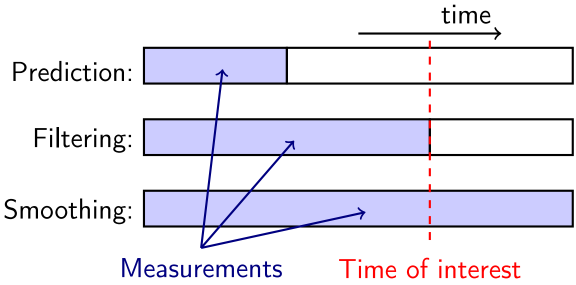
The smoothing problem

Introduction and motivation

Lennart Svensson

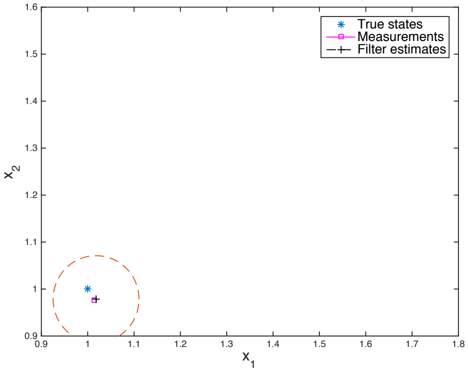
Signal Processing Group
Department of Signals and Systems
Chalmers University of Technology, Sweden



Three versions of smoothing:

1. **Fixed-interval smoothing:** find $p(\mathbf{x}_k | \mathbf{y}_{1:K})$ for all k and a fixed K. *Estimate a trajectory given a measurement sequence.*
2. **Fixed-lag smoothing:** compute $p(\mathbf{x}_k | \mathbf{y}_{1:k+n})$ for all k and a fixed $n > 0$. *Resembles filtering, but we accept a small time delay on our estimates.*
3. **Fixed-point smoothing:** compute $p(\mathbf{x}_k | \mathbf{y}_{1:K})$ at a fixed time k as we collect more data (as K grows).

- **Example:** Suppose we collect noisy observations of an object moving in 2D.
- Let us compare optimal solutions to filtering and smoothing. **Note**: in smoothing we also have data from later times.



Smoothing applications

- Survaillance of, e.g., airports is important for safety reasons.

  $\mathbf{x}_k$: positions of people, bags, etc.

- Other examples:
  - **Communication systems:** having received a complete message you try to decode it.
  - **Sports:** determine where a ball bounced, if someone cheated...
  - **Mapping:** collect data from many vehicles in order to construct a detailed map.
  - **Medicine:** e.g., use sequences of arterial blood pressure to estimate the intracranial pressure.

- Why do we care? Because smoothing can often improve the accuracy considerably compared to filtering!

Check all that apply!

- The objective with smoothing is to compute the distribution of future data.
- In smoothing, we try to estimate a state vector at some time of interest, based on measurements collect before and after that time.
- In smoothing, we only make use of future data, since these are more informative.
- In fixed-lag smoothing we have access to data from a certain time interval and wish to estimate the state sequence in the same interval.

Optimal smoothing

Forward-backward smoothing and two-filter smoothing

Lennart Svensson

Signal Processing Group
Department of Signals and Systems
Chalmers University of Technology, Sweden

Models and optimal solutions

State space models

- We mainly consider models of the type

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

Optimal filtering solution

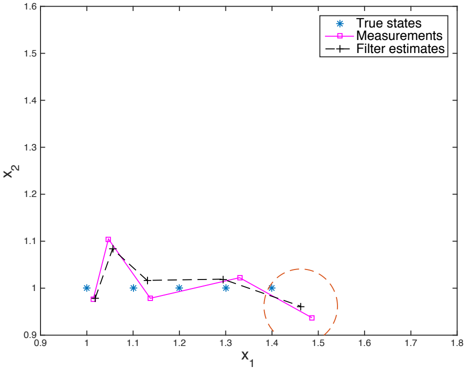
- For filtering we use prediction and update recursions:

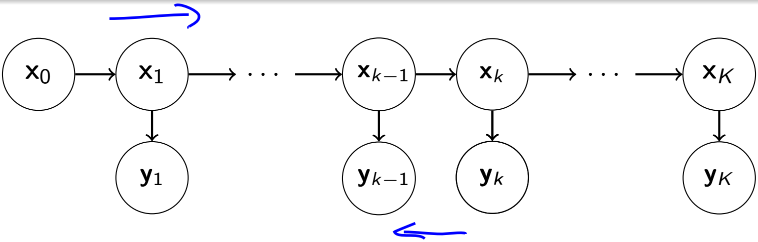
Prediction: $$p(\mathbf{x}_k \mid \mathbf{y}_{1:k-1}) = \int p(\mathbf{x}_k \mid \mathbf{x}_{k-1}) p(\mathbf{x}_{k-1} \mid \mathbf{y}_{1:k-1})\, d\mathbf{x}_{k-1}$$

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

- What is the **optimal solution for smoothing?**

- **Example (revisited):** Suppose we collect noisy observations of an object moving in 2D, and that we model its motion using a constant velocity model.





Two-filter smoothing

1) **Forward filtering:** compute $p(\mathbf{x}_i | \mathbf{y}_{1:i})$ for $i = 1, \ldots, k$.
2) **Backward "filtering":** compute $p(\mathbf{y}_{i+1:K} | \mathbf{x}_i)$ for $i = K, \ldots, k$.
3) **Smoothing solution:** $p(\mathbf{x}_k | \mathbf{y}_{1:K}) \propto p(\mathbf{x}_k | \mathbf{y}_{1:k}) p(\mathbf{y}_{k+1:K} | \mathbf{x}_k)$.

