The Kalman filter

Sensor fusion & nonlinear filtering

Lars Hammarstrand

ANALYTICAL SOLUTION TO THE FILTERING PROBLEM

- The filtering equations

$$p(\mathbf{x}_k|\mathbf{y}_{1:k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})\, d\mathbf{x}_{k-1} \leftarrow \text{prediction}$$

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) \propto p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_{1:k-1}) \leftarrow \text{update}$$

  are applicable to all filtering problems.

- Unfortunately, there are very few examples where the posterior distribution has an analytical expression.

LINEAR AND GAUSSIAN STATE SPACE MODELS

Definition (Linear and Gaussian models)

- For state vector $\mathbf{x}_k$ and observation $\mathbf{y}_k$,

$$
\begin{aligned}
\mathbf{x}_k &= \underbrace{\mathbf{A}_{k-1}}_{\text{Transition matrix}}\mathbf{x}_{k-1} + \underbrace{\mathbf{q}_{k-1}}_{\text{process noise}}, \quad & \mathbf{q}_{k-1} &\sim \mathcal{N}(\overset{0}{\cancel{\bar{\mathbf{q}}_{k-1}}}, \mathbf{Q}_{k-1}) \\
\mathbf{y}_k &= \underbrace{\mathbf{H}_k}_{\text{Measurement model matrix}}\mathbf{x}_k + \underbrace{\mathbf{r}_k}_{\text{measurement noise}}, \quad & \mathbf{r}_k &\sim \mathcal{N}(\underset{0}{\cancel{\bar{\mathbf{r}}_k}}, \mathbf{R}_k)
\end{aligned}
$$

and $\mathbf{x}_0 \sim \mathcal{N}(\bar{\mathbf{x}}_0, \mathbf{P}_{0|0})$.

Note:

- $p(\mathbf{x}_m \mid \mathbf{y}_{1:n})$ is Gaussian for all m and n, i.e., for all filtering, smoothing and prediction problems.

[$m>n$]

$p(x_{1:m}, y_{1:n})$

$\Rightarrow$ marginal Gaussian

$\Rightarrow$ conditional Gaussian

KALMAN FILTER

Kalman filter

- Analytical solution to the filtering equations for linear and Gaussian models.
- The Kalman filter recursively computes

$$p(\mathbf{x}_k|\mathbf{Y}_{1:k-1}) = \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1}) \leftarrow \text{Prediction step}$$

$$p(\mathbf{x}_k|\mathbf{Y}_{1:k}) = \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k}) \leftarrow \text{Update step}$$

for $k = 1, 2, \ldots$.

Note:

- Only need to compute the moments $\hat{\mathbf{x}}_{k|k-1}$, $\mathbf{P}_{k|k-1}$, $\hat{\mathbf{x}}_{k|k}$ and $\mathbf{P}_{k|k}$ in each recursion.

KALMAN FILTER: PREDICTION

Prediction step

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

$$\begin{cases} x_k = A_{k-1}x_{k-1} + q_{k-1} \\ p(x_{k-1} \mid y_{1:k-1}) = N(x_{k-1}; \hat{x}_{k-1|k-1}, P_{k-1|k-1}) \end{cases} \Longrightarrow E\{x_k \mid y_{1:k-1}\} = E\{\underbrace{A_{k-1}x_{k-1}}_{A_{k-1}\cdot\hat{x}_{k-1|k-1}} + \underbrace{q_{k-1}}_{=0} \mid y_{1:k-1}\} = A_{k-1}\hat{x}_{k-1|k-1}$$

Note:

- We assume that process noise $\mathbf{q}_{k-1}$ $\sim N(0, Q_{k-1})$ is zero mean.
- On the right hand side, only $\hat{\mathbf{x}}_{k-1|k-1}$ depend on $\mathbf{y}_{1:k-1}$.

KALMAN FILTER: UPDATE

Update step

- The posterior mean $\hat{\mathbf{x}}_{k|k}$ and covariance $\mathbf{P}_{k|k}$ is computed as

Prior — Correction term

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \mathbf{v}_k$$
$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{S}_k \mathbf{K}_k^T$$

where

Kalman Gain $$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T \mathbf{S}_k^{-1}$$

Innovation $$\mathbf{v}_k = \mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}$$

Innovation Covariance $$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k$$

Note:

- The posterior mean $\hat{\mathbf{x}}_{k|k} = \mathbb{E}\{\mathbf{x}_k | \mathbf{y}_{1:k}\}$ is both the MMSE and MAP estimator.

