Introduction to Stochastic Filtering

Simone Pavarana

University of Freiburg

July 9, 2025

Outline

Moving Object I

Goal: Estimate the true position of a moving object from noisy observations.

Model:

- **Signal (position evolution):**

$$X(t) = X(t-1) + \sigma_X \, \Delta W_X(t)$$

- **Observation:**

$$Y(t) = X(t) + \sigma_Y \, \Delta W_Y(t)$$

- $\Delta W_X(t), \Delta W_Y(t) \sim \mathcal{N}(0,1)$, i.i.d. noise

Interpretation: Position is subject to random motion, and observations are noisy.

Moving Object II

Goal: Estimate the position of a smoothly moving object, e.g., a vehicle.

Model:

- **Position dynamics:**

$$X(t) = X(t-1) + v(t-1) \cdot \Delta t$$

- **Velocity dynamics (mean-reverting):**

$$v(t) = v(t-1) - \lambda v(t-1) + \sigma_v \, \Delta W_v(t)$$

- **Observation:**

$$Y(t) = X(t) + \sigma_Y \, \Delta W_Y(t)$$

Use case: Tracking smoother motion (e.g., bicycles, cars, ships).

Stochastic Volatility (Continuous Time)

Goal: Estimate latent volatility from noisy price observations in continuous time.

Model:

- **Latent process (price dynamics):**

$$dX(t) = \sqrt{v(t)}\, dW_X(t)$$

- **Observed process (with microstructure noise):**

$$dY(t) = X(t)\, dt + \sigma_Y\, dW_Y(t)$$

- $v(t)$: latent Markov process (stochastic volatility)

Filtering task: Estimate $v(t)$ based on observed $Y(s), s \leq t$.

What is Filtering?

- **Objective:** Estimate the hidden state of a stochastic process based on partial and noisy observations.
- **Key References:**
  - Bain & Crisan, *Fundamentals of Stochastic Filtering* — continuous-time models with Brownian observations.
  - Liptser & Shiryaev, *Statistics of Random Processes*, Ch. 8 — classical continuous-time framework.
  - Brémaud, *Point Processes and Queues* — filtering with point process observations.
  - Cappé, Moulines & Rydén, *Inference in Hidden Markov Models* — discrete-time models.
- **Applications:**
  - Target tracking, signal processing, and robotics
  - Finance and Econometrics
  - Epidemiology, neuroscience, and social science

Discrete-Time Filtering: Setup

We work on a filtered probabaility space $(\Omega, \mathcal{F}, (\mathcal{F}_t)_{t\in\mathbb{N}}, P)$.
Goal: Estimate the hidden state $X(t)$ given noisy observations $Y(0), \ldots, Y(t)$.

Model: Hidden Markov Model (HMM)

- $(X_t)_{t\in\mathbb{N}}$: unobserved signal process (Markov)
- $(Y_t)_{t\in\mathbb{N}}$: observations (conditionally independent given X_t)
- Transition kernel of X: $p(x, dx') = P(X(t) \in dx' \mid X(t-1) = x)$
- Joint distribution:

$$P^{(X,Y)(t)|\mathcal{F}_{t-1}}(dx, dy) = K(x, dy) \cdot p(X(t-1), dx,$$

with $K(x, dy) = \lambda(x, y)\,\varphi(dy)$ (partially dominated).

Filtering Problem: Compute the conditional law $\pi_{t|t}(f) = \mathbb{E}[f(X(t)) \mid Y(0), \ldots, Y(t)]$.

Notation and Objectives

Notation:

- $y_{0:s} := (y_0, y_1, \ldots, y_s)$
- $\pi_{t|s}(y_{0:s}, f) := \mathbb{E}[f(X(t)) \mid Y(0:s) = y_{0:s}]$
- φ: reference measure on observation space

Objective: Compute $\pi_{t|t}(f)$ recursively using prediction and correction.

