Signal Processing

Lecture 13: Optimal Filtering II

Konstantinos Chatzilygeroudis - costashatz@upatras.gr

Department of Electrical and Computer Engineering
University of Patras

Template made by Panagiotis Papagiannopoulos

Motivation: Why Kalman Filtering?

- Previously, we studied *Wiener filtering*: an optimal **batch** linear estimator for **stationary** signals, typically in the frequency domain.
- Many real systems are **dynamic** and **non-stationary**:
  - states evolve over time (e.g., position/velocity, temperature, channel gains),
  - measurements arrive **sequentially** and may be noisy or incomplete.
- We want an estimator that is:
  - **recursive** (online): update estimates as each new measurement arrives,
  - **model-based**: uses system dynamics,
  - **uncertainty-aware**: tracks confidence via covariance matrices.
- **Kalman filtering** provides the optimal linear MMSE estimator for **linear Gaussian state-space models**.

Key idea: predict using the model, then correct using measurements.

Learning Outcomes

By the end of this lecture, you should be able to:

1. Formulate a **linear Gaussian state-space model**:

$$\boldsymbol{x}_k = \boldsymbol{A}\boldsymbol{x}_{k-1} + \boldsymbol{B}\boldsymbol{u}_{k-1} + \boldsymbol{w}_{k-1}, \qquad \boldsymbol{y}_k = \boldsymbol{C}\boldsymbol{x}_k + \boldsymbol{v}_k.$$

2. Derive the Kalman filter recursion as the **optimal linear MMSE / MAP estimator** under Gaussian assumptions.
3. Apply the **predict–update** steps to propagate:
   - the state estimate mean $\hat{\boldsymbol{x}}_k$,
   - the estimation error covariance $\boldsymbol{P}_k$.
4. Interpret the roles of:
   - the innovation $\boldsymbol{r}_k$,
   - the Kalman gain $\boldsymbol{K}_k$,
   - the covariance matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$.
5. Recognize key assumptions and limitations of the Kalman filter (linearity, Gaussian noise, correct models).

State-Space Modeling (Generic Form)

Many dynamical systems can be described using a **hidden state** that evolves over time and generates noisy measurements.

State (process) model:

$$\boldsymbol{x}_k = \boldsymbol{f}(\boldsymbol{x}_{k-1},\, \boldsymbol{u}_{k-1},\, k-1) + \boldsymbol{w}_{k-1}$$

Measurement (observation) model:

$$\boldsymbol{y}_k = \boldsymbol{h}(\boldsymbol{x}_k,\, k) + \boldsymbol{v}_k$$

- $\boldsymbol{x}_k \in \mathbb{R}^n$: latent/hidden state (what we want to estimate)
- $\boldsymbol{y}_k \in \mathbb{R}^m$: observed measurement
- $\boldsymbol{u}_k$: known input/control (may be absent)
- $\boldsymbol{w}_k$: process noise (model uncertainty)
- $\boldsymbol{v}_k$: measurement noise (sensor uncertainty)

Goal: infer $\boldsymbol{x}_k$ from measurements $\boldsymbol{y}_{1:k}$.

Kalman Filter: State-Space Assumptions

The Kalman filter is optimal under the following **linear–Gaussian** state-space model:

State (process) model:

$$\boldsymbol{x}_k = \boldsymbol{A}\boldsymbol{x}_{k-1} + \boldsymbol{B}\boldsymbol{u}_{k-1} + \boldsymbol{w}_{k-1}$$

Measurement (observation) model:

$$\boldsymbol{y}_k = \boldsymbol{C}\boldsymbol{x}_k + \boldsymbol{v}_k$$

Noise assumptions:

$$\boldsymbol{w}_k \sim \mathcal{N}(\mathbf{0}, \boldsymbol{Q}), \qquad \boldsymbol{v}_k \sim \mathcal{N}(\mathbf{0}, \boldsymbol{R})$$

$$\mathbb{E}[\boldsymbol{w}_k \boldsymbol{v}_j^\top] = \mathbf{0} \quad \forall k, j, \qquad \text{(independent, white, zero-mean)}$$