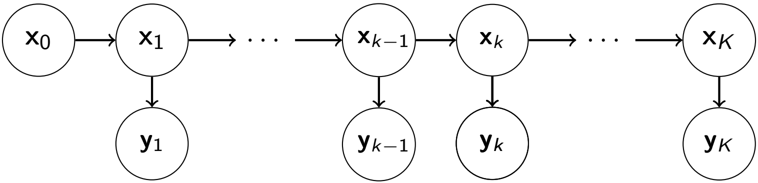
prior likelihood

$$p(x_k | y_{1:K}) = p(x_k | y_{1:k}, y_{k+1:K})$$

$$\propto p(y_{k+1:K} | x_k, \cancel{y_{1:k}})\, p(x_k | y_{1:k}) \qquad \cancel{p(y_{k+1:K} | y_{1:k})}$$

Two optimal smoothing solutions – FBS

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



Forward-backward smoothing

1) **Forward filtering:** compute $p(\mathbf{x}_k | \mathbf{y}_{1:k})$ for $k = 1, \ldots, K$.
2) **Backward smoothing:** compute $p(\mathbf{x}_k | \mathbf{y}_{1:K})$ for $k = K-1, \ldots, 1$.

- Both the Rauch-Tung-Striebel smoother and the Gaussian smoothers can be viewed as forward-backward smoothing algorithms.

- In this lecture we focus on solving **fixed interval smoothing** problems using **forward-backward smoothing**.

- **Objective:** compute $p(\mathbf{x}_k | \mathbf{y}_{1:K})$ for $k = 1, \ldots, K$.

We seek $p(x_k | y_{1:K})$, Assume $p(x_k|y_{1:k})$, $p(x_{k+1}|y_{1:k})$ and $p(x_{k+1}|y_{1:K})$ are known.

Optimal backward smoothing

- For computing $p(\mathbf{x}_k | \mathbf{y}_{1:K})$ recursively for $k = K-1, \ldots, 1$:

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

$$p(x_k | y_{1:K}) = \int p(x_k, x_{k+1} | y_{1:K}) \, dx_{k+1}$$

$\uparrow$ Difficulty: condition on $y_{k+1:K}$

$$p(x_k, x_{k+1} | y_{1:K}) = p(x_k | x_{k+1}, y_{1:K}) p(x_{k+1} | y_{1:K})$$

$$p(x_k | x_{k+1}, y_{1:k}) = \frac{p(x_{k+1} | x_k, \cancel{y_{1:k}}) p(x_k | y_{1:k})}{p(x_{k+1} | y_{1:k})}$$

Check all that apply!

- Using the forward-backward algorithm, we can do fixed interval smoothing by first running a conventional filter from time 1 to K.
- In a two filter smoothing algorithm we essentially compute the square of a conventional forward filter.
- In forward-backward smoothing, we run a conventional filter from time K to 1.

Rauch-Tung-Striebel smoothing

Lennart Svensson

Signal Processing Group
Department of Signals and Systems
Chalmers University of Technology, Sweden

$p(x_{1:K}, y_{1:K})$ is Gaussian $\Rightarrow p(x_{1:K} | y_{1:K})$ is Gaussian
$\Rightarrow p(x_k | y_{1:K})$ is Gaussian

- **Linear and Gaussian models**

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

  are important also in smoothing.

- For such models, the posterior distribution is Gaussian,

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

  for all k and K.

- It is possible to find $\hat{\mathbf{x}}_{k|K}$ and $\mathbf{P}_{k|K}$ analytically for all k and K, i.e., for all filtering, prediction and smoothing problems.

Rauch-Tung-Striebel smoothing

- Kalman did not present the solution to the smoothing problem.
- The smoothing algorithm, for linear and Gaussian models, was instead developed by Rauch, Tuch and Striebel. The algorithm is known as the **Rauch-Tung-Striebel (RTS) smoother**.

The RTS algorithm

1) Run a Kalman filter for $k = 1, \ldots, K$ and store the moments $\hat{\mathbf{x}}_{k|k}$, $\mathbf{P}_{k|k}$, $\hat{\mathbf{x}}_{k+1|k}$ and $\mathbf{P}_{k+1|k}$.
2) For $k = K-1, \ldots, 1$ compute

$$\mathbf{G}_k = \mathbf{P}_{k|k}\mathbf{A}_k^T\mathbf{P}_{k+1|k}^{-1}$$

$$\hat{\mathbf{x}}_{k|K} = \hat{\mathbf{x}}_{k|k} + \mathbf{G}_k\left(\hat{\mathbf{x}}_{k+1|K} - \hat{\mathbf{x}}_{k+1|k}\right)$$

$$\mathbf{P}_{k|K} = \mathbf{P}_{k|k} - \mathbf{G}_k\left[\mathbf{P}_{k+1|k} - \mathbf{P}_{k+1|K}\right]\mathbf{G}_k^T$$

RTS smoothing vs Kalman filtering

- Backward smoothing (RTS) and forward filtering (Kalman) are very similar:

$y_{1:k-1}$

y_k

$$\text{Forward:}\quad \begin{cases} \mathbf{K}_k = \overbrace{\mathbf{P}_{k|k-1}\mathbf{H}_k^T}^{Cov(x_k, y_k | y_{1:k-1})}\mathbf{S}_k^{-1} \\ \hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\left(\mathbf{y}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k|k-1}\right) \\ \mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^T \end{cases}$$