Idea: Introduce unnormalized measure
$\rho_{t|s}(f) = \mathbb{E}\left[f(X(t)) \prod_{r=0}^{s} \lambda(X(r), y_r)\right]$

$$\pi_{t|s}(f) = \frac{\rho_{t|s}(f)}{\rho_{t|s}(1)} \quad \text{(Kallianpur-Striebel formula)}$$

Recursive Computation (Unnormalized)

Let $f : \mathbb{R}^n \to \mathbb{R}$ be measurable and integrable. Define:

- **(1) Inception step:**

$$\rho_{0|0}(f) := \int f(x_0)\lambda(x_0, y_0)p_0(dx_0)$$

- **(2) Prediction step (for $t > s$):**

$$\rho_{t|s}(f) := \iint f(x_t)\, p(x_{t-1}, dx_t)\, \rho_{t-1|s}(dx_{t-1})$$

- **(3) Correction step (for $t = s$):**

$$\rho_{t|t}(f) := \int f(x_t)\lambda(x_t, y_t)\rho_{t|t-1}(dx_t)$$

Normalized filter: $\pi_{t|t}(f) = \dfrac{\rho_{t|t}(f)}{\rho_{t|t}(1)}$

Example: Finite State Space (Discrete HMM)

Setup:

- Hidden process $X_t \in \mathcal{A} = \{a_1, \ldots, a_k\}$: finite state Markov chain
- Transition matrix: $p_{ij} = \mathbb{P}(X_t = a_i \mid X_{t-1} = a_j)$
- Emission density: $\lambda(a_i, y_t)$ w.r.t. fixed measure φ

Recursions for the filter $\pi_{t|t}(i) = \mathbb{P}(X_t = a_i \mid Y_{0:t} = y_{0:t})$:

- **Inception:**

$$\pi_{0|0}(i) = \frac{\lambda(a_i, y_0)\, p_0(i)}{\sum_{j=1}^{k} \lambda(a_j, y_0)\, p_0(j)}$$

- **Prediction:**

$$\pi_{t|t-1}(i) = \sum_{j=1}^{k} p_{ij}\, \pi_{t-1|t-1}(j)$$

- **Correction:**

$$\pi_{t|t}(i) = \frac{\lambda(a_i, y_t)\, \pi_{t|t-1}(i)}{\sum_{j=1}^{k} \lambda(a_j, y_t)\, \pi_{t|t-1}(j)}$$

Linear Gaussian State-Space Model

Model:

$$X(t) = a_X + A_X X(t-1) + B_X Z_X(t)$$
$$Y(t) = a_Y + A_Y X(t) + B_Y Z_Y(t)$$

- $Z_X(t), Z_Y(t) \sim \mathcal{N}(0, I)$ i.i.d. and independent
- Initial state $X(0) \sim \mathcal{N}(\mu_0, \Sigma_0)$

Goal: Estimate $\mu_{t|t} = \mathbb{E}[X(t) \mid Y_{0:t}]$, $\Sigma_{t|t} = \mathrm{Var}(X(t) \mid Y_{0:t})$

Kalman Filter: Recursive Equations

Start with:

$$\mu_{0|0} = \mu_0, \quad \Sigma_{0|0} = \Sigma_0$$

Prediction step (for $t > 0$):

$$\mu_{t|t-1} = a_X + A_X \mu_{t-1|t-1}$$
$$\Sigma_{t|t-1} = A_X \Sigma_{t-1|t-1} A_X^\top + B_X B_X^\top$$

Correction step:

$$K_t = \Sigma_{t|t-1} A_Y^\top \left(A_Y \Sigma_{t|t-1} A_Y^\top + B_Y B_Y^\top \right)^{-1} \quad \text{(Kalman gain)}$$
$$\mu_{t|t} = \mu_{t|t-1} + K_t \left(Y(t) - a_Y - A_Y \mu_{t|t-1} \right)$$
$$\Sigma_{t|t} = \Sigma_{t|t-1} - K_t A_Y \Sigma_{t|t-1}$$

Continuous-time Filtering: Mathematical Setup

Signal Process:

- The signal $X = (X_t)_{t \geq 0}$ is an S-valued process defined on a filtered probability space $(\Omega, \mathcal{F}, (\mathcal{F}_t), \mathbb{P})$.
- It solves the **martingale problem** for a linear operator $\mathcal{A}$ on a domain $\mathcal{D}(\mathcal{A}) \subset C_b(S)$:

$$M_t^{\varphi} := \varphi(X_t) - \varphi(X_0) - \int_0^t \mathcal{A}\varphi(X_s)\, ds$$

  is a martingale for every $\varphi \in \mathcal{D}(\mathcal{A})$.