Initial condition:

$$\boldsymbol{x}_0 \sim \mathcal{N}(\hat{\boldsymbol{x}}_0, \boldsymbol{P}_0)$$

Under these assumptions, $p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k})$ *remains Gaussian.*

We want the posterior over the hidden state at time k:

$$p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k})$$

Bayesian recursion:

1) Prediction (time update):

$$p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k-1}) = \int p(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1})\, p(\boldsymbol{x}_{k-1} \mid \boldsymbol{y}_{1:k-1})\, d\boldsymbol{x}_{k-1}$$

2) Update (measurement update):

$$p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k}) \propto p(\boldsymbol{y}_k \mid \boldsymbol{x}_k)\, p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k-1})$$

- **Prediction:** propagate the previous posterior through the dynamics.
- **Update:** correct the prediction using the new measurement.

Linear dynamics + Gaussian noise $\Rightarrow$ all distributions remain Gaussian $\Rightarrow$ track only mean and covariance.

Prediction Step (Time Update)

Assume at time $k-1$ we have the posterior:

$$p(\boldsymbol{x}_{k-1} \mid \boldsymbol{y}_{1:k-1}) = \mathcal{N}(\hat{\boldsymbol{x}}_{k-1}, \boldsymbol{\Sigma}_{k-1}) .$$

Using the state model $\boldsymbol{x}_k = \boldsymbol{A}\boldsymbol{x}_{k-1} + \boldsymbol{B}\boldsymbol{u}_{k-1} + \boldsymbol{w}_{k-1}$, we compute the **prior (predicted) distribution**:

$$p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k-1}) = \mathcal{N}\left(\hat{\boldsymbol{x}}_k^-, \boldsymbol{\Sigma}_k^-\right) .$$

Predicted mean:

$$\boxed{\hat{\boldsymbol{x}}_k^- = \mathbb{E}[\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k-1}] = \boldsymbol{A}\hat{\boldsymbol{x}}_{k-1} + \boldsymbol{B}\boldsymbol{u}_{k-1}}$$

Predicted covariance:

$$\boxed{\boldsymbol{\Sigma}_k^- = \mathrm{Cov}[\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k-1}] = \boldsymbol{A}\boldsymbol{\Sigma}_{k-1}\boldsymbol{A}^\top + \boldsymbol{Q}}$$

- The covariance **grows** due to model uncertainty $\boldsymbol{Q}$.
- $\hat{\boldsymbol{x}}_k^-, \boldsymbol{\Sigma}_k^-$ summarize **what we believe before seeing** $\boldsymbol{y}_k$.

Update Step (Measurement Update)

After prediction we have the prior:

$$p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k-1}) = \mathcal{N}\left(\hat{\boldsymbol{x}}_k^-, \boldsymbol{\Sigma}_k^-\right).$$

Using the measurement model $\boldsymbol{y}_k = \boldsymbol{C}\boldsymbol{x}_k + \boldsymbol{v}_k$, we incorporate the new observation $\boldsymbol{y}_k$ to obtain:

$$p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k}) = \mathcal{N}(\hat{\boldsymbol{x}}_k, \boldsymbol{\Sigma}_k).$$

Innovation (residual):

$$\boxed{\boldsymbol{r}_k = \boldsymbol{y}_k - \boldsymbol{C}\hat{\boldsymbol{x}}_k^-}$$

Innovation covariance:

$$\boxed{\boldsymbol{S}_k = \boldsymbol{C}\boldsymbol{\Sigma}_k^-\boldsymbol{C}^\top + \boldsymbol{R}}$$

Kalman gain:

$$\boxed{\boldsymbol{K}_k = \boldsymbol{\Sigma}_k^-\boldsymbol{C}^\top\boldsymbol{S}_k^{-1}}$$

Update Step (Measurement Update) (2)

Kalman gain:

$$\boxed{\boldsymbol{K}_k = \boldsymbol{\Sigma}_k^- \boldsymbol{C}^\top \boldsymbol{S}_k^{-1}}$$

Updated mean (posterior estimate):

$$\boxed{\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k \boldsymbol{r}_k}$$

Updated covariance:

$$\boxed{\boldsymbol{\Sigma}_k = (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{C})\, \boldsymbol{\Sigma}_k^-}$$

- $\boldsymbol{r}_k$ measures **new information** in $\boldsymbol{y}_k$.
- $\boldsymbol{K}_k$ balances trust between model ($\boldsymbol{\Sigma}_k^-$) and sensor ($\boldsymbol{R}$).