COMPONENTS IN THE KALMAN FILTER

$y_k = H_k x_k + r_k \Rightarrow E\{y_k | y_{1:k-1}\} = E\{H_k x_k + r_k | y_{1:k-1}\} = H_k \hat{x}_{k|k-1}$

A few remarks:

- It holds that $p(\mathbf{y}_k | \mathbf{y}_{1:k-1}) = \mathcal{N}(\mathbf{y}_k; \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}, \mathbf{S}_k)$, which means that $\mathbf{S}_k$ is the predicted covariance of $\mathbf{y}_k$.
- The innovation $\mathbf{v}_k = \mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}$ captures the new information in $\mathbf{y}_k$.
- In $\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \mathbf{v}_k$, the Kalman gain $\mathbf{K}_k$ determines how much we should trust the new information.

SELF ASSESSMENT

Recall that $\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \mathbf{v}_k$ and $\mathbf{v}_k = \mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}$.

Suppose $y_k = x_k + r_k$, such that $H_k = 1$, and $r_k \sim \mathcal{N}(0, R)$ (they are all scalar). Check all statements that apply:

- If $R = \infty$ then $K_k \approx 0$.
- If $R = 0$ then $K_k = \infty$.
- If $R = 1$ then $K_k = 0$.
- If $R = 0$ then $K_k = 1$.

A Bayesian derivation of the Kalman filter

Sensor fusion & nonlinear filtering

Lars Hammarstrand

LINEAR AND GAUSSIAN STATE SPACE MODELS

- Consider a linear and Gaussian model:

$$\begin{aligned} \mathbf{x}_k &= \mathbf{A}_{k-1}\mathbf{x}_{k-1} + \mathbf{q}_{k-1}, & \mathbf{q}_{k-1} &\sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{k-1}) \\ \mathbf{y}_k &= \mathbf{H}_k\mathbf{x}_k + \mathbf{r}_k, & \mathbf{r}_k &\sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k) \end{aligned}$$

  where $\mathbf{x}_0$ is Gaussian with mean $\hat{\mathbf{x}}_{0|0}$ and covariance matrices $\mathbf{P}_{0|0}$.

- We can also express this model as

$$\begin{aligned} p(\mathbf{x}_k | \mathbf{x}_{k-1}) &= \mathcal{N}(\mathbf{x}_k; \mathbf{A}_{k-1}\mathbf{x}_{k-1}, \mathbf{Q}_{k-1}) \\ p(\mathbf{y}_k | \mathbf{x}_k) &= \mathcal{N}(\mathbf{y}_k; \mathbf{H}_k\mathbf{x}_k, \mathbf{R}_k). \end{aligned}$$

Objective (in this video)

- Derive analytical expressions for $p(\mathbf{x}_k | \mathbf{y}_{1:k-1})$ and $p(\mathbf{x}_k | \mathbf{y}_{1:k})$.

A BRUTE FORCE DERIVATION

- It is possible to derive the Kalman filter equations from the filtering equations

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

$$p(\mathbf{x}_k | \mathbf{y}_{1:k}) \propto p(\mathbf{y}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{y}_{1:k-1})$$

- Unfortunately, the derivation involves various matrix manipulations and is rather tedious.

- **Standard derivations** instead make use of “well known” results regarding Gaussian distributions. We use this approach below.

PREDICTION STEP

Prediction step

- Objective is to compute $p(\mathbf{x}_k | \mathbf{y}_{1:k-1})$ using

$$\begin{cases} p(\mathbf{x}_{k-1} | \mathbf{y}_{1:k-1}) = \mathcal{N}(\mathbf{x}_{k-1}; \hat{\mathbf{x}}_{k-1|k-1}, \mathbf{P}_{k-1|k-1}) \\ \mathbf{x}_k = \mathbf{A}_{k-1}\mathbf{x}_{k-1} + \mathbf{q}_{k-1} \end{cases}$$

Background theory (well know results)