Observation Process:

- The observed process $Y = (Y_t) \in \mathbb{R}^n$ is given by:

$$Y_t = \int_0^t h(X_s)\, ds + W_t$$

  where $h : S \to \mathbb{R}^n$ is measurable, and W is an n-dimensional Brownian motion independent of X.

Filtering Problem and Methods

Filtering Problem:

Given the observation filtration

$$\mathcal{Y}_t := \sigma(Y_s : 0 \leq s \leq t) \vee \mathcal{N}$$

compute the conditional distribution π_t of X_t, such that:

$$\mathbb{E}[\varphi(X_t) \mid \mathcal{Y}_t] = \int \varphi(x)\, \pi_t(dx)$$

for all bounded measurable $\varphi : S \to \mathbb{R}$.

Two Main Derivation Approaches:

- **Change of Measure Method:** A new measure is constructed under which Y becomes a Brownian motion independent on X.
- **Innovation Approach:** It isolates the Brownian motion driving the evolution equation of π (called the innovation process).

Zakai Equation and Normalization

Zakai Equation
Let $\varphi \in \mathcal{D}(\mathcal{A})$. Under suitable conditions, the unnormalized conditional distribution ρ_t satisfies:

$$\rho_t(\varphi) = \pi_0(\varphi) + \int_0^t \rho_s(\mathcal{A}\varphi)\, ds + \int_0^t \rho_s(\varphi h^\top)\, dY_s, \quad \widetilde{\mathbb{P}} - a.s.$$

Normalization (Kallianpur-Striebel Formula):

The normalized filter π_t is given by:

$$\pi_t(\varphi) = \frac{\rho_t(\varphi)}{\rho_t(1)}, \quad \mathbb{P}(\widetilde{\mathbb{P}}) - a.s.$$

Interpretation: The Zakai equation is linear but unnormalized. Normalization is required to recover the conditional law of the signal process.

Kushner–Stratonovich Equation

Theorem 3.30: Under mild conditions, the normalized conditional distribution π_t satisfies:

$$\pi_t(\varphi) = \pi_0(\varphi) + \int_0^t \pi_s(A\varphi)\, ds + \int_0^t \left(\pi_s(\varphi h^\top) - \pi_s(\varphi)\pi_s(h^\top)\right)\, dI_s$$

where

$$I_t = Y_t - \int_0^t \pi_s(h)\, ds, \quad \text{(innovation process)}$$

Interpretation:

- Nonlinear SPDE for the posterior measure π_t
- Derived via innovation approach or by normalizing Zakai

Kalman–Bucy Filter: Setup

Signal Process:

$$dX_t = F_t X_t \, dt + f_t \, dt + \sigma_t \, dV_t$$

- $X_t \in \mathbb{R}^d$, $V_t \in \mathbb{R}^p$ is standard Brownian motion
- F_t: $d \times d$ matrix, σ_t: $d \times p$ matrix
- f_t: drift vector; all coefficients measurable and locally bounded
- Initial state: $X_0 \sim \mathcal{N}(x_0, R_0)$, independent of V

Observation Process:

$$dY_t = H_t X_t \, dt + h_t \, dt + dW_t$$

- $Y_t \in \mathbb{R}^m$, $W_t \in \mathbb{R}^m$ is Brownian motion independent of V
- H_t: $m \times d$ matrix, $h_t \in \mathbb{R}^m$

Kalman–Bucy Filtering Equations

Posterior Mean (Conditional Expectation):

$$d\hat{x}_t = (F_t \hat{x}_t + f_t)\, dt + R_t H_t^\top \left(dY_t - (H_t \hat{x}_t + h_t)\, dt \right)$$

