Nonlinear filtering

Sensor fusion & nonlinear filtering

Lars Hammarstrand

GENERAL FILTERING EQUATIONS

State space models

- We mainly consider models of the type

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

General/conceptual filtering solution

- Solve the filtering problem using:

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

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

LINEAR AND GAUSSIAN MODELS

- An important special case is linear and Gaussian models:

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

- The Kalman filter then provides an analytical solution:

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

$$\text{Update:} \begin{cases} \hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{y}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k|k-1}) \\ \mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}\mathbf{S}_k\mathbf{K}_k^T \\ \vdots \end{cases}$$

- Many important motion and measurement models are nonlinear.

Nonlinear models

- The coordinated turn motion model:

$$\begin{bmatrix} x_k \\ y_k \\ v_k \\ \phi_k \\ \omega_k \end{bmatrix} = \begin{bmatrix} x_{k-1} + \frac{2v_{k-1}}{\omega_{k-1}} \sin\left(\frac{\omega_{k-1}T}{2}\right) \cos\left(\phi_{k-1} + \frac{\omega_{k-1}T}{2}\right) \\ y_{k-1} + \frac{2v_{k-1}}{\omega_{k-1}} \sin\left(\frac{\omega_{k-1}T}{2}\right) \sin\left(\phi_{k-1} + \frac{\omega_{k-1}T}{2}\right) \\ v_{k-1} \\ \phi_{k-1} + T\omega_{k-1} \\ \omega_{k-1} \end{bmatrix} + \mathbf{q}_{k-1},$$

$f_{k-1}(x_{k-1})$

- A sensor that measures distance:

$$y_k = \sqrt{x_k^2 + y_k^2} + r_k.$$

$h_k(x_k)$

NONLINEAR FILTERING METHODS

- For nonlinear models, we have generally no analytical solution to the filtering equations.
- Common approach: Find tractable approximative solutions.
- In this course we will look at two types of nonlinear filters:
  - Gaussian filters (EKF, UKF, CKF, ...)
  - Particle filters

SELF ASSESSMENT

Why is nonlinear filtering hard? (Check all that are correct)

- The optimal filtering equations do not apply in the nonlinear setting
- We can not in general find an analytical solution to our filtering equations.
- For linear systems, it is enough to describe the posterior using the first two moments, the mean and the covariance. For nonlinear systems, we in general need infinitely many moments to describe the true posterior.
- The Markov property does not hold in the non-linear setting.

The extended Kalman filter and the Iterative extended Kalman filter

Sensor fusion & nonlinear filtering

Lars Hammarstrand

THE EXTENDED KALMAN FILTER (EKF)

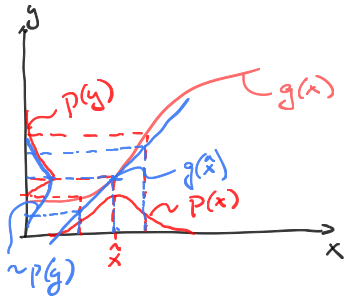
- The extended Kalman filter was developed almost immediately after the Kalman filter, in the early 1960s.

The extended Kalman filter (EKF)

- **Idea:** Linearize $\mathbf{f}_{k-1}(\mathbf{x}_{k-1})$ and $\mathbf{h}_k(\mathbf{x}_k)$ and apply the Kalman filter on the linearized system!
- Prediction: linearize $\mathbf{f}_{k-1}(\mathbf{x}_{k-1})$ around $\hat{\mathbf{x}}_{k-1|k-1}$ and use the Kalman filter prediction.
- Update: linearize $\mathbf{h}_k(\mathbf{x}_k)$ around $\hat{\mathbf{x}}_{k|k-1}$ and use the Kalman filter update.

TAYLOR SERIES EXPANSIONS

- Consider a Gaussian random variable $\mathbf{x} \sim \mathcal{N}(\hat{\mathbf{x}}, \mathbf{P})$ and a nonlinear function $\mathbf{y} = \mathbf{g}(\mathbf{x})$.



Taylor series expansions

- A first order Taylor expansion of $\mathbf{g}(\mathbf{x})$ around $\hat{\mathbf{x}}$ gives

$$\mathbf{y} \approx \mathbf{g}(\hat{\mathbf{x}}) + \mathbf{g}'(\hat{\mathbf{x}})\,(\mathbf{x} - \hat{\mathbf{x}})$$

where $[\mathbf{g}'(\mathbf{x})]_{ij} = \frac{\partial g_i(\mathbf{x})}{\partial x_j}$ is the Jacobian matrix of $\mathbf{g}$.

- Note that we now get

$$\mathbb{E}\{\mathbf{y}\} \approx \mathbf{g}(\hat{\mathbf{x}})$$
$$\mathrm{Cov}\{\mathbf{y}\} \approx \mathbf{g}'(\hat{\mathbf{x}})\,\mathbf{P}\,\mathbf{g}'(\hat{\mathbf{x}})^T.$$

$$E\{y\} = E\{g(\hat{x}) + g'(\hat{x})(x-\hat{x})\} = g(\hat{x})$$

$$Cov\{y\} = Cov\{g(\hat{x}) + g'(\hat{x})(x-\hat{x})\} = g'(\hat{x})\,Cov\{x-\hat{x}\}\,g'(\hat{x})^T$$

SELF ASSESSMENT

- What is $g(\hat{\mathbf{x}})$ and $g'(\hat{\mathbf{x}})$ when

$$\mathbf{x} = \begin{bmatrix} p_1 \\ p_2 \\ v_1 \\ v_2 \end{bmatrix}, \hat{\mathbf{x}} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} \text{ and } y = \sqrt{p_1^2 + p_2^2}?$$