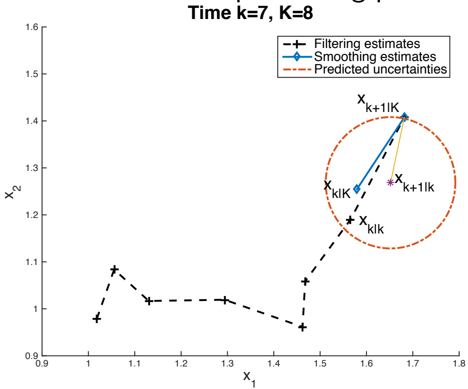
$y_{1:k}$

$y_{k+1:K}$

$$\text{Backward:}\quad \begin{cases} \mathbf{G}_k = \overbrace{\mathbf{P}_{k|k}\mathbf{A}_k^T}^{Cov(x_k, x_{k+1} | y_{1:k})}\mathbf{P}_{k+1|k}^{-1} \\ \hat{\mathbf{x}}_{k|K} = \hat{\mathbf{x}}_{k|k} + \mathbf{G}_k\left(\hat{\mathbf{x}}_{k+1|K} - \hat{\mathbf{x}}_{k+1|k}\right) \\ \mathbf{P}_{k|K} = \mathbf{P}_{k|k} - \mathbf{G}_k\left[\mathbf{P}_{k+1|k} - \mathbf{P}_{k+1|K}\right]\mathbf{G}_k^T \end{cases}$$

- Translations:
  - Gain: $\mathbf{K}_k \longleftrightarrow \mathbf{G}_k$,
  - Innovation: $\mathbf{y}_k - \underbrace{\mathbf{H}_k\hat{\mathbf{x}}_{k|k-1}}_{\hat{y}_{k|k-1}} \longleftrightarrow \hat{\mathbf{x}}_{k+1|K} - \hat{\mathbf{x}}_{k+1|k}$,
  - etc.

- **Example:** let us revisit the positioning problem in 2D.



$$\mathbf{G}_k = \mathbf{P}_{k|k} \mathbf{A}_k^T \mathbf{P}_{k+1|k}^{-1}$$

$$\hat{\mathbf{x}}_{k|K} = \hat{\mathbf{x}}_{k|k} + \mathbf{G}_k \left(\hat{\mathbf{x}}_{k+1|K} - \hat{\mathbf{x}}_{k+1|k} \right)$$

$$\mathbf{P}_{k|K} = \mathbf{P}_{k|k} - \mathbf{G}_k \left[\mathbf{P}_{k+1|k} - \mathbf{P}_{k+1|K} \right] \mathbf{G}_k^T$$

Check all that apply!

- The RTS smoother is a linear technique to compute the posterior mean and covariance of $\mathbf{x}_k$ given $\mathbf{y}_{1:K}$.
- The (forward) filter computes $p(\mathbf{x}_k \middle| \mathbf{y}_{1:K})$ for $k = K$.
- The RTS smoother computes the MMSE estimator of $\mathbf{x}_k$ given $\mathbf{y}_{1:K}$.

Rauch-Tung-Striebel smoothing

The derivations!

Lennart Svensson

Signal Processing Group
Department of Signals and Systems
Chalmers University of Technology, Sweden

Summary of RTS smoothing

- For **linear and Gaussian models**

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

  it holds that $p(\mathbf{x}_k \big| \mathbf{y}_{1:T}) = \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|T}, \mathbf{P}_{k|T})$.

- The RTS algorithm computes its $\hat{\mathbf{x}}_{k|T}$ and $\mathbf{P}_{k|T}$ analytically.

The RTS algorithm

1) Run a Kalman filter for $k = 1, \ldots, T$ and store the moments $\hat{\mathbf{x}}_{k|k}$, $\mathbf{P}_{k|k}$, $\hat{\mathbf{x}}_{k+1|k}$ and $\mathbf{P}_{k+1|k}$.
2) For $k = T-1, \ldots, 1$ compute

$$\mathbf{G}_k = \mathbf{P}_{k|k}\mathbf{A}_k^T\mathbf{P}_{k+1|k}^{-1}$$
$$\hat{\mathbf{x}}_{k|T} = \hat{\mathbf{x}}_{k|k} + \mathbf{G}_k\left(\hat{\mathbf{x}}_{k+1|T} - \hat{\mathbf{x}}_{k+1|k}\right)$$
$$\mathbf{P}_{k|T} = \mathbf{P}_{k|k} - \mathbf{G}_k\left[\mathbf{P}_{k+1|k} - \mathbf{P}_{k+1|T}\right]\mathbf{G}_k^T$$

- **Objective:** prove that the backward recursion is correct.

Derivation strategy

- **Idea 1 (step 3):** to find $p(\mathbf{x}_k|\mathbf{y}_{1:T})$ we first find $p(\mathbf{x}_k,\mathbf{x}_{k+1}|\mathbf{y}_{1:T})$.
  - To get the marginal density we can use
  $$p(\mathbf{x}_k|\mathbf{y}_{1:T}) = \int p(\mathbf{x}_k,\mathbf{x}_{k+1}|\mathbf{y}_{1:T})\,d\mathbf{x}_{k+1},$$
  which is trivial when $\mathbf{x}_k,\mathbf{x}_{k+1}|\mathbf{y}_{1:T}$ is jointly Gaussian.

- **Idea 2 (step 2):** the joint density can be factorized as
$$\begin{aligned}p(\mathbf{x}_k,\mathbf{x}_{k+1}|\mathbf{y}_{1:T}) &= p(\mathbf{x}_k|\mathbf{x}_{k+1},\mathbf{y}_{1:T})p(\mathbf{x}_{k+1}|\mathbf{y}_{1:T})\\ &= \underbrace{p(\mathbf{x}_k|\mathbf{x}_{k+1},\mathbf{y}_{1:k})}_{\text{difficult}}\mathcal{N}(\mathbf{x}_{k+1};\hat{\mathbf{x}}_{k+1|T},\mathbf{P}_{k+1|T})\end{aligned}$$

- **Idea 3 (step 1):** to find $p(\mathbf{x}_k|\mathbf{x}_{k+1},\mathbf{y}_{1:k})$ we first identify
$$p(\mathbf{x}_k,\mathbf{x}_{k+1}|\mathbf{y}_{1:k}) = p(\mathbf{x}_{k+1}|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_{1:k})$$
and then use the lemma on conditional Gaussian densities.