- if $\mathbf{z}_1 \sim \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Lambda}_1)$ and $\mathbf{z}_2 \sim \mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Lambda}_2)$ are independent

$$\Rightarrow \mathbf{z} = \mathbf{B}_1\mathbf{z}_1 + \mathbf{B}_2\mathbf{z}_2$$
$$\sim \mathcal{N}(\mathbf{B}_1\boldsymbol{\mu}_1 + \mathbf{B}_2\boldsymbol{\mu}_2, \mathbf{B}_1\boldsymbol{\Lambda}_1\mathbf{B}_1^T + \mathbf{B}_2\boldsymbol{\Lambda}_2\mathbf{B}_2^T).$$

$p(x_k | y_{1:k-1}) =$

$$\mathcal{N}(x_k; \underbrace{A_{k-1}\hat{x}_{k-1|k-1}}_{\hat{x}_{k|k-1}}, \underbrace{A_{k-1} P_{k-1|k-1} A_{k-1}^T + Q_{k-1}}_{P_{k|k-1}})$$

A LEMMA FOR THE UPDATE STEP

Conditional distribution of Gaussian variables

- If $\mathbf{x}$ and $\mathbf{y}$ are two Gaussian random variables with the joint probability density function

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \overset{E\{x\}}{\boldsymbol{\mu}_x} \\ \underset{E\{y\}}{\boldsymbol{\mu}_y} \end{bmatrix}, \begin{bmatrix} \overset{Cov\{x\}}{\mathbf{P}_{xx}} & \overset{Cov\{x,y\}}{\mathbf{P}_{xy}} \\ \underset{P_{xy}^T}{\mathbf{P}_{yx}} & \underset{Cov\{y\}}{\mathbf{P}_{yy}} \end{bmatrix} \right)$$

then the conditional density of $\mathbf{x}$ given $\mathbf{y}$ is

$$p(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_x + \mathbf{P}_{xy}\mathbf{P}_{yy}^{-1}(\mathbf{y} - \boldsymbol{\mu}_y), \mathbf{P}_{xx} - \mathbf{P}_{xy}\mathbf{P}_{yy}^{-1}\mathbf{P}_{yx})$$

Note:

- $\mathbf{P}_{xy} = \mathbf{0} \Rightarrow p(\mathbf{x}|\mathbf{y}) = p(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_x, \mathbf{P}_{xx})$.
- $\mathrm{Cov}\{\mathbf{x}|\mathbf{y}\} \le \mathbf{P}_{xx}$.

THE UPDATE STEP

$$\begin{bmatrix} x_k \\ y_k \end{bmatrix} = \begin{bmatrix} I \\ H_k \end{bmatrix} x_k + \begin{bmatrix} 0 \\ I \end{bmatrix} r_k$$

- We have a predicted density $\mathbf{x}_k \big| \mathbf{y}_{1:k-1} \sim \mathcal{N}(\hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1})$ and observe a measurement $\mathbf{y}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{r}_k$

$$\Rightarrow \begin{bmatrix} \mathbf{x}_k \\ \mathbf{y}_k \end{bmatrix} \Bigg| \, \mathbf{y}_{1:k-1} \sim \mathcal{N}\left(\begin{bmatrix} \overbrace{\hat{\mathbf{x}}_{k|k-1}}^{\mu_x} \\ \underbrace{\mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}}_{\mu_y} \end{bmatrix}, \begin{bmatrix} \overbrace{\mathbf{P}_{k|k-1}}^{P_{xx}} & \overbrace{\mathbf{P}_{k|k-1} \mathbf{H}_k^T}^{P_{xy}} \\ \underbrace{\mathbf{H}_k \mathbf{P}_{k|k-1}}_{P_{yx} = P_{xy}^T} & \underbrace{\mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k}_{P_{yy} = S_k} \end{bmatrix} \right)$$

$$\boxed{S_k \cdot S_k^{-1} = I}$$

- We use the notation $p(\mathbf{x}_k \big| \mathbf{y}_{1:k}) = \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k})$ where

$$\hat{x}_{k|k} = \mu_x + P_{xx} \cdot P_{yy}^{-1} (y - \mu_y)$$

$$= \hat{x}_{k|k-1} + \underbrace{P_{k|k-1} \cdot H_k^T S_k^{-1}}_{K_k} \underbrace{(y_k - H_k \hat{x}_{k|k-1})}_{v_k}$$

$$= \hat{x}_{k|k-1} + K_k \cdot v_k$$

$$P_{k|k} = P_{xx} - P_{xy} \cdot P_{yy}^{-1} \cdot P_{xy}^T = P_{k|k-1} - P_{k|k-1} \cdot H_k^T S_k^{-1} \left(P_{k|k-1} H_k^T \right)^T$$

$$= P_{k|k-1} - \underbrace{P_{k|k-1} \cdot H^T S_k^{-1}}_{K_k} S_k \underbrace{S_k^{-T} \cdot (P_{k|k-1} H_k^T)^T}_{K_k^T} = P_{k|k-1} - K_k \cdot S_k \cdot K_k^T$$