Posterior Covariance:

$$\frac{d}{dt} R_t = \sigma_t \sigma_t^\top + F_t R_t + R_t F_t^\top - R_t H_t^\top H_t R_t, \quad \text{(Riccati equation)}$$

Interpretation:

- $\hat{x}_t = \mathbb{E}[X_t \mid \mathcal{Y}_t]$, $R_t = \mathrm{Cov}(X_t \mid \mathcal{Y}_t)$
- $\hat{x}_t$ is updated online using new data; R_t is deterministic and can be computed offline.

Extended Kalman Filter (EKF): Setup and Linearization

Nonlinear state-space model (Eq. 8.5):

$$dX_t = f(X_t)\,dt + \sigma(X_t)\,dV_t + g(X_t)\,dW_t$$
$$dY_t = h(X_t)\,dt + dW_t$$

- V_t, W_t: independent Brownian motions
- X_t: signal, Y_t: observation

First-order (Taylor) approximation:

$$dX_t \approx f(\bar{x}_t)\,dt + f'(\bar{x}_t)(X_t - \bar{x}_t)\,dt + \sigma(\bar{x}_t)\,dV_t + g(\bar{x}_t)\,dW_t$$
$$dY_t \approx h(\bar{x}_t)\,dt + h'(\bar{x}_t)(X_t - \bar{x}_t)\,dt + dW_t$$

Idea: Locally linearize the system around a deterministic trajectory $\bar{x}_t$ solving $d\bar{x}_t = f(x_t)\,dt$.

Extended Kalman Filter Equations

Idea: Given the linearized system, we can apply the Kalman–Bucy filter. More generally, we may consider any $\mathcal{Y}_t$-adapted estimator m_t, and define a mapping:

$$\Lambda : m_t \mapsto \hat{x}_t$$

where $\hat{x}_t$ is the Kalman–Bucy estimate based on linearization around m_t. The **Extended Kalman Filter** is the *fixed point* of this mapping: $\Lambda(\hat{x}_t) = \hat{x}_t$.

EKF Update Equations:

$$\begin{aligned} d\hat{x}_t &= (f - gh)(\hat{x}_t)\,dt + g(\hat{x}_t)\,dY_t + R_t h'(\hat{x}_t)^\top \left[dY_t - h(\hat{x}_t)\,dt\right] \\ \frac{dR_t}{dt} &= (f' - gh')(\hat{x}_t)R_t + R_t(f' - gh')^\top(\hat{x}_t) + \sigma\sigma^\top(\hat{x}_t) \\ &\quad - R_t h'^\top(\hat{x}_t)h'(\hat{x}_t)R_t \end{aligned}$$

Initialization: $\hat{x}_0 = x_0$, $R_0 = p_0$

Particle Filtering: Basic Idea

Objective: Approximate the conditional distribution π_t of the signal process given observations $Y_{[0,t]}$.

Unnormalized formulation (Kallianpur–Striebel):

$$\pi_t(\varphi) = \frac{\rho_t(\varphi)}{\rho_t(1)}, \quad \rho_t(\varphi) = \widetilde{\mathbb{E}}\left[\varphi(X_t)\widetilde{Z}_t \mid Y_t\right]$$

Motivation:

- Particle filtering is the most widely used nonlinear filtering method in practice.
- It applies to general (nonlinear, non-Gaussian) models.
- Particularly effective when Gaussian approximations like EKF fail.

Particle Filtering: Monte Carlo Approximation

Simulation-based approximation:

- Simulate n independent particle paths $\{v_j(t)\}_{j=1}^n \sim X_t$
- Compute the exponential martingales:

$$a_j(t) = \exp\left(\int_0^t h(v_j(s))^\top \, dY_s - \frac{1}{2}\int_0^t \|h(v_j(s))\|^2 ds\right)$$

- Approximate the unnormalized conditional expectation:

$$\rho_t^n(\varphi) = \frac{1}{n}\sum_{j=1}^n \varphi(v_j(t)) \cdot a_j(t)$$

- Normalize:

$$\pi_t^n(\varphi) = \frac{\rho_t^n(\varphi)}{\rho_t^n(1)}$$

Result: Empirical measure from weighted particles approximates π_t.