- $g(\hat{\mathbf{x}}) = \sqrt{5}$ and $g'(\hat{\mathbf{x}}) = \begin{bmatrix} \frac{1}{\sqrt{5}} & \frac{2}{\sqrt{5}} & 0 & 0 \end{bmatrix}^T$
- $g(\hat{\mathbf{x}}) = \sqrt{5}$ and $g'(\hat{\mathbf{x}}) = \begin{bmatrix} \frac{1}{\sqrt{5}} & \frac{2}{\sqrt{5}} & 0 & 0 \end{bmatrix}$
- $g(\hat{\mathbf{x}}) = \sqrt{5}$ and $g'(\hat{\mathbf{x}}) = \begin{bmatrix} \frac{1}{\sqrt{5}} & \frac{2}{\sqrt{5}} \end{bmatrix}$
- $g(\hat{\mathbf{x}}) = \sqrt{5}$ and $g'(\hat{\mathbf{x}}) = \begin{bmatrix} \frac{1}{\sqrt{5}} & \frac{2}{\sqrt{5}} \end{bmatrix}^T$

EKF: THE PREDICTION STEP

- We have

$$\mathbf{x}_{k-1} \,|\, \mathbf{y}_{1:k-1} \sim \mathcal{N}(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{P}_{k-1|k-1}).$$

- By using the approximation

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

we get:

EKF: the prediction step

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

$\tilde{\mathbf{F}}$ $\tilde{\mathbf{F}}^T$

EKF: THE UPDATE STEP

- Similarly, if we approximate

$$\begin{aligned}\mathbf{y}_k &= \mathbf{h}(\mathbf{x}_k) + \mathbf{r}_k \\ &\approx \mathbf{h}(\hat{\mathbf{x}}_{k|k-1}) + \mathbf{h}'(\hat{\mathbf{x}}_{k|k-1})\left(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}\right) + \mathbf{r}_{k-1},\end{aligned}$$

we get:

EKF: the update step

$$\begin{aligned}\hat{\mathbf{x}}_{k|k} &= \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\left(\mathbf{y}_k - \mathbf{h}(\hat{\mathbf{x}}_{k|k-1})\right) \\ \mathbf{P}_{k|k} &= \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{S}_k\mathbf{K}^T \\ \mathbf{S}_k &= \mathbf{h}'(\hat{\mathbf{x}}_{k|k-1})\mathbf{P}_{k|k-1}\mathbf{h}'(\hat{\mathbf{x}}_{k|k-1})^T + \mathbf{R}_k \\ \mathbf{K}_k &= \mathbf{P}_{k|k-1}\mathbf{h}'(\hat{\mathbf{x}}_{k|k-1})^T\mathbf{S}_k^{-1}\end{aligned}$$

The extended Kalman filter – Examples and remarks

Sensor fusion & nonlinear filtering

Lars Hammarstrand

EKF: THE UPDATE STEP

$p(x|y) \propto p(x,y) = p(y|x)p(x)$

EKF update step – Example 1

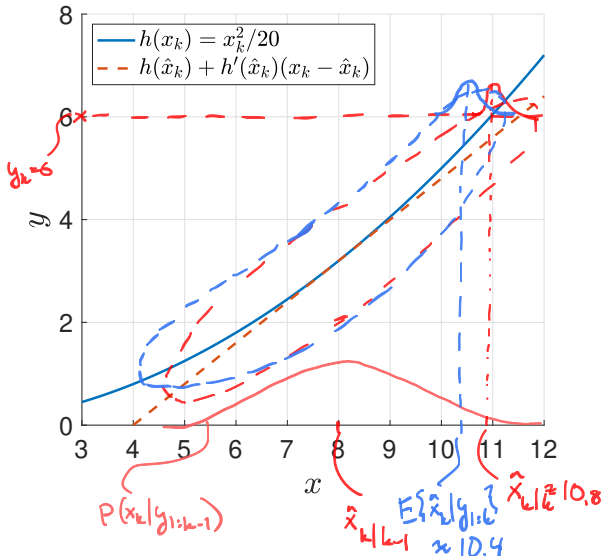
- Prediction:

$$x_k|y_{1:k-1} \sim \mathcal{N}(8, 3.5^2)$$

- Measuremens: $y_k = 6$

$$y_k = h(x_k) + r_k$$
$$h(x_k) = x_k^2/20$$
$$r_k \sim \mathcal{N}(0, 2)$$

- Find the posterior mean $\mathbb{E}\{x_k|y_{1:k}\}$ and the approximate posterior mean $\hat{x}_{k|k}$ given by EKF.



EKF: THE UPDATE STEP

EKF update step – Example 2

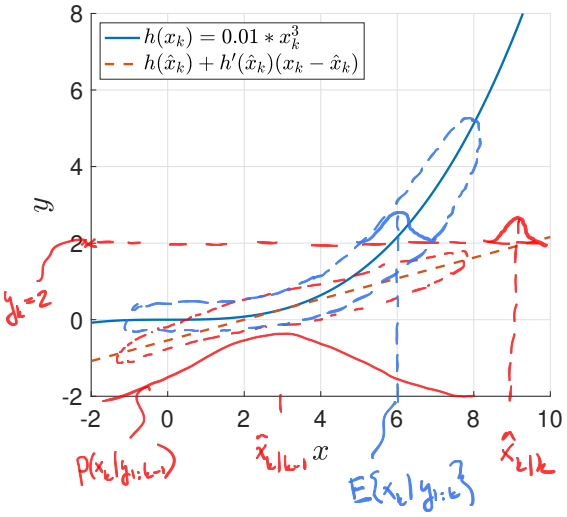
- Prediction:

$$x_k | y_{1:k-1} \sim \mathcal{N}(3, 3.5^2)$$

- Measurement: $y_k = 2$

$$y_k = h(x_k) + r_k$$
$$h(x_k) = 0.01 x_k^3$$
$$r_k \sim \mathcal{N}(0, 0.3)$$

- Find the posterior mean $\mathbb{E}\{x_k | y_{1:k}\}$ and the approximate posterior mean $\hat{x}_{k|k}$ given by EKF.