Worked Example: 1D Kalman Filter

Model (random walk):

$$x_k = x_{k-1} + w_{k-1}, \qquad y_k = x_k + v_k, \qquad \Rightarrow \boldsymbol{A} = 1, \boldsymbol{C} = 1.$$

Noise: $w_{k-1} \sim \mathcal{N}(0, Q)$ with $Q = 1$, $\quad v_k \sim \mathcal{N}(0, R)$ with $R = 4$.
Prior at $k = 0$: $\hat{x}_0 = 0, \qquad \Sigma_0 = 2.$
New measurement: $y_1 = 3.$
Prediction: $\hat{x}_1^- = A\hat{x}_0 = 1 \cdot 0 = 0, \qquad \Sigma_1^- = A\Sigma_0 A^\top + Q = 1 \cdot 2 \cdot 1 + 1 = 3.$
Update:

$$r_1 = y_1 - C\hat{x}_1^- = 3 - 1 \cdot 0 = 3$$

$$S_1 = C\Sigma_1^- C^\top + R = 1 \cdot 3 \cdot 1 + 4 = 7$$

$$K_1 = \Sigma_1^- C^\top S_1^{-1} = \frac{3}{7}$$

$$\hat{x}_1 = \hat{x}_1^- + K_1 r_1 = 0 + \frac{3}{7} \cdot 3 = \frac{9}{7} \approx 1.29$$

$$\Sigma_1 = (1 - K_1 C)\Sigma_1^- = \left(1 - \frac{3}{7}\right) 3 = \frac{12}{7} \approx 1.71$$

Interpretation: measurement pulls the estimate from 0 *toward* 3*, and uncertainty drops.*

Worked Example: 2D Kalman Filter

State: $\boldsymbol{x}_k = \begin{bmatrix} q_k \\ v_k \end{bmatrix}$ (position q, velocity v), and we only observe q.

Model (constant velocity):

$$\boldsymbol{x}_k = \boldsymbol{A}\boldsymbol{x}_{k-1} + \boldsymbol{w}_{k-1}, \qquad \boldsymbol{y}_k = \boldsymbol{C}\boldsymbol{x}_k + \boldsymbol{v}_k$$

$$\boldsymbol{A} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \qquad \boldsymbol{C} = \begin{bmatrix} 1 & 0 \end{bmatrix}.$$

Noise:

$$\boldsymbol{w}_{k-1} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{Q}), \quad \boldsymbol{Q} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad \boldsymbol{v}_k \sim \mathcal{N}(0, R), \quad R = 4.$$

Prior at $k = 0$:

$$\hat{\boldsymbol{x}}_0 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \qquad \boldsymbol{\Sigma}_0 = \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix}, \qquad y_1 = 2.$$

Prediction:

$$\hat{\boldsymbol{x}}_1^- = \boldsymbol{A}\hat{\boldsymbol{x}}_0 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\boldsymbol{\Sigma}_1^- = \boldsymbol{A}\boldsymbol{\Sigma}_0\boldsymbol{A}^\top + \boldsymbol{Q} = \begin{bmatrix} 6 & 1 \\ 1 & 2 \end{bmatrix}.$$

Worked Example: 2D Kalman Filter (2)

Prediction:

$$\hat{\boldsymbol{x}}_1^- = \boldsymbol{A}\hat{\boldsymbol{x}}_0 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\boldsymbol{\Sigma}_1^- = \boldsymbol{A}\boldsymbol{\Sigma}_0\boldsymbol{A}^\top + \boldsymbol{Q} = \begin{bmatrix} 6 & 1 \\ 1 & 2 \end{bmatrix}.$$