Product of Gaussian densities (step 1)

- **Objective:** find $p(\mathbf{x}_k, \mathbf{x}_{k+1} | \mathbf{y}_{1:k}) = p(\mathbf{x}_{k+1} | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{y}_{1:k})$.
- We have

$$p(\mathbf{x}_{k+1} | \mathbf{x}_k) = \mathcal{N}(\mathbf{x}_{k+1}; \mathbf{A}_k \mathbf{x}_k, \mathbf{Q}_k) \quad \Leftrightarrow \quad \begin{cases} \mathbf{x}_{k+1} = \mathbf{A}_k \mathbf{x}_k + \mathbf{q}_k \\ \mathbf{q}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{q}_k) \end{cases}$$

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

  which implies that $p(\mathbf{x}_k, \mathbf{x}_{k+1} | \mathbf{y}_{1:k}) =$

  - $\mathcal{N}\left(\begin{bmatrix} \mathbf{x}_k \\ \mathbf{x}_{k+1} \end{bmatrix}; \begin{bmatrix} \mathbf{x}_k \\ \mathbf{x}_{k+1} \end{bmatrix}, \begin{bmatrix} \mathbf{P}_{k|k} & \mathbf{P}_{k|k} + \mathbf{Q}_k \\ \mathbf{P}_{k|k} + \mathbf{Q}_k & \mathbf{P}_{k|k} \mathbf{A}_k^2 + \mathbf{Q}_k \end{bmatrix} \right)$
  - $\mathcal{N}\left(\begin{bmatrix} \mathbf{x}_k \\ \mathbf{x}_{k+1} \end{bmatrix}; \begin{bmatrix} \hat{\mathbf{x}}_{k|k} \\ \hat{\mathbf{x}}_{k+1|T} \end{bmatrix}, \begin{bmatrix} \mathbf{P}_{k|k} & \mathbf{P}_{k|T} \\ \mathbf{P}_{k|T} & \mathbf{P}_{k+1|T} \end{bmatrix} \right)$
  - $\mathcal{N}\left(\begin{bmatrix} \mathbf{x}_k \\ \mathbf{x}_{k+1} \end{bmatrix}; \begin{bmatrix} \hat{\mathbf{x}}_{k|k} \\ \mathbf{A}_k \hat{\mathbf{x}}_{k|k} \end{bmatrix}, \begin{bmatrix} \mathbf{P}_{k|k} & \mathbf{P}_{k|k} \mathbf{A}_k^T \\ \mathbf{A}_k \mathbf{P}_{k|k} & \mathbf{A}_k \mathbf{P}_{k|k} \mathbf{A}_k^T + \mathbf{Q}_k \end{bmatrix} \right)$

- Please identify the correct answer.

Product of Gaussian densities (step 1)

- **Objective:** find $p(\mathbf{x}_k, \mathbf{x}_{k+1} | \mathbf{y}_{1:k}) = p(\mathbf{x}_{k+1} | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{y}_{1:k})$.
- We have

$$p(\mathbf{x}_{k+1} | \mathbf{x}_k) = \mathcal{N}(\mathbf{x}_{k+1}; \mathbf{A}_k \mathbf{x}_k, \mathbf{Q}_k) \quad \Leftrightarrow \quad \begin{cases} \mathbf{x}_{k+1} = \mathbf{A}_k \mathbf{x}_k + \mathbf{q}_k \\ \mathbf{q}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{q}_k) \end{cases}$$

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

which implies that

$P_{k+1|k}$

$$p(\mathbf{x}_k, \mathbf{x}_{k+1} | \mathbf{y}_{1:k}) = \mathcal{N}\left(\begin{bmatrix} \mathbf{x}_k \\ \mathbf{x}_{k+1} \end{bmatrix}; \begin{bmatrix} \hat{\mathbf{x}}_{k|k} \\ \mathbf{A}_k \hat{\mathbf{x}}_{k|k} \end{bmatrix}, \begin{bmatrix} \mathbf{P}_{k|k} & \mathbf{P}_{k|k} \mathbf{A}_k^T \\ \mathbf{A}_k \mathbf{P}_{k|k} & \mathbf{A}_k \mathbf{P}_{k|k} \mathbf{A}_k^T + \mathbf{Q}_k \end{bmatrix} \right)$$

- **Proof:** $\begin{bmatrix} x_k \\ x_{k+1} \end{bmatrix} = \begin{bmatrix} I \\ A_k \end{bmatrix} x_k + \begin{bmatrix} 0 \\ I \end{bmatrix} q_k$ independent

$$\Rightarrow E\left\{ \begin{bmatrix} x_k \\ x_{k+1} \end{bmatrix} \middle| y_{1:k} \right\} = \begin{bmatrix} I \\ A_k \end{bmatrix} \hat{x}_{k|k} + \begin{bmatrix} 0 \\ I \end{bmatrix} 0$$

$$\mathrm{Cov}\left\{ \begin{bmatrix} x_k \\ x_{k+1} \end{bmatrix} \middle| y_{1:k} \right\} = \begin{bmatrix} I \\ A_k \end{bmatrix} P_{k|k} \begin{bmatrix} I & A_k^T \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} Q_k \begin{bmatrix} 0 & I \end{bmatrix}$$