EKF: REMARKS

- The EKF recursively computes Gaussian approximations to $p(\mathbf{x}_k|\mathbf{y}_{1:k-1})$ and $p(\mathbf{x}_k|\mathbf{y}_{1:k})$.
- The EKF is also approximately LMMSE.
- As long as the systems are not too nonlinear, the EKF tends to perform well.
- To implement an EKF you need to compute derivatives of the nonlinear functions.
- Compared to other nonlinear filters that we study, it has the lowest computational complexity.

SELF ASSESSMENT

Which of the following statements are true? Check all that apply!

- The ordinary Kalman filter can be used also for nonlinear models but it will perform worse than, e.g., the extended Kalman filter.
- If we have a mix of linear and nonlinear models, let's say that, e.g., the motion model is linear while the measurement model is nonlinear. In this case, it is appropriate to combine the Kalman filter prediction step with the EKF update step to approximate the posterior.
- For linear and Gaussian state space models, the EKF is equivalent to the Kalman filter.
- We can use the EKF for all nonlinear state space models, but it will only be accurate for "mildly" nonlinear models.

The Iterative extended Kalman filter

Sensor fusion & nonlinear filtering

Lars Hammarstrand

ITERATED EXTENDED KALMAN FILTERS

The Iterated EKF (IEKF)

- Once we have computed $\hat{\mathbf{x}}_{k|k}$ we have a better guess about $\mathbf{x}_k$

  $\rightsquigarrow$ we can linearize $\mathbf{h}(\mathbf{x}_k)$ around that point and update $p(\mathbf{x}_k|\mathbf{y}_{1:k-1})$ using the new linearization and $\mathbf{y}_k$.

- **Note:**
  - we obtain an iterative algorithm that we can run until convergence,
  - each update starts with the predicted density $\mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1})$ to avoid using $\mathbf{y}_k$ multiple times.

IEKF: REMARKS

IEKF – Gauss Newton search

- One can show that the IEKF solves

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k}^{\mathbf{MAP}} = \arg\max_{\mathbf{x}_k} p(\mathbf{x}_k | \mathbf{y}_{1:k})$$

  iteratively using a Gauss-Newton search.

EKF: THE UPDATE STEP

EKF update step – Example 2

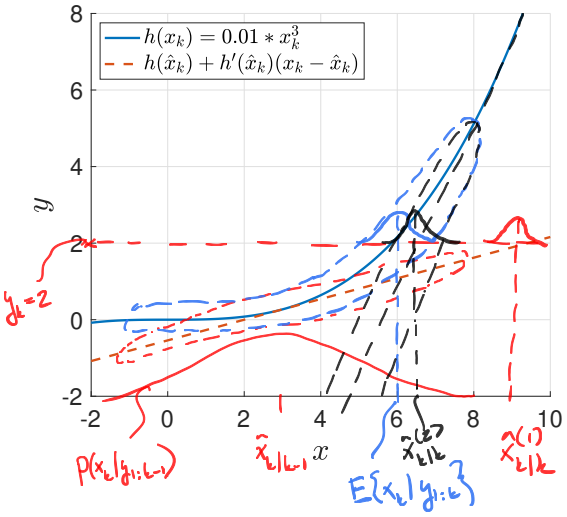
- Prediction:

$$x_k | y_{1:k-1} \sim \mathcal{N}(3, 3.5^2)$$

- Measurement: $y_k = 2$

$$y_k = h(x_k) + r_k$$
$$h(x_k) = 0.01 x_k^3$$
$$r_k \sim \mathcal{N}(0, 0.3)$$

- Find the posterior mean $\mathbb{E}\{x_k | y_{1:k}\}$ and the approximate posterior mean $\hat{x}_{k|k}$ given by EKF.



IEKF: REMARKS

IEKF – Gauss Newton search

- One can show that the IEKF solves

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k}^{\mathbf{MAP}} = \arg\max_{\mathbf{x}_k} p(\mathbf{x}_k | \mathbf{y}_{1:k})$$

  iteratively using a Gauss-Newton search.

Pros:

- the IEKF usually performs very well when the measurement noise is small (the MAP estimate is accurate then).
- the IEKF often converges in very few iterations.

Cons:

- posterior pdfs are often rather skewed
  $\Rightarrow$ the MAP estimate is far from the posterior mean.
- the IEKF may diverge (more robust alternatives exist).

Assumed density filters

Sensor fusion & nonlinear filtering

Lars Hammarstrand

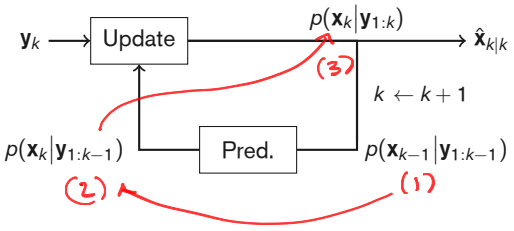
GENERAL FILTERING STRATEGIES

- Performing nonlinear filtering well is difficult!

Methodology

- Recursively compute $p(\mathbf{x}_k|\mathbf{y}_{1:k})$.
- Find an estimate $\hat{\mathbf{x}}_{k|k}$ based on $p(\mathbf{x}_k|\mathbf{y}_{1:k})$.

- Most methods follow the below strategy, but what else do they have in common?