SELF ASSESSMENT

With an ideal sensor we would have $y_k = x_k$. (We consider a scalar case here for simplicity.) Under that assumption, which of the following apply?

- $p(y_k | y_{1:k-1}) = \mathcal{N}(y_k; \hat{x}_{k|k-1}, \mathbf{P}_{k|k-1})$
- $p(y_k | y_{1:k-1}) = \delta(y_k - x_k)$
- $p(x_k, y_k | y_{1:k-1}) = \mathcal{N}\left(\begin{bmatrix} x_k \\ y_k \end{bmatrix} \middle| \begin{bmatrix} \hat{x}_{k|k-1} \\ \hat{\mathbf{x}}_{k|k-1} \end{bmatrix}, \begin{bmatrix} P_{k|k-1} & P_{k|k-1} \\ P_{k|k-1} & P_{k|k-1} \end{bmatrix} \right)$
- $p(x_k, y_k | y_{1:k-1}) = \mathcal{N}\left(\begin{bmatrix} x_k \\ y_k \end{bmatrix} \middle| \begin{bmatrix} \hat{x}_{k|k-1} \\ x_k \end{bmatrix}, \begin{bmatrix} P_{k|k-1} & 0 \\ 0 & P_{k|k-1} \end{bmatrix} \right)$

Kalman filter tuning and consistency

Sensor fusion & nonlinear filtering

Lars Hammarstrand

A MATHEMATICAL RESULT BEFORE WE START

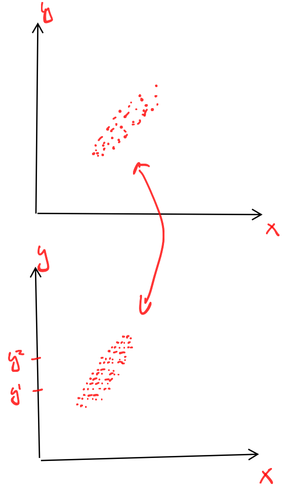
Decomposing joint expectations (Product rule)

For any two random variables x and y, it holds that

$$\mathbb{E}\{g(x,y)\} = \mathbb{E}\{\underbrace{\mathbb{E}\{g(x,y)|y\}}_{h(y)}\} = \mathbb{E}\{h(y)\}$$

$$\iint g(x,y)\underbrace{p(x,y)}_{p(x|y)p(y)}\,dx\,dy = \iint \underbrace{g(x,y)\,p(x|y)\,dx}_{h(y)}\; p(y)\,dy$$

$$= \int h(y)\,p(y)\,dy = E\{h(y)\}$$



THE KALMAN FILTER

$$x_k = \begin{bmatrix} x_1 \\ x_2 \\ \dot{x}_1 \\ \dot{x}_2 \end{bmatrix}$$

Prediction

$$\begin{cases} \hat{\mathbf{x}}_{k|k-1} &= \mathbf{A}_{k-1}\hat{\mathbf{x}}_{k-1|k-1} \\ \mathbf{P}_{k|k-1} &= \mathbf{A}_{k-1}\mathbf{P}_{k-1|k-1}\mathbf{A}_{k-1}^T + \mathbf{Q}_{k-1} \end{cases}$$

Update

$$\begin{cases} \hat{\mathbf{x}}_{k|k} &= \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\mathbf{v}_k \\ \mathbf{P}_{k|k} &= \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^T \end{cases}$$

$$\begin{cases} \mathbf{K}_k &= \mathbf{P}_{k|k-1}\mathbf{H}_k^T\mathbf{S}_k^{-1} \\ \mathbf{v}_k &= \mathbf{y}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k|k-1} \\ \mathbf{S}_k &= \mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^T + \mathbf{R}_k \end{cases}$$

Does the filter perform well?