- **Objective:** find $p(\mathbf{x}_k|\mathbf{x}_{k+1}, \mathbf{y}_{1:k})$ when

$$\underbrace{p(\mathbf{x}_k}_{x}, \underbrace{\mathbf{x}_{k+1}}_{y}|\mathbf{y}_{1:k}) = \mathcal{N}\left(\begin{bmatrix}\mathbf{x}_k \\ \mathbf{x}_{k+1}\end{bmatrix}; \begin{bmatrix}\overset{\mu_x}{\hat{\mathbf{x}}_{k|k}} \\ \underset{\mu_y}{\mathbf{A}_k\hat{\mathbf{x}}_{k|k}}\end{bmatrix}, \begin{bmatrix}\overset{P_{xx}}{\mathbf{P}_{k|k}} & \overset{P_{xy}}{\mathbf{P}_{k|k}\mathbf{A}_k^T} \\ \mathbf{A}_k\mathbf{P}_{k|k} & \underset{P_{yy}}{\mathbf{P}_{k+1|k}}\end{bmatrix}\right)$$

Conditional distribution of Gaussian variables

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

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

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

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

- Using the notation $\mathbf{G}_k = \overbrace{\mathbf{P}_{k|k}\mathbf{A}_k^T\mathbf{P}_{k+1|k}^{-1}}^{P_{xy}P_{yy}^{-1}}$, we get

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

Product of Gaussian densities (step 2)

- **Objective, step 2:** find

$$p(\mathbf{x}_k, \mathbf{x}_{k+1} | \mathbf{y}_{1:T}) = p(\mathbf{x}_k | \mathbf{x}_{k+1}, \mathbf{y}_{1:k}) p(\mathbf{x}_{k+1} | \mathbf{y}_{1:T}),$$

where step 1 gave us that

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

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

$+ G_k P_{k+1|T} G_k^T$

- Similar to the previous product of Gaussian densities:

$$\mathbf{x}_k, \mathbf{x}_{k+1} | \mathbf{y}_{1:T} \sim \mathcal{N}(\bar{\mathbf{x}}, \mathbf{P})$$

where

$$\bar{\mathbf{x}} = \begin{bmatrix} \hat{\mathbf{x}}_{k|k} + \mathbf{G}_k(\hat{\mathbf{x}}_{k+1|T} - \hat{\mathbf{x}}_{k+1|k}) \\ \hat{\mathbf{x}}_{k|T} \end{bmatrix}$$

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{k|k} - \mathbf{G}_k \left(\mathbf{P}_{k+1|k} - \mathbf{P}_{k+1|T}\right) \mathbf{G}_k^T & \mathbf{P}_{k+1|T} \mathbf{A}_k^T \\ \mathbf{A}_k \mathbf{P}_{k+1|T} & \mathbf{P}_{k+1|T} \end{bmatrix}$$

Computing $p(\mathbf{x}_k | \mathbf{y}_{1:T})$ (step 3)

- **Objective, step 3:** find $p(\mathbf{x}_k | \mathbf{y}_{1:T})$ when

$$\begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \end{bmatrix}, \quad \begin{bmatrix} P_{11} & P_{12} \\ P_{12}^T & P_{22} \end{bmatrix}$$

$$p\left(\begin{bmatrix} \mathbf{x}_k \\ \mathbf{x}_{k+1} \end{bmatrix} \middle| \mathbf{y}_{1:T} \right) = \mathcal{N}\left(\begin{bmatrix} \mathbf{x}_k \\ \mathbf{x}_{k+1} \end{bmatrix}; \bar{\mathbf{x}}, \mathbf{P} \right).$$

- **Short derivation:** to find the marginal density we note that

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x}_k \\ \mathbf{x}_{k+1} \end{bmatrix}$$

  which implies that

$$\mathbb{E}[\mathbf{x}_k | \mathbf{y}_{1:T}] = \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix} \bar{\mathbf{x}}_k$$

$$\mathrm{Cov}[\mathbf{x}_k | \mathbf{y}_{1:T}] = \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix} \mathbf{P} \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix}^T$$

- **Conclusion:** $p(\mathbf{x}_k | \mathbf{y}_{1:T})$ is Gaussian with the moments

$$\hat{\mathbf{x}}_{k|T} = \hat{\mathbf{x}}_{k|k} + \mathbf{G}_k(\hat{\mathbf{x}}_{k+1|T} - \hat{\mathbf{x}}_{k+1|k})$$

$$\mathbf{P}_{k|T} = \mathbf{P}_{k|k} - \mathbf{G}_k\left(\mathbf{P}_{k+1|k} - \mathbf{P}_{k+1|T}\right)\mathbf{G}_k^T$$

General Gaussian smoothing

Lennart Svensson

Signal Processing Group
Department of Signals and Systems
Chalmers University of Technology, Sweden

- Suppose that we have a nonlinear model

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

Gaussian smoothing

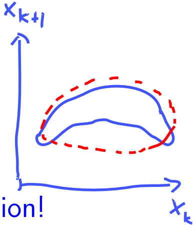
1. Run a Gaussian filter for $k = 1, \ldots, T$ and store the moments $\hat{\mathbf{x}}_{k|k}$, $\mathbf{P}_{k|k}$, $\hat{\mathbf{x}}_{k+1|k}$ and $\mathbf{P}_{k+1|k}$.
2. Run a Gaussian backward smoother from $k = T - 1$ to $k = 1$, that computes $\hat{\mathbf{x}}_{k|T}$ and $\mathbf{P}_{k|T}$.

- How can we design a **Gaussian backward smoother**?

A backward recursion with Gaussian approximations

Assumptions: we already have Gaussian approximations to $p(\mathbf{x}_k | \mathbf{y}_{1:k})$, $p(\mathbf{x}_{k+1} | \mathbf{y}_{1:k})$ and $p(\mathbf{x}_{k+1} | \mathbf{y}_{1:T})$.

