Kalman Filters and Extensions

Authors: Matvei Kreinin, Maria Nikitina, Anastasia Voznyuk, Petr Babkin
Course: Bayesian Multimodelling 2024 - 2025

1 Bringing Kalman Filters to Deep Learning

Despite their foundational role in probabilistic state-space modeling, Kalman filters remain surprisingly underrepresented in the deep learning community. Our new framework aims to change that.

Kalman filters serve as the backbone for many advanced models, including:

- Gaussian Processes (GPs) – used for regression and time-series analysis.
- State-Space Models (SSMs) like S4/S6 – crucial for sequence modeling in modern deep learning.
- Hidden Markov Models (HMMs) – widely applied in speech recognition and bioinformatics.
- Recurrent Neural Networks (RNNs) – which share conceptual similarities in sequential data processing.

1.1 Why Should DL Researchers Care?

Kalman filters provide a principled way to estimate hidden states from noisy observations—something neural networks often do implicitly but with less interpretability. By integrating these filters into DL pipelines, we gain:

- Stronger theoretical grounding – Kalman-based approaches have well-defined probabilistic properties.
- Better uncertainty quantification – Unlike many DL models, Kalman filters explicitly model uncertainty.
- Efficiency interpretability – Compared to deep learning, Kalman filters can be computationally efficient and easier to analyze.

2 Theoretical Background

The Kalman filter is a recursive algorithm that estimates the hidden state of a dynamic system based on noisy observations. It provides an optimal solution (in a least-squares sense) for linear Gaussian state-space models and is widely used in signal processing, control systems, robotics, and finance.

A standard discrete-time Kalman filter models a system with:

- State transition equation (describes how the system evolves):

$$x_k = Ax_{k-1} + w_k$$

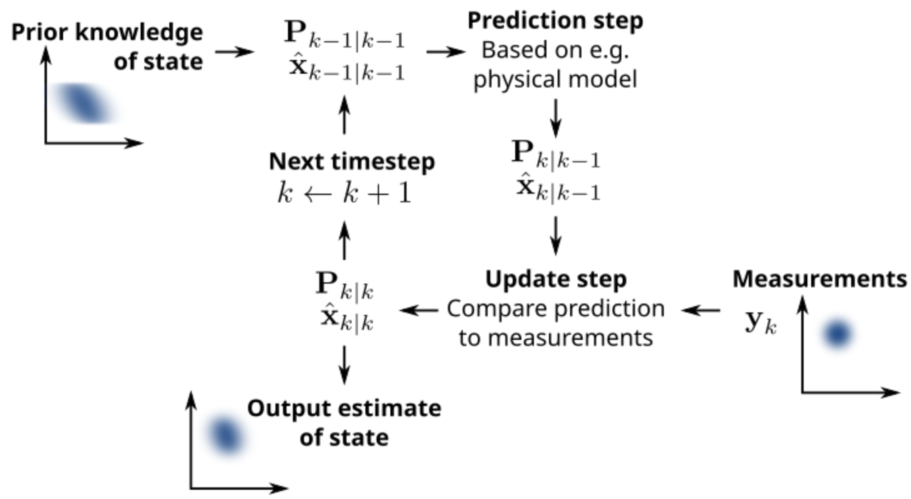

Figure 1: Caption

where x_k is the hidden state at time k, A is the transition matrix, and w_k is Gaussian process noise:

$$w_k \sim \mathcal{N}(0, Q)$$

- Observation equation (describes how we measure the state):

$$y_k = Hx_k + v_k$$

where y_k is the observation, H is the observation matrix, and v_k is Gaussian measurement noise:

$$v_k \sim \mathcal{N}(0, R)$$

Thus, estimates the posterior distribution of x_k given all previous observations by maintaining a mean estimate $\hat{x}_k$ and uncertainty covariance P_k.

2.1 Algorithm

At its core, the Kalman filter operates on a predict-update cycle. This process is Bayesian, meaning the filter maintains a probabilistic belief over the system's state, updating it as new information comes in. Moreover, the recursive structure makes Kalman filtering computationally efficient, as it only requires matrix multiplications and inversions per time step.

1. **Predict step**: Given the previous state estimate, we predict the next state using a known transition model.

$$\hat{x}_{k|k-1} = A\hat{x}_{k-1|k-1}$$
$$P_{k|k-1} = AP_{k-1|k-1}A^T + Q$$

2. **Update Step**: When a new observation arrives, we correct the predicted state using the measurement and its associated uncertainty.

   (a) Compute Kalman Gain:

$$K_k = P_{k|k-1}H^T \left(HP_{k|k-1}H^T + R\right)^{-1}$$

(b) Update state estimate:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \left(y_k - H \hat{x}_{k|k-1} \right)$$

(c) Update uncertainty:

$$P_{k|k} = (I - K_k H) \, P_{k|k-1}$$

3 Our Framework

We provide a clean, minimalistic, and extensible implementation of different Kalman filter variants:

- Standard Kalman Filter – The foundational model for linear state estimation.
- Extended Kalman Filter (EKF) – Handling nonlinear dynamics through local linearization.
- Unscented Kalman Filter (UKF) – A more accurate approach using sigma-point sampling.
- Variational Kalman Filters – Leveraging modern probabilistic techniques for scalable inference.