GAUSSIAN FILTERING

Gaussian Filtering

1. Start each recursion with
   $p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1}) \simeq \mathcal{N}(\mathbf{x}_{k-1}; \hat{\mathbf{x}}_{k-1|k-1}, \mathbf{P}_{k-1|k-1})$.
2. Prediction step: Find
   $p(\mathbf{x}_k|\mathbf{y}_{1:k-1}) \simeq \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1})$
3. Update step: Find
   $p(\mathbf{x}_k|\mathbf{y}_{1:k}) \simeq \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k})$.

Key property:

- Both $p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})$ and $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ are Gaussian $\Rightarrow$ we have a recursive algorithm!

Note:

- KF, EKF, and IEKF are all examples of Gaussian filters.
- Although $p(\mathbf{x}_k|\mathbf{y}_{1:k-1})$ is also approximated as Gaussian, this is not a requirement.

ASSUMED DENSITY FILTERS

General solution

- A wide range of Bayesian filters can be written on the following form:
  1. Select a density parameterization $p(\mathbf{x}; \boldsymbol{\theta})$.
  2. Start from

$$p(\mathbf{x}_{k-1} | \mathbf{y}_{1:k-1}) \approx p(\mathbf{x}_{k-1}; \boldsymbol{\theta}_{k-1|k-1})$$

  and find $\boldsymbol{\theta}_{k|k}$ such that

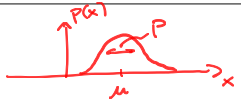
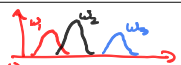
$$p(\mathbf{x}_k | \mathbf{y}_{1:k}) \approx p(\mathbf{x}_k; \boldsymbol{\theta}_{k|k})$$

Note:

- The strategy is sometimes called assumed density filtering or canonical form filtering.

ASSUMED DENSITY FILTERS

Some important examples:

Assumed density	Filters
$p(\mathbf{x};\boldsymbol{\theta}) = \mathcal{N}(\mathbf{x};\boldsymbol{\mu},\mathbf{P})$ 	KF, EKF, UKF, CKF, NN, GNN, PDA, JPDA.
$p(\mathbf{x};\boldsymbol{\theta}) = \sum_{i=1}^{N} w_i \mathcal{N}(\mathbf{x};\boldsymbol{\mu}_i,\mathbf{P}_i)$ 	IMM, MHT.
$p(\mathbf{x};\boldsymbol{\theta}) = \sum_{i=1}^{N} w^{(i)} \delta(\mathbf{x}-\mathbf{x}^{(i)})$ 	Particle filters.
$p(\mathbf{x}_l,\mathbf{x}_n;\boldsymbol{\theta}) = \sum_{i=1}^{N} w^{(i)} \delta(\mathbf{x}_n-\mathbf{x}_n^{(i)}) \mathcal{N}(\mathbf{x}_l;\boldsymbol{\mu}^{(i)},\mathbf{P}^{(i)})$	Rao-Blackwellized particle filters.

- Algorithms in blue are covered in this course, whereas algorithms in red are related to data association problems.

SELF ASSESSMENT

Which of the following statements are true? Check all that apply!

- Our choice of density parametrization will both effect the performance of our filter and its computational complexity.
- In real-time applications, having a recursive filter means that our compute time for producing a new estimate is more or less constant with each new measurement.
- In nonlinear filtering, considering one measurement at a time (typical recursive filter) will produce better or equally good approximations of the true posterior as if we were allowed to consider the complete measurement history (non-recursive filter).
- A prerequisite for a recursive filter is that the prior and posterior density in each recursion can be described using the same density parametrization.

Integrals involved in Gaussian filtering

Sensor fusion & nonlinear filtering

Lars Hammarstrand

INTEGRALS IN GAUSSIAN FILTERING – PREDCTION

- The prediction step in Gaussian filtering involves two integrals on the form, $\int \mathbf{g}(\mathbf{x})\mathcal{N}(\mathbf{x}, \hat{\mathbf{x}}, \mathbf{P})\, d\mathbf{x}$:

$$E_{\mathcal{N}(x;\hat{x},P)}\{g(x)\}$$

Integrals in Gaussian filter prediction

$$\hat{\mathbf{x}}_{k|k-1} = \int f(\mathbf{x}_{k-1})\mathcal{N}(\mathbf{x}_{k-1}; \hat{\mathbf{x}}_{k-1|k-1}, \mathbf{P}_{k-1|k-1})\, d\mathbf{x}_{k-1}$$

$$\mathbf{P}_{k|k-1} = \mathbf{Q}_{k-1} + \int (f(\mathbf{x}_{k-1}) - \hat{\mathbf{x}}_{k|k-1})(\cdot)^T \mathcal{N}(\mathbf{x}_{k-1}; \hat{\mathbf{x}}_{k-1|k-1}, \mathbf{P}_{k-1|k-1})\, d\mathbf{x}_{k-1}$$

- From these we can approximate

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

INTEGRALS IN GAUSSIAN FILTERING

- The update step in Gaussian filtering involves three integrals on the form, $\int \mathbf{g}(\mathbf{x})\mathcal{N}(\mathbf{x}, \hat{\mathbf{x}}, \mathbf{P})\, d\mathbf{x}$:

Integrals in Gaussian filter update

$$\hat{\mathbf{y}}_{k|k-1} = \int h(\mathbf{x}_k)\mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1})\, d\mathbf{x}_k$$

$$\mathbf{P}_{xy} = \int (\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})(h(\mathbf{x}_k) - \hat{\mathbf{y}}_{k|k-1})^T \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1}) d\mathbf{x}_k$$

$$\mathbf{S}_k = \mathbf{R}_k + \int (h(\mathbf{x}_k) - \hat{\mathbf{y}}_{k|k-1})(\cdot)^T \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1})\, d\mathbf{x}_k$$

