation y_t using Bayes' rule.

$$p(x_t | y_{1:t}) = \frac{p(y_t | x_t) p(x_t | y_{1:t-1})}{p(y_t | y_{1:t-1})} \quad (53)$$

From the observation model:

$$p(y_t|x_t) = \mathcal{N}(y_t|hx_t, R) \quad (54)$$

We need to compute the product of two Gaussians:

$$p(x_t|y_{1:t}) \propto \mathcal{N}(y_t|hx_t, R) \cdot \mathcal{N}(x_t|m_t^-, P_t^-) \quad (55)$$

$$\propto \exp\left(-\frac{1}{2}\frac{(y_t - hx_t)^2}{R}\right) \exp\left(-\frac{1}{2}\frac{(x_t - m_t^-)^2}{P_t^-}\right) \quad (56)$$

Expanding the exponents:

$$-\frac{1}{2} \left[\frac{(y_t - hx_t)^2}{R} + \frac{(x_t - m_t^-)^2}{P_t^-} \right] = -\frac{1}{2} \left[\frac{y_t^2 - 2y_t h x_t + h^2 x_t^2}{R} + \frac{x_t^2 - 2x_t m_t^- + (m_t^-)^2}{P_t^-} \right] \quad (57)$$

Collecting terms in x_t^2 and x_t :

$$= -\frac{1}{2} \left[\left(\frac{h^2}{R} + \frac{1}{P_t^-} \right) x_t^2 - 2 \left(\frac{hy_t}{R} + \frac{m_t^-}{P_t^-} \right) x_t + \text{const} \right] \quad (58)$$

This is a Gaussian in x_t . Completing the square:

Posterior variance:

$$\frac{1}{P_t} = \frac{h^2}{R} + \frac{1}{P_t^-} \quad (59)$$

$$P_t = \frac{1}{\frac{h^2}{R} + \frac{1}{P_t^-}} = \frac{RP_t^-}{h^2 P_t^- + R} \quad (60)$$

Define the **Kalman gain**:

$$K_t = \frac{P_t^- h}{h^2 P_t^- + R} \quad (61)$$

Then:

$$P_t = (1 - K_t h) P_t^- \quad (62)$$

Posterior mean:

$$\frac{m_t}{P_t} = \frac{hy_t}{R} + \frac{m_t^-}{P_t^-} \quad (63)$$

$$m_t = P_t \left(\frac{hy_t}{R} + \frac{m_t^-}{P_t^-} \right) \quad (64)$$

Substituting $P_t = (1 - K_t h) P_t^-$:

$$m_t = (1 - K_t h) P_t^- \left(\frac{hy_t}{R} + \frac{m_t^-}{P_t^-} \right) \quad (65)$$

$$= (1 - K_t h) \left(\frac{P_t^- hy_t}{R} + m_t^- \right) \quad (66)$$

$$= (1 - K_t h) m_t^- + (1 - K_t h) \frac{P_t^- hy_t}{R} \quad (67)$$

Using $K_t = \frac{P_t^- h}{h^2 P_t^- + R}$, we have $\frac{P_t^- h}{R} = K_t \frac{h^2 P_t^- + R}{R} = K_t(1 + \frac{h^2 P_t^-}{R})$.

After algebra:

$$m_t = m_t^- + K_t(y_t - hm_t^-) \quad (68)$$

Result:

$K_t = \frac{P_t^- h}{h^2 P_t^- + R}$ $m_t = m_t^- + K_t(y_t - hm_t^-)$ $P_t = (1 - K_t h)P_t^-$	(69)
--	------

Summary: 1D Kalman Filter Algorithm

Initialize: m_0, P_0

For $t = 1, 2, 3, \dots$:

Predict:

$$m_t^- = am_{t-1} \quad (70)$$

$$P_t^- = a^2 P_{t-1} + Q \quad (71)$$

Update:

$$K_t = \frac{P_t^- h}{h^2 P_t^- + R} \quad (72)$$

$$m_t = m_t^- + K_t(y_t - hm_t^-) \quad (73)$$

$$P_t = (1 - K_t h)P_t^- \quad (74)$$

Output: $p(x_t|y_{1:t}) = \mathcal{N}(x_t|m_t, P_t)$

From here it is trivial to generalize to higher dimension.