Other Numerical Approaches

Beyond EKF and Particle Filters:

- **Finite-dimensional nonlinear filters:** Rare cases where the filtering equations close in finite dimension (e.g., Beneš filter).
- **Projection filters and moment methods:** Approximate the filter within a finite-dimensional manifold using ideas from differential geometry and information geometry.
- **Spectral methods:** Represent the filtering distribution via eigenfunction expansions of the signal generator.
- **PDE-based methods:** Numerically solve the Zakai or Kushner–Stratonovich equations using finite differences, finite elements, or splitting schemes.
- **Affine filters:** For affine signal models, the filtering problem reduces to solving stochastic Riccati equations. *(See: Gonon & Teichmann, "Linearized filtering of affine processes using stochastic Riccati equations", 2021)*

What is a Transformer?

Transformer = Neural Network Architecture for Sequential Data

Introduced by: Vaswani et al., *"Attention Is All You Need"* (2017)

Key Features:

- Based on **self-attention** mechanism — models dependencies between all positions in a sequence.
- Processes input **in parallel** (unlike RNNs), enabling efficient training.
- **Highly scalable** and adaptable to many tasks.

Architecture Highlights:

- Encoder–Decoder structure (or stacked encoders for e.g. BERT).
- Each layer contains: Multi-head attention + Feed-forward network.
- Positional encodings added to input to retain sequence order.

Applications:

- NLP: GPT, BERT, T5, ChatGPT
- Time series prediction, speech recognition, protein folding
- More recently: **stochastic filtering** and state estimation

Transformer Encoder: Step-by-Step

Input: sequence of token embeddings $X = [x_1, \ldots, x_n] \in \mathbb{R}^{n \times d}$

1. Add Positional Encoding

$$Z^{(0)} = X + P \quad \text{with} \quad P \in \mathbb{R}^{n \times d}$$

2. Multi-Head Self-Attention (MHA)

$$\text{head}_i = \text{softmax}\left(\frac{Q_i K_i^\top}{\sqrt{d_h}}\right) V_i \quad \text{where} \quad Q_i = ZW_i^Q,\ K_i = ZW_i^K,\ V_i = ZW_i^V$$

$$\text{MultiHead}(Z) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h) W^O$$

3. Residual Connection + Layer Norm

$$Z' = \text{LayerNorm}(Z + \text{MultiHead}(Z))$$

Transformer Encoder: Feedforward and Output

4. Feedforward Neural Network (FFN)

$$\mathrm{FFN}(z') = \sigma(z' W_1 + b_1) W_2 + b_2 \quad \text{(applied to each token independently)}$$

5. Residual Connection + Layer Norm

$$Z^{(1)} = \mathrm{LayerNorm}(Z' + \mathrm{FFN}(Z'))$$

Final Output: $Z^{(1)} \in \mathbb{R}^{n \times d}$ is passed to the next encoder layer

Transformer Encoder = Stack of L such layers

$$Z^{(L)} = \mathrm{Encoder}_L \circ \cdots \circ \mathrm{Encoder}_1(Z^{(0)})$$

All weights ($W^Q, W^K, W^V, W^O, W_1, W_2, P$) are trainable.

Transformer Architecture

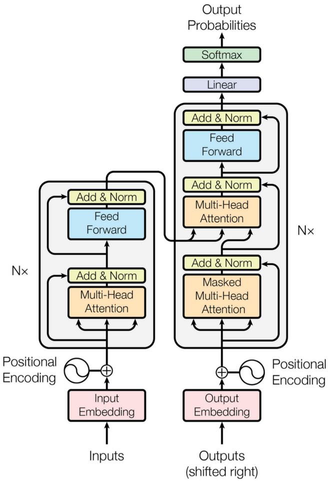

Figure 1: The Transformer - model architecture.