Objective: find a Gaussian approximation to $p(\mathbf{x}_k | \mathbf{y}_{1:T})$.



- **Step 1 (a):** approximate

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

  as jointly Gaussian. This is the only new approximation!

- **Step 1 (b):** use the lemma on conditional Gaussian densities to obtain

$$p(\mathbf{x}_k | \mathbf{x}_{k+1}, \mathbf{y}_{1:k}).$$

- **Step 2:** identify the joint density

$$p(\mathbf{x}_k, \mathbf{x}_{k+1} | \mathbf{y}_{1:T}) = p(\mathbf{x}_k | \mathbf{x}_{k+1}, \mathbf{y}_{1:T}) p(\mathbf{x}_{k+1} | \mathbf{y}_{1:T})$$

- **Step 3:** if $p(\mathbf{x}_k, \mathbf{x}_{k+1} | \mathbf{y}_{1:T})$ is jointly Gaussian, it is now trivial to find $p(\mathbf{x}_k | \mathbf{y}_{1:T})$.

Gaussian approximation in smoothing

Strategy

Approximate $\mathbf{x}_k, \mathbf{x}_{k+1} | \mathbf{y}_{1:k}$ as jointly Gaussian using moment matching

known ← ← unknown

$$\begin{bmatrix} \mathbf{x}_k \\ \mathbf{x}_{k+1} \end{bmatrix} \Bigg| \mathbf{y}_{1:k} \sim \mathcal{N}\left(\begin{bmatrix} \hat{\mathbf{x}}_{k|k} \\ \hat{\mathbf{x}}_{k+1|k} \end{bmatrix}, \begin{bmatrix} \mathbf{P}_{k|k} & \mathbf{P}_{k,k+1|k} \\ \mathbf{P}_{k,k+1|k}^T & \mathbf{P}_{k+1|k} \end{bmatrix} \right)$$

→ known ←

- Based on this approximation we obtain the **backward recursions**

→ $P_{xy} P_{yy}^{-1}$, $P_{k|k} A_k^T P_{k+1|k}^{-1}$

$$\begin{aligned} \mathbf{G}_k &= \mathbf{P}_{k,k+1|k} \mathbf{P}_{k+1|k}^{-1} \\ \hat{\mathbf{x}}_{k|T} &= \hat{\mathbf{x}}_{k|k} + \mathbf{G}_k \left(\hat{\mathbf{x}}_{k+1|T} - \hat{\mathbf{x}}_{k+1|k} \right) \\ \mathbf{P}_{k|T} &= \mathbf{P}_{k|k} - \mathbf{G}_k \left[\mathbf{P}_{k+1|k} - \mathbf{P}_{k+1|T} \right] \mathbf{G}_k^T \end{aligned} \qquad (1)$$

$\rightsquigarrow$ *we do smoothing using moment matching and* (1).

- Apart from $\hat{\mathbf{x}}_{k|k}$, $\mathbf{P}_{k|k}$, $\hat{\mathbf{x}}_{k+1|k}$, $\mathbf{P}_{k+1|k}$, $\hat{\mathbf{x}}_{k+1|T}$ and $\mathbf{P}_{k+1|T}$ we need the **cross-covariance matrix**

$$\begin{aligned}\mathbf{P}_{k,k+1|k} &= \mathbb{E}\left[\left(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}\right)\left(\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1|k}\right)^T \middle| \mathbf{y}_{1:k}\right] \\ &= \mathbb{E}\left[\left(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}\right)\left(\mathbf{f}_k(\mathbf{x}_k) - \hat{\mathbf{x}}_{k+1|k}\right)^T \middle| \mathbf{y}_{1:k}\right].\end{aligned}$$

where $\mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k) + \mathbf{q}_k$ and $\mathbf{q}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$.

Remarks:

1. Expectations are taken with respect to the filtering density $p(\mathbf{x}_k \middle| \mathbf{y}_{1:k})$.
2. It is recommended to approximate $\hat{\mathbf{x}}_{k+1|k}$, $\mathbf{P}_{k+1|k}$ and $\mathbf{P}_{k,k+1|k}$ using the same moment matching technique (e.g., linearization or unscented transform)
   $\rightsquigarrow$ if you use different techniques during filtering and smoothing you should compute $\hat{\mathbf{x}}_{k+1|k}$ and $\mathbf{P}_{k+1|k}$ again during smoothing.

$E\{(x_k - \hat{x}_{k|k})(A_k x_k + q_{k-1} - A_k \hat{x}_{k|k})^T | y_{1:k}\} = E\{(x_k - \hat{x}_{k|k})(x_k - \hat{x}_{k|k})^T A_k^T | y_{1:k}\}$

$= P_{k|k} A_k^T$

- For models of the type

$$\mathbf{x}_k = \mathbf{A}_{k-1}\mathbf{x}_{k-1} + \mathbf{q}_{k-1}, \qquad \mathbf{q}_{k-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{k-1})$$
$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k) + \mathbf{r}_k, \qquad \mathbf{r}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k).$$

  it holds that

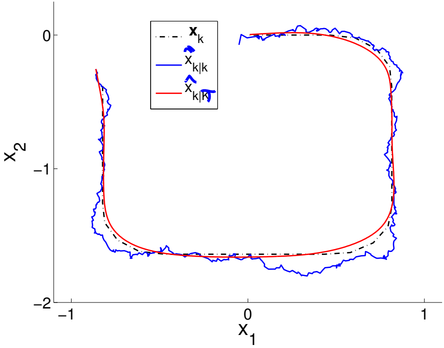
$$\begin{cases} \mathbf{P}_{k,k+1|k} & = \mathbf{P}_{k|k}\mathbf{A}_k^T \\ \mathbf{G}_k & = \mathbf{P}_{k,k+1|k}\mathbf{P}_{k+1|k}^{-1} \\ \hat{\mathbf{x}}_{k|T} & = \hat{\mathbf{x}}_{k|k} + \mathbf{G}_k\left(\hat{\mathbf{x}}_{k+1|T} - \hat{\mathbf{x}}_{k+1|k}\right) \\ \mathbf{P}_{k|T} & = \mathbf{P}_{k|k} - \mathbf{G}_k\left[\mathbf{P}_{k+1|k} - \mathbf{P}_{k+1|T}\right]\mathbf{G}_k^T \end{cases}$$

- **Remarks:**
  - no need to perform additional approximations during backward smoothing,
  - the equations for the backward recursions are identical to the RTS equations.