Update:

$$r_1 = y_1 - \boldsymbol{C}\hat{\boldsymbol{x}}_1^- = 2 - 1 = 1$$

$$S_1 = \boldsymbol{C}\boldsymbol{\Sigma}_1^-\boldsymbol{C}^\top + R = 6 + 4 = 10$$

$$\boldsymbol{K}_1 = \boldsymbol{\Sigma}_1^-\boldsymbol{C}^\top S_1^{-1} = \frac{1}{10}\begin{bmatrix} 6 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{3}{5} \\ \frac{1}{10} \end{bmatrix}$$

$$\hat{\boldsymbol{x}}_1 = \hat{\boldsymbol{x}}_1^- + \boldsymbol{K}_1 r_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} \frac{3}{5} \\ \frac{1}{10} \end{bmatrix} = \begin{bmatrix} \frac{8}{5} \\ \frac{11}{10} \end{bmatrix} \approx \begin{bmatrix} 1.60 \\ 1.10 \end{bmatrix}$$

$$\boldsymbol{\Sigma}_1 = (\boldsymbol{I} - \boldsymbol{K}_1\boldsymbol{C})\boldsymbol{\Sigma}_1^- = \begin{bmatrix} \frac{12}{5} & \frac{2}{5} \\ \frac{2}{5} & \frac{19}{10} \end{bmatrix}.$$

Even though we only measure q, the update also improves v via the dynamics/coupling.

Derivation of the Kalman Filter (Setup)

We assume the linear–Gaussian model:

$$\boldsymbol{x}_k = \boldsymbol{A}\boldsymbol{x}_{k-1} + \boldsymbol{B}\boldsymbol{u}_{k-1} + \boldsymbol{w}_{k-1}, \qquad \boldsymbol{y}_k = \boldsymbol{C}\boldsymbol{x}_k + \boldsymbol{v}_k,$$

$$\boldsymbol{w}_{k-1} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}), \qquad \boldsymbol{v}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R}).$$

Assume the prior (prediction) is Gaussian:

$$p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k-1}) = \mathcal{N}\left(\hat{\boldsymbol{x}}_k^-, \boldsymbol{\Sigma}_k^-\right).$$

We look for a **linear correction** using $\boldsymbol{y}_k$:

$$\boxed{\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k\left(\boldsymbol{y}_k - \boldsymbol{C}\hat{\boldsymbol{x}}_k^-\right)}$$

where $\boldsymbol{K}_k$ is chosen to minimize the MMSE.

Derivation of the Kalman Gain

Define the estimation error:

$$\boldsymbol{e}_k = \boldsymbol{x}_k - \hat{\boldsymbol{x}}_k, \qquad \boldsymbol{e}_k^- = \boldsymbol{x}_k - \hat{\boldsymbol{x}}_k^-.$$

Using $\boldsymbol{y}_k = \boldsymbol{C}\boldsymbol{x}_k + \boldsymbol{v}_k$,

$$\boldsymbol{e}_k = \boldsymbol{e}_k^- - \boldsymbol{K}_k\left(\boldsymbol{C}\boldsymbol{e}_k^- + \boldsymbol{v}_k\right) = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{C})\boldsymbol{e}_k^- - \boldsymbol{K}_k\boldsymbol{v}_k.$$

Hence the posterior covariance is

$$\boldsymbol{\Sigma}_k = \mathbb{E}[\boldsymbol{e}_k\boldsymbol{e}_k^\top] = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{C})\boldsymbol{\Sigma}_k^-(\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{C})^\top + \boldsymbol{K}_k\boldsymbol{R}\boldsymbol{K}_k^\top.$$

Choose $\boldsymbol{K}_k$ to minimize

$$J(\boldsymbol{K}_k) = \mathrm{tr}(\boldsymbol{\Sigma}_k).$$

Derivation of the Kalman Gain (Minimization)

Expand $J(\boldsymbol{K}_k)$ (trace is linear and invariant under cyclic permutations):

$$J = \mathrm{tr}(\boldsymbol{\Sigma}_k^-) - 2\,\mathrm{tr}(\boldsymbol{K}_k \boldsymbol{C} \boldsymbol{\Sigma}_k^-) + \mathrm{tr}\Big(\boldsymbol{K}_k (\boldsymbol{C}\boldsymbol{\Sigma}_k^- \boldsymbol{C}^\top + \boldsymbol{R}) \boldsymbol{K}_k^\top\Big).$$