Stochastic Volatility Model (1D)

State Evolution:

$$x_t = \alpha x_{t-1} + \sigma w_t, \quad w_t \sim \mathcal{N}(0, 1) \quad (75)$$

Observation:

$$y_t = \beta \exp\left(\frac{x_t}{2}\right) v_t, \quad v_t \sim \mathcal{N}(0, 1) \quad (76)$$

Parameters:

- x_t = log-volatility (hidden state)
- y_t = observed returns
- $\alpha \in (0, 1)$ = persistence parameter
- σ = volatility of volatility
- β = scale parameter

Key Features:

- **Nonlinear observation:** $\exp(x_t/2)$ makes observation nonlinear in state
- **Non-Gaussian likelihood:** $p(y_t|x_t)$ is not Gaussian in x_t
- Linear state evolution + nonlinear/non-Gaussian observations \Rightarrow Kalman Filter fails

Range-Bearing Observation Model

A nonlinear tracking problem with polar observations.

State (Cartesian coordinates):

$$\mathbf{x}_t = \begin{bmatrix} p_x \\ p_y \\ v_x \\ v_y \end{bmatrix}_t \quad (77)$$

where (p_x, p_y) = position, (v_x, v_y) = velocity.

State Evolution (linear):

$$\mathbf{x}_t = \mathbf{F}\mathbf{x}_{t-1} + \mathbf{w}_t, \quad \mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}) \quad (78)$$

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (79)$$

Observation (nonlinear):

$$\mathbf{y}_t = \begin{bmatrix} r_t \\ \theta_t \end{bmatrix} = h(\mathbf{x}_t) + \mathbf{v}_t, \quad \mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{R}) \quad (80)$$

where:

$$r_t = \sqrt{p_x^2 + p_y^2} \quad (\text{range}) \quad (81)$$

$$\theta_t = \arctan(p_y, p_x) \quad (\text{bearing}) \quad (82)$$

Key Features:

- Highly nonlinear observation function (Cartesian \rightarrow polar transformation)
- Linear state dynamics but nonlinear measurement
- Standard benchmark for nonlinear filtering methods

Model Reminder:

$$x_t = \alpha x_{t-1} + \sigma w_t, \quad w_t \sim \mathcal{N}(0, 1) \quad (83)$$

$$y_t = \beta \exp\left(\frac{x_t}{2}\right) v_t, \quad v_t \sim \mathcal{N}(0, 1) \quad (84)$$

1. Extended Kalman Filter (EKF)

Idea: Linearize the nonlinear observation function around the current state estimate.

State Transition:

$$f(x_{t-1}) = \alpha x_{t-1}, \quad F_t = \frac{\partial f}{\partial x} \Big|_{x=m_{t-1}} = \alpha \quad (85)$$

Observation Function:

$$h(x_t) = \beta \exp\left(\frac{x_t}{2}\right) \cdot 0 = 0 \quad (\text{mean of observation given } x_t) \quad (86)$$

Wait, this needs clarification. The observation model is:

$$y_t|x_t \sim \mathcal{N}(0, \beta^2 \exp(x_t)) \quad (87)$$

So $h(x_t) = 0$ (mean) but variance depends on x_t .

For EKF, we linearize the observation model. Since $y_t = \beta \exp(x_t/2)v_t$, we have:

$$p(y_t|x_t) = \mathcal{N}(y_t|0, \beta^2 \exp(x_t)) \quad (88)$$

The log-likelihood is:

$$\log p(y_t|x_t) = -\frac{1}{2} \log(2\pi) - \frac{1}{2} \log(\beta^2 \exp(x_t)) - \frac{y_t^2}{2\beta^2 \exp(x_t)} \quad (89)$$

$$= -\frac{1}{2} \log(2\pi) - \frac{1}{2} \log(\beta^2) - \frac{x_t}{2} - \frac{y_t^2}{2\beta^2 \exp(x_t)} \quad (90)$$

For EKF, we linearize around the predicted mean m_t^- :

$$y_t \approx h(m_t^-) + H_t(x_t - m_t^-) + \text{noise} \quad (91)$$

where $h(x) = 0$ and

$$H_t = \frac{\partial h}{\partial x} \Big|_{x=m_t^-} = 0 \quad (92)$$

Problem: The standard EKF linearization doesn't work well here because the observation mean is always 0.