- Have we implemented the filter correctly?
- Have we selected good model types?
- Are the covariance matrices properly tuned?

IDEAL PROPERTIES OF FILTER OUTPUTS

- The filter output is the posterior mean and covariance:

$$p(\mathbf{x}_k | \mathbf{y}_{1:k}) = \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k})$$

Both over x_k and $y_{1:n}$

Deterministic function of $y_{1:n}$

A well performing filter should satisfy

$$E\{E\{x_k - \hat{x}_{k|k} | y_{1:k}\}\} = \mathbb{E}\{\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}\} = 0$$

$$\underbrace{\mathbb{E}\left\{ (\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}) (\mathbf{x}_k - \hat{\mathbf{x}}_{k|k})^T \Big| \mathbf{y}_{1:k} \right\}}_{\mathbf{P}_{k|k}} = \underbrace{\mathbb{E}\left\{ (\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}) (\mathbf{x}_k - \hat{\mathbf{x}}_{k|k})^T \right\}}_{\text{MSE}}$$

Ind. of $y_{1:k}$

- Weakness: need to know $\mathbf{x}_k$ to check these conditions!
  - ⇝ simulations?
  - ⇝ reference sensors in test environment?

SELF-ASSESSMENT

Why is it often difficult to check if $\mathbb{E}\{\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}\}$ using real data (measurements that we have not simulated in a computer):

- It is not a good idea to approximate expected values using ensemble averaging.
- It is difficult to compute $\hat{\mathbf{x}}_{k|k}$.
- We do not know the values of $\mathbf{x}_k$.

Check all that apply.

Kalman filter tuning and consistency – Innovation

Sensor fusion & nonlinear filtering

Lars Hammarstrand

INNOVATION CONSISTENCY

Innovation consistency

The innovation $\mathbf{v}_k = \mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}$ should satisfy

$$p(\mathbf{v}_k | \mathbf{y}_{1:k-1}) = \mathcal{N}(\mathbf{v}_k; \mathbf{0}, \mathbf{S}_k)$$

$$\mathrm{Cov}(\mathbf{v}_k, \mathbf{v}_{k-l}) = \begin{cases} \mathrm{Cov}\{\mathbf{v}_k\} & \text{if } l = 0 \\ \mathbf{0} & \text{otherwise.} \end{cases}$$

- Note: it is enough to have a filter and a measurement sequence to compute $\mathbf{v}_1, \mathbf{v}_2, \ldots$

Proof:

$$E\{\underbrace{v_k}_{y_k - H_k \hat{x}_{k|k-1}} | y_{1:k-1}\} = \underbrace{E\{y_k | y_{1:k-1}\}}_{H_k \hat{x}_{k|k-1}} - H_k \hat{x}_{k|k-1}$$

$$= 0$$

$$\Rightarrow E\{v_k\} = 0$$

$$E\{v_k \cdot v_{k-l}^T | y_{1:k-1}\} = 0$$

$$\hookrightarrow y_{k-l} + H_{k-l} \cdot \hat{x}_{k-l|k-l-1}$$

$$\Rightarrow E\{v_k v_{k-l}^T\} = 0 \quad l > 0$$

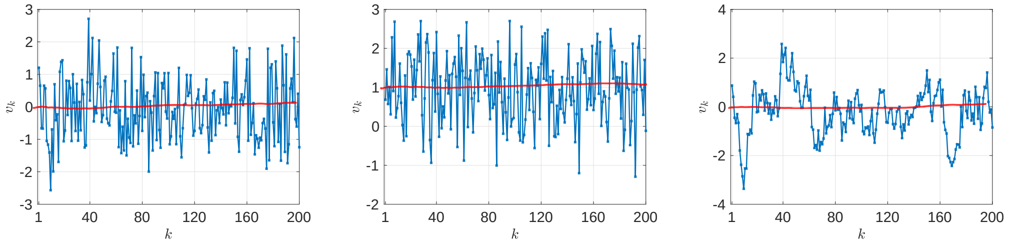
$$l = 0 \Rightarrow E\{v_k \cdot v_k^T\} = \mathrm{Cov}\{v_k\}$$

TEST OF INNOVATION PROPERTIES – VISUAL INSPECTION

- There are ways to test the properties of the innovation.