Differentiate w.r.t. $\boldsymbol{K}_k$ and set to zero:

$$\frac{\partial J}{\partial \boldsymbol{K}_k} = -2\boldsymbol{\Sigma}_k^- \boldsymbol{C}^\top + 2\boldsymbol{K}_k(\boldsymbol{C}\boldsymbol{\Sigma}_k^- \boldsymbol{C}^\top + \boldsymbol{R}) = \boldsymbol{0}.$$

Solve for $\boldsymbol{K}_k$:

$$\boxed{\boldsymbol{K}_k = \boldsymbol{\Sigma}_k^- \boldsymbol{C}^\top \big(\boldsymbol{C}\boldsymbol{\Sigma}_k^- \boldsymbol{C}^\top + \boldsymbol{R}\big)^{-1}}$$

Define the innovation covariance $\boldsymbol{S}_k = \boldsymbol{C}\boldsymbol{\Sigma}_k^- \boldsymbol{C}^\top + \boldsymbol{R}$, so $\boldsymbol{K}_k = \boldsymbol{\Sigma}_k^- \boldsymbol{C}^\top \boldsymbol{S}_k^{-1}$.

Resulting Kalman Update Equations

Innovation:

$$\boldsymbol{r}_k = \boldsymbol{y}_k - \boldsymbol{C}\hat{\boldsymbol{x}}_k^-, \qquad \boldsymbol{S}_k = \boldsymbol{C}\boldsymbol{\Sigma}_k^-\boldsymbol{C}^\top + \boldsymbol{R}.$$

Kalman gain:

$$\boldsymbol{K}_k = \boldsymbol{\Sigma}_k^-\boldsymbol{C}^\top\boldsymbol{S}_k^{-1}.$$

Posterior mean:

$$\boxed{\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k\boldsymbol{r}_k}$$

Posterior covariance:

$$\boxed{\boldsymbol{\Sigma}_k = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{C})\boldsymbol{\Sigma}_k^-}$$

The optimal linear MMSE estimator is obtained by minimizing the posterior error covariance.

Intuition I: The Role of the Kalman Gain

$$\boldsymbol{K}_k = \boldsymbol{\Sigma}_k^- \boldsymbol{C}^\top \left(\boldsymbol{C} \boldsymbol{\Sigma}_k^- \boldsymbol{C}^\top + \boldsymbol{R} \right)^{-1} = \boldsymbol{\Sigma}_k^- \boldsymbol{C}^\top \boldsymbol{S}_k^{-1}.$$

- **Large measurement noise $\boldsymbol{R}$**

  $\Rightarrow \boldsymbol{S}_k$ large $\Rightarrow \boldsymbol{K}_k$ small $\Rightarrow$ trust the model (prediction) more.

- **Large prior uncertainty $\boldsymbol{\Sigma}_k^-$**

  $\Rightarrow \boldsymbol{K}_k$ large $\Rightarrow$ trust the measurement more.

$\boldsymbol{K}_k$ automatically balances model vs. sensor reliability.

Intuition II: Innovation = New Information

The innovation (measurement residual) is

$$\boxed{\boldsymbol{r}_k = \boldsymbol{y}_k - \boldsymbol{C}\hat{\boldsymbol{x}}_k^-}$$

- $\boldsymbol{C}\hat{\boldsymbol{x}}_k^-$ is what the model **expected** to measure.
- $\boldsymbol{r}_k$ is what is **unexpected** (new information).

Interpretation

- If $\boldsymbol{r}_k \approx \boldsymbol{0}$: measurement matches prediction $\Rightarrow$ small correction.
- If $\|\boldsymbol{r}_k\|$ is large: measurement disagrees with prediction $\Rightarrow$ strong correction through $\boldsymbol{K}_k\boldsymbol{r}_k$.

Intuition III: Why Uncertainty Shrinks After Update

Posterior covariance update:

$$\boxed{\boldsymbol{\Sigma}_k = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{C})\boldsymbol{\Sigma}_k^-}$$

- The measurement provides information only in the **observed subspace** (defined by $\boldsymbol{C}$).
- Multiplying by $(\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{C})$ removes uncertainty along those directions.
- Unobserved directions keep their uncertainty (or may even grow in prediction).