- From these we compute:

$$\begin{cases} \hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{P}_{xy}\mathbf{S}_k^{-1}(\mathbf{y}_k - \hat{\mathbf{y}}_{k|k-1}) \\ \mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{P}_{xy}\mathbf{S}_k^{-1}\mathbf{P}_{xy}^T. \end{cases}$$

INTEGRALS IN GAUSSIAN FILTERING – EXAMPLE

Polar measurements

- Suppose we observe a measurement

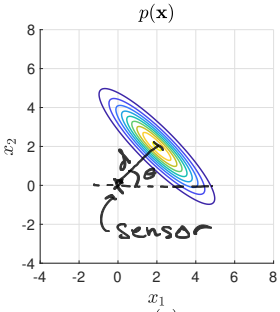
$$\mathbf{y} = \mathbf{h}(\mathbf{x}) + \mathbf{r} = \begin{bmatrix} \sqrt{x_1^2 + x_2^2} \\ \arctan\left(\frac{x_2}{x_1}\right) \end{bmatrix} + \mathbf{r}, \quad \mathbf{r} = \mathbf{0} :$$

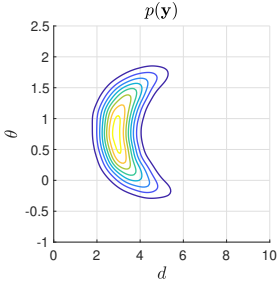
Where our prior is

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}, \mathbf{P}) = \mathcal{N}\left(\mathbf{x}; \begin{bmatrix} 2 \\ 2 \end{bmatrix}, \begin{bmatrix} 2 & -1.8 \\ -1.8 & 2 \end{bmatrix}\right)$$

- To predict the measurement we compute

$$\mathbb{E}\{\mathbf{y}\} = \int \mathbf{h}(\mathbf{x}) \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}, \mathbf{P})\, d\mathbf{x}$$





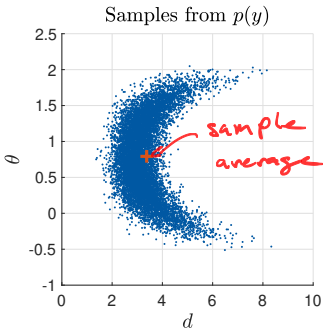
MONTE CARLO SAMPLING

The Monte Carlo method

- Generate independent and identically distributed (i.i.d.) samples, $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(N)}$, from $p(\mathbf{x})$. Approximate

$$\int g(\mathbf{x}) p(\mathbf{x})\, d\mathbf{x} \approx \frac{1}{N} \sum_{i=1}^{N} g(\mathbf{x}^{(i)}).$$

- The Monte Carlo is
  - simple to use to perform Gaussian filtering.
  - asymptotically exact!
  - seldom used (need many samples).



Cov: $g(x) = (h(x) - \hat{y})(\cdot)^T$

STOCHASTIC DECOUPLING

- Suppose $\mathbf{P}^{1/2}$ is a matrix such that $\mathbf{P}^{1/2}(\mathbf{P}^{1/2})^T = \mathbf{P}$.
- We often use the Cholesky decomposition to find $\mathbf{P}^{1/2}$.
- If $\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ then

$$\mathbf{x} = \hat{\mathbf{x}} + \mathbf{P}^{1/2}\boldsymbol{\xi} \sim \mathcal{N}(\hat{\mathbf{x}}, \mathbf{P})$$

- By changing the variable of integration from $\mathbf{x}$ to $\boldsymbol{\xi}$ we get

$$\int \mathbf{g}(\mathbf{x})\mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}, \mathbf{P})\, d\mathbf{x} = \int \mathbf{g}(\hat{\mathbf{x}} + \mathbf{P}^{1/2}\boldsymbol{\xi})\mathcal{N}(\boldsymbol{\xi}; \mathbf{0}, \mathbf{I})\, d\boldsymbol{\xi}.$$

Conclusion

- To perform Gaussian filtering, it is sufficient to be able to compute

$$\int \mathbf{g}(\boldsymbol{\xi})\mathcal{N}(\boldsymbol{\xi}; \mathbf{0}, \mathbf{I})\, d\boldsymbol{\xi}$$

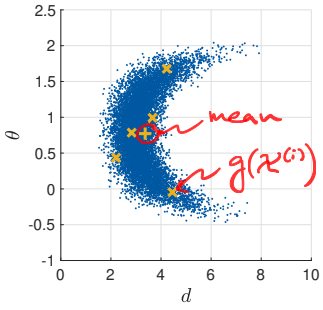
SIGMA-POINT METHODS

General idea

- Approximate

$$\int g(\mathbf{x})\mathcal{N}(\mathbf{x};\hat{\mathbf{x}},\mathbf{P})\,d\mathbf{x} \approx \sum_{i=1}^{N} W^{(i)} g(\mathcal{X}^{(i)})$$

where $\mathcal{X}^{(i)}$ are called σ-points and $W^{(i)}$ are weights.



- Remarks:
  - Compared to MC approximation, points selected deterministically.
  - Many σ-point methods: unscented transform, cubature rule, Gauss-Hermite quadrature, Gaussian process quadrature, marginalized transform, etc.
  - Each used in Gaussian filtering and the filters are known as UKF, CKF, GHKF, GPKF, MKF, etc.

A LOOK AHEAD

We will cover

- The unscented transform (UT)
  - The most commonly used σ-point method. Has several tuning parameters.
- The cubature rule
  - Uses one point less than UT. No tuning parameters. A simple method that often performs similar to UT.

and explain how these can be used for Gaussian filtering!