Alternative EKF Approach: Linearize the observation variance. We approximate:

$$\sigma_y^2(x_t) = \beta^2 \exp(x_t) \approx \beta^2 \exp(m_t^-) \cdot \exp(x_t - m_t^-) \quad (93)$$

Using first-order Taylor expansion: $\exp(x_t - m_t^-) \approx 1 + (x_t - m_t^-)$

This gives:

$$\sigma_y^2(x_t) \approx \beta^2 \exp(m_t^-)[1 + (x_t - m_t^-)] \quad (94)$$

EKF Algorithm:

Predict:

$$m_t^- = \alpha m_{t-1} \quad (95)$$

$$P_t^- = \alpha^2 P_{t-1} + \sigma^2 \quad (96)$$

Update:

The EKF update for this model is non-standard because the observation noise variance depends on the state. A common approximation is to use the predicted state to evaluate the observation noise:

$$R_t = \beta^2 \exp(m_t^-) \quad (97)$$

Then apply a modified update based on the innovation:

$$\nu_t = y_t^2 - \beta^2 \exp(m_t^-) \quad (\text{innovation in squared observation}) \quad (98)$$

The linearization of $\mathbb{E}[y_t^2 | x_t] = \beta^2 \exp(x_t)$ gives:

$$H_t = \beta^2 \exp(m_t^-) \quad (99)$$

$$S_t = H_t P_t^- H_t + \text{Var}[y_t^2] \approx 2\beta^4 \exp(2m_t^-) \quad (100)$$

$$K_t = P_t^- H_t S_t^{-1} \quad (101)$$

$$m_t = m_t^- + K_t \nu_t \quad (102)$$

$$P_t = (1 - K_t H_t) P_t^- \quad (103)$$

Note: EKF for stochastic volatility is notoriously poor because the linearization is inadequate for the exponential nonlinearity.

2. Unscented Kalman Filter (UKF)

Idea: Use sigma points to propagate mean and covariance through the nonlinearity without linearization.

Predict Step:

1. **Generate sigma points from** $p(x_{t-1} | y_{1:t-1}) = \mathcal{N}(m_{t-1}, P_{t-1})$:

For 1D, use 3 sigma points:

$$\chi_{t-1}^{(0)} = m_{t-1} \quad (104)$$

$$\chi_{t-1}^{(1)} = m_{t-1} + \sqrt{(1 + \lambda) P_{t-1}} \quad (105)$$

$$\chi_{t-1}^{(2)} = m_{t-1} - \sqrt{(1 + \lambda) P_{t-1}} \quad (106)$$

with weights:

$$W_0^{(m)} = \frac{\lambda}{1 + \lambda}, \quad W_0^{(c)} = \frac{\lambda}{1 + \lambda} + (1 - \alpha^2 + \beta) \quad (107)$$

$$W_i^{(m)} = W_i^{(c)} = \frac{1}{2(1 + \lambda)}, \quad i = 1, 2 \quad (108)$$

where $\lambda = \alpha^2(1 + \kappa) - 1$ (typical: $\alpha = 10^{-3}, \kappa = 0, \beta = 2$).

2. Propagate through state dynamics:

$$\chi_{t|t-1}^{(i)} = \alpha \chi_{t-1}^{(i)}, \quad i = 0, 1, 2 \quad (109)$$

3. Compute predicted mean and covariance:

$$m_t^- = \sum_{i=0}^2 W_i^{(m)} \chi_{t|t-1}^{(i)} \quad (110)$$

$$P_t^- = \sum_{i=0}^2 W_i^{(c)} (\chi_{t|t-1}^{(i)} - m_t^-)^2 + \sigma^2 \quad (111)$$

Update Step:

1. Generate sigma points from predicted distribution:

$$\chi_t^{(0)} = m_t^- \quad (112)$$

$$\chi_t^{(1)} = m_t^- + \sqrt{(1 + \lambda) P_t^-} \quad (113)$$

$$\chi_t^{(2)} = m_t^- - \sqrt{(1 + \lambda) P_t^-} \quad (114)$$

2. Propagate through observation function:

For stochastic volatility, the observation mean is 0, but we work with the variance. A common approach is to use the transformed observation $z_t = y_t^2$:

$$\gamma_t^{(i)} = \beta^2 \exp(\chi_t^{(i)}), \quad i = 0, 1, 2 \quad (115)$$

3. Compute predicted observation statistics:

$$\hat{z}_t = \sum_{i=0}^2 W_i^{(m)} \gamma_t^{(i)} \quad (116)$$

$$S_t = \sum_{i=0}^2 W_i^{(c)} (\gamma_t^{(i)} - \hat{z}_t)^2 + 2\beta^4 \exp(2m_t^-) \quad (117)$$

$$C_t = \sum_{i=0}^2 W_i^{(c)} (\chi_t^{(i)} - m_t^-)(\gamma_t^{(i)} - \hat{z}_t) \quad (118)$$

4. Kalman update:

$$K_t = C_t S_t^{-1} \quad (119)$$

$$m_t = m_t^- + K_t (y_t^2 - \hat{z}_t) \quad (120)$$

$$P_t = P_t^- - K_t S_t K_t^T \quad (121)$$

Note: UKF better captures the nonlinearity than EKF but still assumes the posterior remains approximately Gaussian.

3. Particle Filter

Idea: Represent the filtering distribution $p(x_t|y_{1:t})$ using weighted particles (samples).

Representation:

$$p(x_t|y_{1:t}) \approx \sum_{i=1}^N w_t^{(i)} \delta(x_t - x_t^{(i)}) \quad (122)$$

where $\{x_t^{(i)}, w_t^{(i)}\}_{i=1}^N$ are particles and weights, with $\sum_{i=1}^N w_t^{(i)} = 1$.

Sequential Importance Sampling (SIS):

Initialization ($t = 0$):

$$x_0^{(i)} \sim p(x_0), \quad i = 1, \dots, N \quad (123)$$

$$w_0^{(i)} = \frac{1}{N} \quad (124)$$

For $t = 1, 2, \dots$:

1. Predict (Propagate particles):

Sample from the state transition:

$$x_t^{(i)} = \alpha x_{t-1}^{(i)} + \sigma \epsilon^{(i)}, \quad \epsilon^{(i)} \sim \mathcal{N}(0, 1) \quad (125)$$

2. Update (Weight particles):

Compute importance weights based on the observation likelihood:

$$p(y_t|x_t^{(i)}) = \mathcal{N}(y_t|0, \beta^2 \exp(x_t^{(i)})) \quad (126)$$

$$= \frac{1}{\sqrt{2\pi\beta^2 \exp(x_t^{(i)})}} \exp\left(-\frac{y_t^2}{2\beta^2 \exp(x_t^{(i)})}\right) \quad (127)$$

Update weights:

$$\tilde{w}_t^{(i)} = w_{t-1}^{(i)} \cdot p(y_t|x_t^{(i)}) \quad (128)$$

3. Normalize weights:

$$w_t^{(i)} = \frac{\tilde{w}_t^{(i)}}{\sum_{j=1}^N \tilde{w}_t^{(j)}} \quad (129)$$

4. Resampling (when needed):

Check effective sample size:

$$N_{\text{eff}} = \frac{1}{\sum_{i=1}^N (w_t^{(i)})^2} \quad (130)$$

If $N_{\text{eff}} < N_{\text{threshold}}$ (e.g., $N/2$), resample:

- Draw N new particles $\{\bar{x}_t^{(i)}\}_{i=1}^N$ from $\{x_t^{(i)}\}_{i=1}^N$ with probabilities $\{w_t^{(i)}\}_{i=1}^N$

- Set $x_t^{(i)} = \bar{x}_t^{(i)}$ and $w_t^{(i)} = 1/N$ for all i

State Estimates:

Filtered mean:

$$\hat{x}_t = \sum_{i=1}^N w_t^{(i)} x_t^{(i)} \quad (131)$$

Filtered variance:

$$\hat{P}_t = \sum_{i=1}^N w_t^{(i)} (x_t^{(i)} - \hat{x}_t)^2 \quad (132)$$

Key Issues:

- **Particle degeneracy:** After several iterations, most weights become negligible
- **Sample impoverishment:** After resampling, many particles are duplicates
- **Curse of dimensionality:** Number of particles needed grows exponentially with state dimension

Advantages:

- No linearity or Gaussianity assumptions
- Asymptotically exact as $N \rightarrow \infty$
- Can handle arbitrary nonlinearities

3 Literature review for part one

3.1