Prediction spreads uncertainty; update "pulls it back" using data.

Kalman Filter Algorithm (Predict–Update)

Input: $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{Q}, \boldsymbol{R}$; initial $\hat{\boldsymbol{x}}_0, \boldsymbol{\Sigma}_0$; measurements $\{\boldsymbol{y}_k\}$; inputs $\{\boldsymbol{u}_k\}$
Output: Filtered estimates $\{\hat{\boldsymbol{x}}_k, \boldsymbol{\Sigma}_k\}$

for $k = 1, 2, \ldots$ **do**

- `// Prediction (Time Update)`
- $\hat{\boldsymbol{x}}_k^- \leftarrow \boldsymbol{A}\hat{\boldsymbol{x}}_{k-1} + \boldsymbol{B}\boldsymbol{u}_{k-1}$
- $\boldsymbol{\Sigma}_k^- \leftarrow \boldsymbol{A}\boldsymbol{\Sigma}_{k-1}\boldsymbol{A}^\top + \boldsymbol{Q}$
- `// Update (Measurement Update)`
- $\boldsymbol{r}_k \leftarrow \boldsymbol{y}_k - \boldsymbol{C}\hat{\boldsymbol{x}}_k^-$ `// innovation`
- $\boldsymbol{S}_k \leftarrow \boldsymbol{C}\boldsymbol{\Sigma}_k^-\boldsymbol{C}^\top + \boldsymbol{R}$ `// innovation cov.`
- $\boldsymbol{K}_k \leftarrow \boldsymbol{\Sigma}_k^-\boldsymbol{C}^\top\boldsymbol{S}_k^{-1}$ `// Kalman gain`
- $\hat{\boldsymbol{x}}_k \leftarrow \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k\boldsymbol{r}_k$
- $\boldsymbol{\Sigma}_k \leftarrow (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{C})\boldsymbol{\Sigma}_k^-$

Assumptions for Optimality

The Kalman filter is the **optimal linear MMSE estimator** only if:

1. **Linearity:**

$$\boldsymbol{x}_k = \boldsymbol{A}\boldsymbol{x}_{k-1} + \boldsymbol{B}\boldsymbol{u}_{k-1} + \boldsymbol{w}_{k-1}, \qquad \boldsymbol{y}_k = \boldsymbol{C}\boldsymbol{x}_k + \boldsymbol{v}_k$$

2. **Gaussian, zero-mean noises:**

$$\boldsymbol{w}_k \sim \mathcal{N}(\mathbf{0}, \boldsymbol{Q}), \qquad \boldsymbol{v}_k \sim \mathcal{N}(\mathbf{0}, \boldsymbol{R})$$

3. **Correct noise covariances $\boldsymbol{Q}, \boldsymbol{R}$.**
4. **Correct model matrices $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$.**

Under these conditions, the posterior remains Gaussian and is fully described by $(\hat{\boldsymbol{x}}_k, \boldsymbol{\Sigma}_k)$.

When Assumptions Are Violated

- **Nonlinear dynamics / observations:**

$$\boldsymbol{x}_k = \boldsymbol{f}(\boldsymbol{x}_{k-1}) + \boldsymbol{w}_{k-1}, \quad \boldsymbol{y}_k = \boldsymbol{h}(\boldsymbol{x}_k) + \boldsymbol{v}_k$$

$\Rightarrow$ Extended KF (EKF), Unscented KF (UKF), Particle Filters

- **Heavy-tailed noise / outliers (non-Gaussian):**

$\Rightarrow$ Robust Kalman filtering variants

- **Incorrect tuning of $\boldsymbol{Q}, \boldsymbol{R}$:**

$\Rightarrow$ sluggish response, noisy estimates, or divergence

The filter is only as good as the model and its uncertainty description.