  ⇝ we will look at three methods.

- Visual inspection:
  - Zero mean?
  - uncorrelated?



TEST OF INNOVATION PROPERTIES – CONSISTENCY

Consistency

- Ideally $\mathbf{v}_k | \mathbf{y}_{1:k-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{S}_k)$ and then

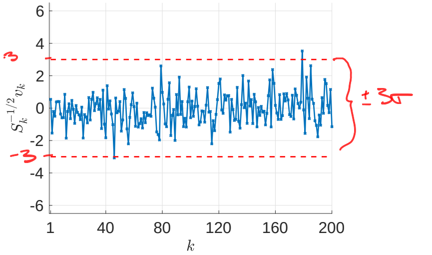
$$\mathbf{S}_k^{-1/2}\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \Rightarrow \mathbf{v}_k^T \mathbf{S}_k^{-1} \mathbf{v}_k \sim \chi^2_{n_y}$$

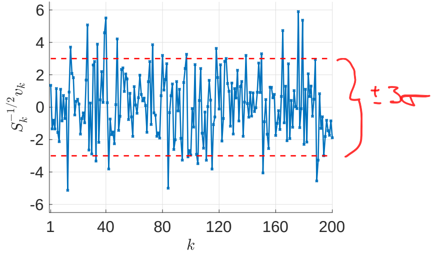
dim. of y_k

- Given a sequence $v_1, v_2, \ldots, v_K$ we can compute

$$\xi_K = \sum_{k=1}^{K} \mathbf{v}_k^T \mathbf{S}_k^{-1} \mathbf{v}_k \sim \mathcal{N}(Kn_y, 2Kn_y)$$

Within 3σ-region?





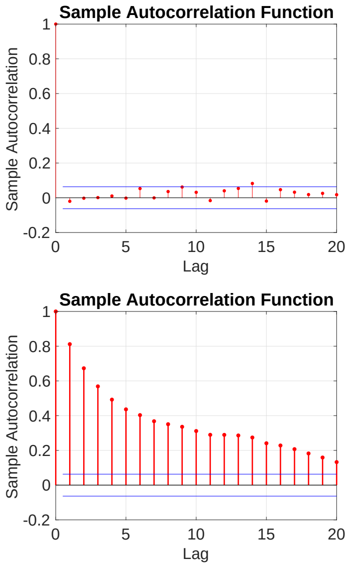
TEST OF INNOVATION PROPERTIES – CORRELATION

Whiteness

- Estimate the autocorrelation function (autocov. normalised to 1 at lag 0):

$$\rho(l) = \frac{\sum_{k=l+1}^{K} \mathbf{v}_k^T \mathbf{v}_{k-l}}{\sum_{\tau=l+1}^{K} \mathbf{v}_\tau^T \mathbf{v}_\tau}$$

and check if $\rho(l) \approx 0$ for $l > 0$.



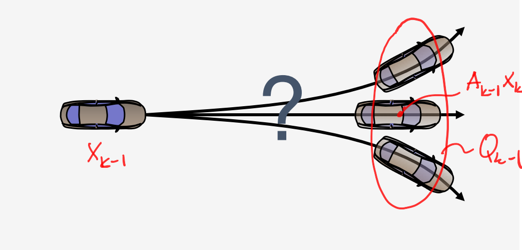
Kalman filter tuning and consistency – Motion and measurement models

Sensor fusion & nonlinear filtering

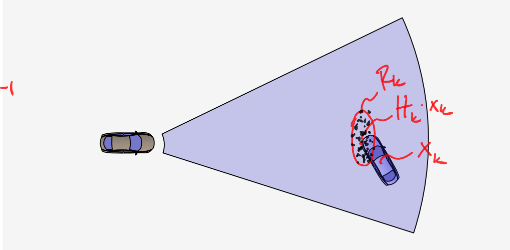
Lars Hammarstrand

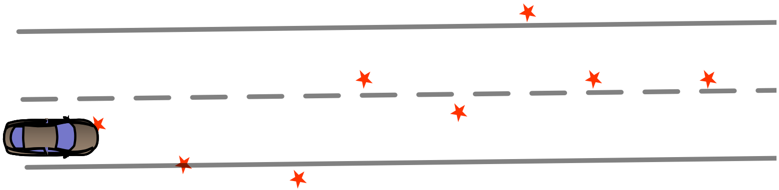
TUNING MOTION AND MEASUREMENT NOISE COVARIANCES

$p(\mathbf{x}_k|\mathbf{x}_{k-1}) = \mathcal{N}(\mathbf{x}_k; \mathbf{A}_{k-1}\mathbf{x}_{k-1}, \mathbf{Q}_{k-1})$:



$p(\mathbf{y}_k|\mathbf{x}_k) = \mathcal{N}(\mathbf{y}_k; \mathbf{H}_k\mathbf{x}_k, \mathbf{R}_k)$:





TUNING MOTION AND MEASUREMENT NOISE COVARIANCES

- A key aspect in tuning is to select the SNR $\|\mathbf{Q}\|/\|\mathbf{R}\|$:
  - If SNR is large $\Rightarrow$ a quickly adapting filter that relies more on new data than predictions.
  - If SNR is low $\Rightarrow$ the data is noise and we rely more on the predictions, the filter thus adapts slowly to data.
- The sensor noise, $\mathbf{R}$, is often described by the manufacturer and/or possible to collect data from which it can be estimated.
- The motion noise, $\mathbf{Q}$, is then selected by tuning.
- Unless you know the state sequence, study properties of the innovation to guide the tuning of the filter.

SELF-ASSESSMENT

If we design our filter such that the motion noise $\|\mathbf{Q}\|$ is small and the measurement noise $\|\mathbf{R}\|$ is large we get:

- a filter that adapts quickly to changes.
- a filter that adapts slowly to changes.
- we cannot select $\|\mathbf{Q}\|$ and $\|\mathbf{R}\|$ ourselves since they depend on the real system.

Check all that apply.

The Kalman filter and LMMSE estimators

Sensor fusion & nonlinear filtering

Lars Hammarstrand

THE KALMAN FILTER IS AN LMMSE ESTIMATOR

- The Kalman filter computes

$$\hat{\mathbf{x}}_{k|k-1} = \mathbb{E}\{\mathbf{x}_k | \mathbf{y}_{1:k-1}\}$$
$$\hat{\mathbf{x}}_{k|k} = \mathbb{E}\{\mathbf{x}_k | \mathbf{y}_{1:k}\}$$

$$\hat{x}_{k|k}(y_{1:k}) = \hat{x}_{k|k-1}(y_{1:k-1}) + K_k(y_k - \hat{y}_{k|k-1}(y_{1:k-1}))$$

  for linear and Gaussian models.

- Note:
  1. The Kalman filter is a linear function of $\mathbf{y}_{1:k}$.
  2. $\hat{\mathbf{x}}_{k|k}$ is the minimum mean square error (MMSE) estimator.

$\Rightarrow$ The Kalman filter is the linear minimum mean square error (LMMSE) estimator!

LMMSE ESTIMATION

LMMSE objective (static example)

- Find $\mathbf{A}$ and $\mathbf{b}$ such that $\hat{\mathbf{x}} = \mathbf{A}\mathbf{y} + \mathbf{b}$ yields the smallest possible MSE, $\mathbb{E}\{(\mathbf{x} - \hat{\mathbf{x}})^T(\mathbf{x} - \hat{\mathbf{x}})\}$.

- Finding optimum:
  Setting derivatives of MSE with respect to $\mathbf{b}$ and $\mathbf{A}$ to $\mathbf{0}$ yields

$$\mathbf{b} = \bar{\mathbf{x}} - \mathbf{A}\bar{\mathbf{y}}$$

$$\mathbf{A} = \mathbf{P}_{xy}\mathbf{P}_{yy}^{-1}.$$

$$\Rightarrow \hat{x} = Ay + \bar{x} - A\bar{y} = \bar{x} + A(y - \bar{y})$$

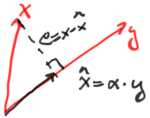
$$= \bar{x} + P_{xx} \cdot P_{yy}^{-1}(y - \bar{y})$$

- Orthogonality principle:
  select $\mathbf{A}$ such that $\mathbb{E}\{(\mathbf{x} - \hat{\mathbf{x}})\mathbf{y}^T\} = \mathbf{0}$.

$$\Leftrightarrow P_{xy} - A \cdot P_{yy} = 0$$

$$\Leftrightarrow A = P_{xy} \cdot P_{yy}^{-1}$$

$E\{xy\}$: inner product $\Rightarrow E\{xy\} = 0 \Leftrightarrow x \perp y$



$x - \hat{x} \perp y \Leftrightarrow E\{(x - \hat{x})y\} = 0$

SELF ASSESSMENT

What is different in LMMSE estimation compared to MMSE estimation.