SELF ASSESSMENT

Check all statements that apply:

- The Monte Carlo method approximates expected values by sample averages.
- A σ-point method makes use of a small number of weighted random samples.
- Stochastic decoupling is about rewriting an integral with respect to a vector of correlated random variables as an integral with respect to a vector of independent random variables that have zero mean and unit variance.
- The Gauss-Hermite Kalman filter, the Unscented Kalman filter and the Cubature Kalman filter are all Gaussian filters and they only differ in how they approximate the involved integrals.

Gaussian filters and moment matching

Sensor fusion & nonlinear filtering

Lars Hammarstrand

GAUSSIAN FILTERING

- We mainly consider models of the type

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

Gaussian filtering solution

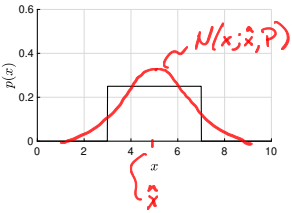
- Approximate distributions as Gaussian in the prediction and update steps:

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

Update:
$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) = \frac{p(\mathbf{y}_k|\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{y}_{1:k-1})}{\int p(\mathbf{y}_k|\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{y}_{1:k-1})\, d\mathbf{x}_k}$$
$$\approx \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k})$$

GAUSSIAN APPROXIMATIONS BY MOMENT MATCHING

- Suppose that we are given a non-Gaussian density $p(\mathbf{x})$.
- Task: find $\hat{\mathbf{x}}$ and $\mathbf{P}$ such that $p(\mathbf{x}) \approx \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}, \mathbf{P})$.



Moment matching

- One strategy is to select $\hat{\mathbf{x}}$ and $\mathbf{P}$ to match the moments of $p(\mathbf{x})$:

$$\hat{\mathbf{x}} = \mathbb{E}_{p(\mathbf{x})}\{\mathbf{x}\} = \int \mathbf{x} p(\mathbf{x})\, d\mathbf{x}$$

$$\mathbf{P} = \mathrm{Cov}_{p(\mathbf{x})}\{\mathbf{x}\} = \int (\mathbf{x} - \hat{\mathbf{x}})(\mathbf{x} - \hat{\mathbf{x}})^T p(\mathbf{x})\, d\mathbf{x}$$

- One can show that moment matching minimizes the Kullback-Leibler divergence

$$\mathrm{KL}\left(p(\mathbf{x}) \middle| \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}, \mathbf{P})\right) = \int p(\mathbf{x}) \log \frac{p(\mathbf{x})}{\mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}, \mathbf{P})}\, d\mathbf{x}.$$

GAUSSIAN PREDICTION BY MOMENT MATCHING

- Given

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

$\Rightarrow p(x_k|y_{1:k-1}) \approx \mathcal{N}(x_k; \hat{x}_{k|k-1}, P_{k|k-1})$

the first two predicted moments of $\mathbf{x}_k$ are

Prediction by moment matching

$$\begin{aligned} \hat{\mathbf{x}}_{k|k-1} &= \mathbb{E}\{\mathbf{x}_k \,|\, \mathbf{y}_{1:k-1}\} \\ &= \int f(\mathbf{x}_{k-1}) \mathcal{N}(\mathbf{x}_{k-1}; \hat{\mathbf{x}}_{k-1|k-1}, \mathbf{P}_{k-1|k-1})\, d\mathbf{x}_{k-1} \\ \mathbf{P}_{k|k-1} &= \mathrm{Cov}\{\mathbf{x}_k \,|\, \mathbf{y}_{1:k-1}\} = \mathbf{Q}_{k-1} + \\ &\int (f(\mathbf{x}_{k-1}) - \hat{\mathbf{x}}_{k|k-1})(\cdot)^T \mathcal{N}(\mathbf{x}_{k-1}; \hat{\mathbf{x}}_{k-1|k-1}, \mathbf{P}_{k-1|k-1})\, d\mathbf{x}_{k-1} \end{aligned}$$

GAUSSIAN UPDATE BY MOMENT MATCHING

- We are given

$$\begin{cases} \mathbf{x}_k | \mathbf{y}_{1:k-1} \sim \mathcal{N}(\hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1}) \\ \mathbf{y}_k = h(\mathbf{x}_k) + \mathbf{r}_k. \end{cases}$$

- Ideal solution: set $\hat{\mathbf{x}}_{k|k}$ and $\mathbf{P}_{k|k}$ to the first two moments of

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

- Unfortunately, it is difficult to efficiently compute the moments of $p(\mathbf{x}_k | \mathbf{y}_{1:k})$.

Alternative moment matching strategy

- Approximate $p(\mathbf{x}_k, \mathbf{y}_k | \mathbf{y}_{1:k-1})$ as Gaussian using moment matching.
  $\rightsquigarrow$ When $\mathbf{x}_k, \mathbf{y}_k | \mathbf{y}_{1:k-1}$ is Gaussian, we can find $p(\mathbf{x}_k | \mathbf{y}_k, \mathbf{y}_{1:k-1})$ analytically.

GAUSSIAN UPDATE BY MOMENT MATCHING

- We approximate $(\mathbf{x}_k, \mathbf{y}_k)$ as jointly Gaussian using moment matching:

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

where

$$\hat{\mathbf{y}}_{k|k-1} = \mathbb{E}\{\mathbf{y}_k | \mathbf{y}_{1:k-1}\} = \mathbb{E}\{h(\mathbf{x}_k) | \mathbf{y}_{1:k-1}\} = \int h(\mathbf{x}_k) \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1})\, d\mathbf{x}_k$$

$$\begin{aligned} \mathbf{P}_{xy} &= \mathbb{E}\{(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})(\mathbf{y}_k - \hat{\mathbf{y}}_{k|k-1})^T | \mathbf{y}_{1:k-1}\} \\ &= \int (\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})(h(\mathbf{x}_k) - \hat{\mathbf{y}}_{k|k-1})^T \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1})\, d\mathbf{x}_k \end{aligned}$$

$$\begin{aligned} \mathbf{S}_k &= \mathrm{Cov}\{\mathbf{y}_k | \mathbf{y}_{1:k-1}\} = \mathbf{R}_k + \mathrm{Cov}\{h(\mathbf{x}_k) | \mathbf{y}_{1:k-1}\} \\ &= \mathbf{R}_k + \int (h(\mathbf{x}_k) - \hat{\mathbf{y}}_{k|k-1})(\cdot)^T \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1})\, d\mathbf{x}_k \end{aligned}$$