Can a Transformer Represent a Kalman Filter?

Reference: G. Goel and P. Bartlett, *Can a Transformer Represent a Kalman Filter?*, (2024)

Kalman Filter Model:

$$x_{t+1} = Ax_t + w_t, \quad y_t = Cx_t + v_t, \quad \hat{x}^\star_t = (A - LC)\hat{x}^\star_{t-1} + Ly_{t-1}$$

- $A \in \mathbb{R}^{n \times n}$, $C \in \mathbb{R}^{p \times n}$: system and observation matrices
- L: Kalman gain (fixed)
- $\hat{x}^\star_t$: Kalman estimate at time t

Main Question: Can a Transformer, despite its nonlinear structure, approximate the Kalman filter uniformly in time?

Transformer Approximation of the Kalman Filter

Theorem: For every $\varepsilon > 0$, there exists a Transformer Filter such that:

$$\|\hat{x}_t - \hat{x}_t^\star\| \leq \varepsilon \quad \text{for all } t \geq 0$$

where $\hat{x}_t$ is the state estimate produced by the Transformer Filter. **Key Elements:**

- H: number of past estimates and observations used in attention (temporal window size)
- The Transformer estimate is given by a weighted average:

$$\hat{x}_t = \sum_{i=t-H+1}^{t} \alpha_{i,t}(\beta)\, \tilde{x}_i$$

- $\tilde{x}_i$: pseudo-Kalman updates — i.e., the estimates that would be produced by the Kalman recursion if $\hat{x}_{i-1}^\star$ were replaced with $\hat{x}_{i-1}$

Conclusion: A Transformer with softmax attention over H past steps can uniformly approximate Kalman filtering in time, with arbitrarily small error.

Paper:
Horvath, B., Kratsios, A., Limmer, Y., & Yang, X. (2023). "Transformers Can Solve Non-Linear and Non-Markovian Filtering Problems in Continuous Time For Conditionally Gaussian Signals".

State-Space Model:

$$dX_t = [a_0(t, Y_{[0:t]}) + a_1(t, Y_{[0:t]})X_t]dt + \sum_{i=1}^{2} b_i(t, Y_{[0:t]})dW_t^{(i)} \qquad \text{(Signal)}$$

$$dY_t = [A_0(t, Y_{[0:t]}) + A_1(t, Y_{[0:t]})X_t]dt + \sum_{i=1}^{2} B_i(t, Y_{[0:t]})dW_t^{(i)} \qquad \text{(Observation)}$$

where $W^{(1)}, W^{(2)}$ are independent Brownian motions.

Informal Theorem 1 (Universal Conditionally-Gaussian Filtering):
For conditionally Gaussian signal processes (X_t, Y_t) satisfying mild regularity conditions, there exists a Filterformer $\hat{F}$ such that:

$$\max_{0\leq t\leq T,\, y\in K} \mathcal{W}_p\big(\mathbb{P}(X_t \in \cdot | y_{[0:t]}), \hat{F}(t, y)\big) < \varepsilon$$

uniformly over compact path sets K, for any $\varepsilon > 0$ and $1 \leq p \leq 2$.

Summary and Key Takeaways

- **Stochastic filtering** addresses the estimation of latent states from noisy and partial observations — a core problem in statistics, control theory, and machine learning.
- **Discrete-time filtering** is formulated as a recursive update (e.g., HMMs, Kalman Filter), while
- **Continuous-time filtering** involves stochastic PDEs (Zakai, Kushner–Stratonovich), often analytically and numerically challenging.
- **Exact solutions** are available only in special cases (linear-Gaussian models, finite state spaces), motivating approximate methods.
- **Numerical methods** such as the Extended Kalman Filter, Particle Filters, and PDE solvers provide tractable solutions in nonlinear and non-Gaussian settings.
- **Transformer-based models** offer a powerful and flexible alternative: they can approximate classical filters (e.g., Kalman), and — crucially — handle **nonlinear, non-Markovian** filtering problems with provable accuracy, opening new directions for data-driven state estimation.