- In LMMSE estimation we restrict the estimator to be a linear (or at least affine) function of data (measurements).
- In LMMSE the noise cannot be Gaussian.
- In MMSE estimation we normally compute a posterior distribution conditioned on data.

Check all that apply.

SEQUENTIAL LMMSE IN THE DYNAMIC CASE

LMMSE objective

- Sequentially find $\{\mathbf{L}_{k|k-1}, \mathbf{b}_{k|k-1}\}$ and $\{\mathbf{L}_{k|k}, \mathbf{b}_{k|k}\}$ such that

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{L}_{k|k-1}\underline{\mathbf{y}_{1:k-1}} + \mathbf{b}_{k|k-1}$$
$$\hat{\mathbf{x}}_{k|k} = \mathbf{L}_{k|k}\underline{\mathbf{y}_{1:k}} + \mathbf{b}_{k|k}$$

  minimize the MSE, $\mathbb{E}\{(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})^T(\cdot)\}$ and $\mathbb{E}\{(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k})^T(\cdot)\}$.

Note:

- Find a linear mapping based on all the data up to the relevant time.
- We generalise and allow us to consider affine functions of data.

LMMSE FOR LINEAR STATE SPACE MODELS

Linear state space model with additive (non-Gaussian) noise

- Consider state space model

$$\mathbf{x}_k = \mathbf{A}_{k-1}\mathbf{x}_{k-1} + \mathbf{q}_{k-1},$$
$$\mathbf{y}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{r}_k$$

where $\mathbf{x}_0$, $\mathbf{q}_{k-1}$ and $\mathbf{r}_k$ are independed random variables with known mean and covariances.

Key results (Additive non-Gaussian noise)

- The Kalman filter gives LMMSE estimates, $\hat{\mathbf{x}}_{k|k-1}$ and $\hat{\mathbf{x}}_{k|k}$, with the correct error covariances

$$\mathbf{P}_{k|k-1} = \mathbb{E}\{(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})(\cdot)^T\}$$
$$\mathbf{P}_{k|k} = \mathbb{E}\{(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k})(\cdot)^T\}$$

PROOF OUTLINE

- Assumption: $\begin{cases} \mathbb{E}\left[(\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1|k-1})\mathbf{y}_{1:k-1}^T\right] = 0 \\ \mathbb{E}\left[(\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1|k-1})(\cdot)^T\right] = \mathbf{P}_{k-1|k-1} \end{cases}$

Prediction	Update
$\mathbb{E}\left[(\mathbf{x}_k - \hat{\mathbf{x}}_{k\|k-1})\mathbf{y}_{1:k-1}^T\right] = 0$ $\mathbb{E}\left[(\mathbf{x}_k - \hat{\mathbf{x}}_{k\|k-1})(\cdot)^T\right] = \mathbf{P}_{k\|k-1}$	$\mathbb{E}\left[(\mathbf{x}_k - \hat{\mathbf{x}}_{k\|k})\mathbf{y}_{1:k-1}^T\right] = \mathbf{0}$ $\mathbb{E}\left[(\mathbf{x}_k - \hat{\mathbf{x}}_{k\|k})\mathbf{y}_k^T\right] = \mathbf{0}$ $\mathbb{E}\left[(\mathbf{x}_k - \hat{\mathbf{x}}_{k\|k})(\mathbf{x}_k - \hat{\mathbf{x}}_{k\|k})^T\right] = \mathbf{P}_{k\|k}$

- DIY: Make the proof for the scalar case where x_0, q_{k-1}, r_k are zero mean.

SELF ASSESSMENT

Fact:

For linear state space models with additive noise, the Kalman filter computes the LMMSE estimate recursively, also when the noise is not Gaussian.

Statement for you to verify or reject:

However, the Kalman filter is merely the best linear estimator among all *recursive* algorithms and we can sometime do better if we consider all measurements at the same time.

- True.
- False