GAUSSIAN UPDATE BY MOMENT MATCHING

- If we know that

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

then it holds that

$$\begin{cases} \hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{P}_{xy}\mathbf{S}_k^{-1}(\mathbf{y}_k - \hat{\mathbf{y}}_{k|k-1}) \\ \mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{P}_{xy}\mathbf{S}_k^{-1}\mathbf{P}_{xy}^T. \end{cases}$$

- Key difference compared to the Kalman filter: we need to *approximate* $\hat{\mathbf{y}}_{k|k-1}$, $\mathbf{S}_k$ and $\mathbf{P}_{xy}$.
- Note: we find these components by solving integrals of the type $\int g(\mathbf{x})\mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}, \mathbf{P})\, d\mathbf{x}$.

SELF ASSESSMENT

In Gaussian filtering using moment matching we need to solve several different integrals (two during the prediction and three during the update step). Among the following integrals, which ones do we need to solve?

- To compute $\mathbf{P}_{xy}$ in the update step: $\int f(\mathbf{x}_k)h(\mathbf{x}_k)\, d\mathbf{x}_k$.
- To compute $\mathbf{P}_{k|k-1}$ in the prediction step we take $\mathbf{Q}_{k-1}$ plus this integral: $\int (f(\mathbf{x}_{k-1}) - \hat{\mathbf{x}}_{k|k-1})(\cdot)^T \mathcal{N}(\mathbf{x}_{k-1}; \hat{\mathbf{x}}_{k-1|k-1}, \mathbf{P}_{k-1|k-1})\, d\mathbf{x}_{k-1}$
- To compute $\hat{\mathbf{y}}_{k|k-1}$ in the update step: $\int h(\mathbf{x}_k)\mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1})\, d\mathbf{x}_k$
- To compute $\mathbf{P}_{k|k}$ in the update step: $\int (\mathbf{x}_k - \hat{\mathbf{x}}_{k|k})(\cdot)^T \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k})\, d\mathbf{x}_k$

Sigma-point methods

Sensor fusion & nonlinear filtering

Lars Hammarstrand

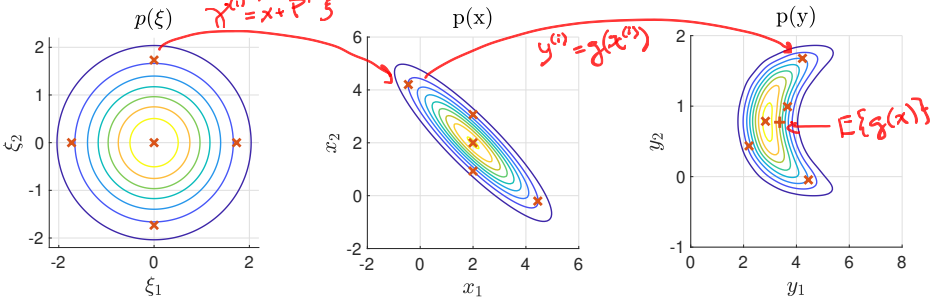
SIGMA-POINT METHODS – INTEGRAL APPROXIMATION

General idea

- Suppose $\mathbf{x} \sim \mathcal{N}(\hat{\mathbf{x}}, \mathbf{P})$, we can then approximate

$$\mathbb{E}\{\mathbf{g}(\mathbf{x})\} = \int \mathbf{g}(\hat{\mathbf{x}} + \mathbf{P}^{1/2}\boldsymbol{\xi})\mathcal{N}(\boldsymbol{\xi}; \mathbf{0}, \mathbf{I})\, d\boldsymbol{\xi} \approx \sum_{i=1}^{N} W_i \mathbf{g}(\hat{\mathbf{x}} + \mathbf{P}^{1/2}\boldsymbol{\xi}^{(i)})$$

where $\boldsymbol{\xi}^{(i)}$ are called σ-points and W_i are weights.



SIGMA-POINT METHOD IN GAUSSIAN FILTERING

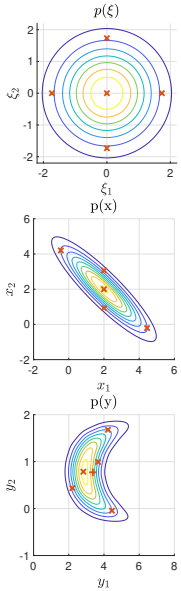
Filtering idea

- Idea 1: *it is easier to approximate a probability distribution than it is to approximate an arbitrary nonlinear function or transformation.*
- Idea 2: if $\mathbf{y} = \mathbf{g}(\mathbf{x})$ and $\mathbf{x} \sim \mathcal{N}(\hat{\mathbf{x}}, \mathbf{P})$ we can approximate $p(\mathbf{y})$ using

$$\mathbb{E}\{\mathbf{y}\} \approx \boldsymbol{\mu}_y = \sum_{i=0}^{N} W_i \, \mathbf{g}(\mathcal{X}^{(i)})$$

$$\text{Cov}\{\mathbf{y}\} \approx \sum_{i=0}^{N} W_i \left(\mathbf{g}(\mathcal{X}^{(i)}) - \boldsymbol{\mu}_y\right)\left(\mathbf{g}(\mathcal{X}^{(i)}) - \boldsymbol{\mu}_y\right)^T$$

where $\mathcal{X}^{(i)}$ are σ-points and W_i the associated weights.