Effect of Tuning Q and R

Recall the Kalman gain:

$$\boldsymbol{K}_k = \boldsymbol{\Sigma}_k^- \boldsymbol{C}^\top (\boldsymbol{C}\boldsymbol{\Sigma}_k^- \boldsymbol{C}^\top + \boldsymbol{R})^{-1}.$$

Increase $\boldsymbol{Q}$ (more process noise)

- $\boldsymbol{\Sigma}_k^-$ becomes larger.
- $\Rightarrow \boldsymbol{K}_k$ increases.
- **Behavior:** filter trusts measurements more $\Rightarrow$ more responsive but noisier.

Increase $\boldsymbol{R}$ (more measurement noise)

- Denominator increases.
- $\Rightarrow \boldsymbol{K}_k$ decreases.
- **Behavior:** filter trusts model more $\Rightarrow$ smoother but slower to react.

Filtering vs. Smoothing

Given measurements $\boldsymbol{y}_{1:T}$ from a state-space model:

- **Kalman filter (online / causal):**

$$p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k}) \quad \Rightarrow \quad (\hat{\boldsymbol{x}}_k, \boldsymbol{\Sigma}_k)$$

  uses only measurements up to time k.

- **Kalman smoother (offline / non-causal):**

$$p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:T}) \quad \Rightarrow \quad (\hat{\boldsymbol{x}}_k^s, \boldsymbol{\Sigma}_k^s)$$

  uses *all* measurements in the interval $1{:}T$.

- **Key fact:** smoothing always improves (or keeps) accuracy:

$$\boldsymbol{\Sigma}_k^s \preceq \boldsymbol{\Sigma}_k \qquad \forall k.$$

Future measurements help correct past state estimates.

Kalman Smoothing (RTS) in Two Passes

Step 1: Forward pass (Kalman filter)
Compute and store $\hat{\boldsymbol{x}}_k, \hat{\boldsymbol{x}}_{k+1}^-, \boldsymbol{\Sigma}_k, \boldsymbol{\Sigma}_{k+1}^-$ for $k = 0{:}T$.

Step 2: Backward pass (Rauch–Tung–Striebel smoother)
For $k = T-1, \dots, 0$:

Smoother gain:

$$\boxed{\boldsymbol{G}_k = \boldsymbol{\Sigma}_k \boldsymbol{A}^\top (\boldsymbol{\Sigma}_{k+1}^-)^{-1}}$$

Smoothed mean:

$$\boxed{\hat{\boldsymbol{x}}_k^s = \hat{\boldsymbol{x}}_k + \boldsymbol{G}_k \left(\hat{\boldsymbol{x}}_{k+1}^s - \hat{\boldsymbol{x}}_{k+1}^- \right)}$$

Smoothed covariance:

$$\boxed{\boldsymbol{\Sigma}_k^s = \boldsymbol{\Sigma}_k + \boldsymbol{G}_k \left(\boldsymbol{\Sigma}_{k+1}^s - \boldsymbol{\Sigma}_{k+1}^- \right) \boldsymbol{G}_k^\top}$$

Batch View

Under the linear–Gaussian model, estimating the whole trajectory $\boldsymbol{x}_{0:T}$ is equivalent to a **batch MAP** problem:

$$\boxed{\boldsymbol{x}^{\star}_{0:T} = \arg\min_{\boldsymbol{x}_{0:T}} \left[\sum_{k=1}^{T} \|\boldsymbol{y}_k - \boldsymbol{C}\boldsymbol{x}_k\|^2_{\boldsymbol{R}^{-1}} + \sum_{k=1}^{T} \|\boldsymbol{x}_k - \boldsymbol{A}\boldsymbol{x}_{k-1} - \boldsymbol{B}\boldsymbol{u}_{k-1}\|^2_{\boldsymbol{Q}^{-1}} + \|\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0\|^2_{\boldsymbol{\Sigma}_0^{-1}} \right]}$$

- This is a **quadratic** optimization with block-tridiagonal structure (**Least Squares!**).
- Kalman filtering/smoothing are **efficient recursive solvers** of this problem.

Filtering: online solution vs. Smoothing/Batch MAP: offline solution.

Thank you

- **Any Questions?**
- **Office Hours**:
  - **Tue & Thu** *(09:00-11:00)*
  - 24/7 by email (`costashatz@upatras.gr`, subject: *ECE_SP_AM*)
- **Material and Announcements**

Laboratory of Automation & Robotics