A Gaussian smoothing illustration

Example:

- Two angular sensors positioned at $(-1.5, 0.5)$ and $(1, 1)$. The standard deviation of the noise is 0.05 radians.
- Constant velocity model, continuous time variance 0.1 and sampling time $T = 0.01$.
- The figure shows $\mathbf{x}_k, \hat{\mathbf{x}}_{k|k}$ and $\hat{\mathbf{x}}_{k|T}$ in one simulation:



Check all the apply!

- In the backward recursions of a Gaussian smoother, we approximate $p(\mathbf{x}_k, \mathbf{x}_{k+1} \big| \mathbf{y}_{1:k})$ as jointly Gaussian.
- In Gaussian smoothing, the forward filter is an exact filter, but we introduce approximations in the backward recursions.
- If the measurement model is linear and Gaussian, we do not need to introduce any new approximations during the backward recursions.

Extended RTS, Unscented RTS, Cubature RTS and Gauss-Hermite RTS smoothing

Lennart Svensson

Signal Processing Group
Department of Signals and Systems
Chalmers University of Technology, Sweden

General Gaussian smoothing

- Suppose that we have a nonlinear model

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

Gaussian smoothing

1. Run a Gaussian filter for $k = 1, \ldots, T$ and store the moments $\hat{\mathbf{x}}_{k|k}$ and $\mathbf{P}_{k|k}$.
2. Run a Gaussian backward smoother from $k = T - 1$ to $k = 1$, that computes $\hat{\mathbf{x}}_{k|T}$ and $\mathbf{P}_{k|T}$.

- Though we could also store $\hat{\mathbf{x}}_{k+1|k}$ and $\mathbf{P}_{k+1|k}$, we now assume they are recomputed during the backward recursions.

Apart from the moments of $p(\mathbf{x}_k | \mathbf{y}_{1:k})$ and $p(\mathbf{x}_{k+1} | \mathbf{y}_{1:T})$, we need

$$\hat{\mathbf{x}}_{k+1|k} = \mathbb{E}\left[\mathbf{f}_k(\mathbf{x}_k) \middle| \mathbf{y}_{1:k}\right],$$

$x_{k+1} = f_k(x_k) + q_k,\ q_k \sim \mathcal{N}(0, Q_k)$

$$\mathbf{P}_{k+1|k} = \mathbf{Q}_k + \mathbb{E}\left[\left(\mathbf{f}_k(\mathbf{x}_k) - \hat{\mathbf{x}}_{k+1|k}\right)\left(\mathbf{f}_k(\mathbf{x}_k) - \hat{\mathbf{x}}_{k+1|k}\right)^T \middle| \mathbf{y}_{1:k}\right],$$

$$\mathbf{P}_{k,k+1|k} = \mathbb{E}\left[\left(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}\right)\left(\mathbf{f}_k(\mathbf{x}_k) - \hat{\mathbf{x}}_{k+1|k}\right)^T \middle| \mathbf{y}_{1:k}\right].$$

Backward recursions in a Gaussian filter

- We can now run the backward recursions

$$\begin{aligned} \mathbf{G}_k &= \mathbf{P}_{k,k+1|k}\mathbf{P}_{k+1|k}^{-1} \\ \hat{\mathbf{x}}_{k|T} &= \hat{\mathbf{x}}_{k|k} + \mathbf{G}_k\left(\hat{\mathbf{x}}_{k+1|T} - \hat{\mathbf{x}}_{k+1|k}\right) \\ \mathbf{P}_{k|T} &= \mathbf{P}_{k|k} - \mathbf{G}_k\left[\mathbf{P}_{k+1|k} - \mathbf{P}_{k+1|T}\right]\mathbf{G}_k^T \end{aligned} \tag{1}$$

$\rightsquigarrow$ *we do smoothing using moment matching and* (1).

Extended RTS smoothing

- An extended RTS smoother (ERTSS) assumes

$$\mathbf{x}_{k+1} \approx \mathbf{f}_k(\hat{\mathbf{x}}_{k|k}) + \mathbf{f}'_k(\hat{\mathbf{x}}_{k|k}) \left(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}\right) + \mathbf{q}_k$$

which implies that

$$\begin{cases} \hat{\mathbf{x}}_{k+1|k} & = \mathbf{f}_k(\hat{\mathbf{x}}_{k|k}) \\ \mathbf{P}_{k+1|k} & = \mathbf{f}'_k(\hat{\mathbf{x}}_{k|k})\mathbf{P}_{k|k}\mathbf{f}'_k(\hat{\mathbf{x}}_{k|k})^T + \mathbf{Q}_k \\ \mathbf{P}_{k,k+1|k} & = \mathbf{P}_{k|k}\mathbf{f}'_k(\hat{\mathbf{x}}_{k|k})^T \end{cases}$$

- We can easily combine these with (1) to obtain

$$\begin{aligned} \mathbf{G}_k &= \mathbf{P}_{k|k}\mathbf{f}'_k(\hat{\mathbf{x}}_{k|k})^T\mathbf{P}^{-1}_{k+1|k} \\ \hat{\mathbf{x}}_{k|K} &= \hat{\mathbf{x}}_{k|k} + \mathbf{G}_k \left(\hat{\mathbf{x}}_{k+1|K} - \mathbf{f}_k(\hat{\mathbf{x}}_{k|k})\right) \\ \mathbf{P}_{k|K} &= \mathbf{P}_{k|k} - \mathbf{G}_k \left[\mathbf{P}_{k+1|k} - \mathbf{P}_{k+1|K}\right] \mathbf{G}_k^T \end{aligned}$$

$x_k | y_{1:k} \sim \mathcal{N}(\hat{x}_{k|k}, P_{k|k})$

To perform Gauss-Hermite RTS smoothing:

1. Form a set of p^n σ-points ($n = \dim(\mathbf{x}_k)$, $p = \sharp$ points/dim)

$$\mathcal{X}_k^{(i_1,\ldots,i_n)} = \hat{\mathbf{x}}_{k|k} + \mathbf{P}_{k|k}^{1/2} \xi^{(i_1,i_2,\ldots,i_n)}, \quad i_1,\ldots,i_n = 1,\ldots,p.$$

2. Compute the desired moments

$$\hat{\mathbf{x}}_{k+1|k} \approx \sum_{i_1,\ldots,i_n=1}^{p} \mathbf{f}_k(\mathcal{X}_k^{(i_1,i_2,\ldots,i_n)}) \prod_{j=1}^{n} W_{i_j}$$

$$\mathbf{P}_{k+1|k} \approx \mathbf{Q}_k + \sum_{i_1,\ldots,i_n=1}^{p} (\mathbf{f}_k(\mathcal{X}_k^{(i_1,i_2,\ldots,i_n)}) - \hat{\mathbf{x}}_{k+1|k})(\cdot)^T \prod_{j=1}^{n} W_{i_j}$$

$$\mathbf{P}_{k,k+1|k} \approx \sum_{i_1,\ldots,i_n=1}^{p} \left(\mathcal{X}_k^{(i_1,\ldots,i_n)} - \hat{\mathbf{x}}_{k|k}\right)(\mathbf{f}_k(\mathcal{X}_k^{(i_1,\ldots,i_n)}) - \hat{\mathbf{x}}_{k+1|k})^T \prod_{j=1}^{n} W_{i_j}$$

3. Use (1) to do backward smoothing.

Unscented RTS smoothing

To perform unscented RTS smoothing:

1. Form a set of $2n+1$ σ-points

$$\mathcal{X}_k^{(0)} = \hat{\mathbf{x}}_{k|k}, \qquad W_0 = 1 - n/3, \quad W_i = 1/6, \; i > 1,$$
$$\mathcal{X}_k^{(i)} = \hat{\mathbf{x}}_{k|k} + \sqrt{3}\left(\mathbf{P}_{k|k}^{1/2}\right)_i, \qquad i = 1, 2, \ldots, n,$$
$$\mathcal{X}_k^{(i+n)} = \hat{\mathbf{x}}_{k|k} - \sqrt{3}\left(\mathbf{P}_{k|k}^{1/2}\right)_i, \qquad i = 1, 2, \ldots, n,$$

2. Compute the desired moments

$$\begin{cases} \hat{\mathbf{x}}_{k+1|k} \approx \sum_{i=0}^{2n} \mathbf{f}_k(\mathcal{X}_k^i) W_i \\ \mathbf{P}_{k+1|k} \approx \mathbf{Q}_k + \sum_{i=0}^{2n} (\mathbf{f}_k(\mathcal{X}_k^i) - \hat{\mathbf{x}}_{k+1|k})(\cdot)^T W_i \\ \mathbf{P}_{k,k+1|k} \approx \sum_{i=0}^{2n} \left(\mathcal{X}_k^i - \hat{\mathbf{x}}_{k|k}\right)\left(\mathbf{f}_k(\mathcal{X}_k^i) - \hat{\mathbf{x}}_{k+1|k}\right)^T W_i \end{cases}$$

3. Use (1) to do backward smoothing.

To perform cubature RTS smoothing:

1. Form a set of $2n$ σ-points

$$\begin{aligned}
\mathcal{X}_k^{(i)} &= \hat{\mathbf{x}}_{k|k} + \sqrt{n}\left(\mathbf{P}_{k|k}^{1/2}\right)_i, && i = 1, 2, \ldots, n,\\
\mathcal{X}_k^{(i+n)} &= \hat{\mathbf{x}}_{k|k} - \sqrt{n}\left(\mathbf{P}_{k|k}^{1/2}\right)_i, && i = 1, 2, \ldots, n,\\
W_i &= 1/(2n), && i = 1, 2, \ldots, 2n.
\end{aligned}$$

2. Compute the desired moments

$$\begin{cases}
\hat{\mathbf{x}}_{k+1|k} \approx \sum_{i=1}^{2n} \mathbf{f}_k(\mathcal{X}_k^i) W_i\\
\mathbf{P}_{k+1|k} \approx \mathbf{Q}_k + \sum_{i=1}^{2n} (\mathbf{f}_k(\mathcal{X}_k^i) - \hat{\mathbf{x}}_{k+1|k})(\cdot)^T W_i\\
\mathbf{P}_{k,k+1|k} \approx \sum_{i=1}^{2n} \left(\mathcal{X}_k^i - \hat{\mathbf{x}}_{k|k}\right) \left(\mathbf{f}_k(\mathcal{X}_k^i) - \hat{\mathbf{x}}_{k+1|k}\right)^T W_i
\end{cases}$$

3. Use (1) to do backward smoothing.