THE UNSCENTED TRANSFORM (UT)

Unscented transform (UT)

- Form a set of $2n+1$ σ-points as follows:

$$\mathcal{X}^{(0)} = \hat{\mathbf{x}}$$

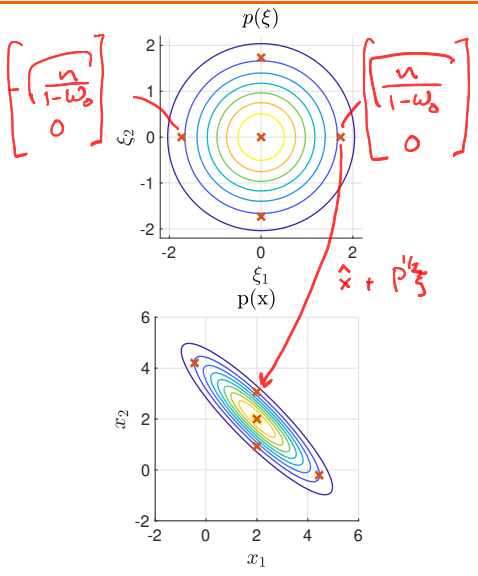
$$\mathcal{X}^{(i)} = \hat{\mathbf{x}} + \sqrt{\frac{n}{1-W_0}}\mathbf{P}_i^{1/2}, \quad i = 1, 2, \ldots, n,$$

$$\mathcal{X}^{(i+n)} = \hat{\mathbf{x}} - \sqrt{\frac{n}{1-W_0}}\mathbf{P}_i^{1/2}, \quad i = 1, 2, \ldots, n,$$

$$W_i = \frac{1-W_0}{2n}$$

where $\mathbf{P}_i^{1/2}$ is the ith column of $\mathbf{P}^{1/2}$.

- Other versions with more design parameters.
- If $\mathbf{x}$ is Gaussian, set $W_0 = 1 - n/3$.



AN ILLUSTRATION OF THE UNSCENTED TRANSFORM

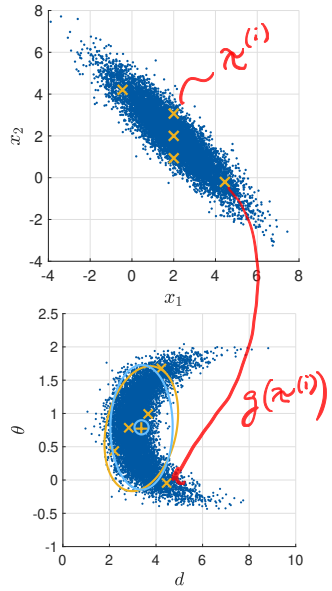
UT for polar measurements

- Consider again the example where

$$\mathbf{y} = \mathbf{g}(\mathbf{x}) = \begin{bmatrix} \sqrt{x_1^2 + x_2^2} \\ \arctan\left(\frac{x_2}{x_1}\right) \end{bmatrix}$$

and

$$\mathbf{x} \sim \mathcal{N}\left(\begin{bmatrix} 2 \\ 2 \end{bmatrix}, \begin{bmatrix} 2 & -1.8 \\ -1.8 & 2 \end{bmatrix} \right)$$



THE CUBATURE RULE

Cubature rule

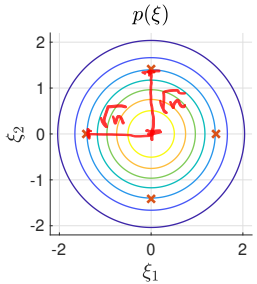
- Forms a set of $2n$ σ-points as follows:

$$\mathcal{X}^{(i)} = \hat{\mathbf{x}} + \sqrt{n}\mathbf{P}_i^{1/2}, \qquad i = 1, 2, \ldots, n,$$
$$\mathcal{X}^{(i+n)} = \hat{\mathbf{x}} - \sqrt{n}\mathbf{P}_i^{1/2}, \qquad i = 1, 2, \ldots, n,$$
$$W_i = \frac{1}{2n}$$

where $\mathbf{P}_i^{1/2}$ is the ith column of $\mathbf{P}^{1/2}$.

Note:

- Special case of UT: $W_0 = 0$!
- No tuning parameters and no negative weights.
- Popularized in 2009.



AN ILLUSTRATION OF THE CUBATURE RULE

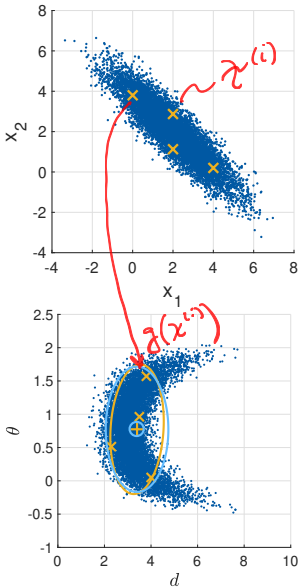
Cubature rule for polar measurements

- Consider again the example where

$$\mathbf{y} = \mathbf{g}(\mathbf{x}) = \begin{bmatrix} \sqrt{x_1^2 + x_2^2} \\ \arctan\left(\frac{x_2}{x_1}\right) \end{bmatrix}$$

and

$$\mathbf{x} \sim \mathcal{N}\left(\begin{bmatrix} 2 \\ 2 \end{bmatrix}, \begin{bmatrix} 2 & -1.8 \\ -1.8 & 2 \end{bmatrix} \right)$$



AN ILLUSTRATION OF EKF

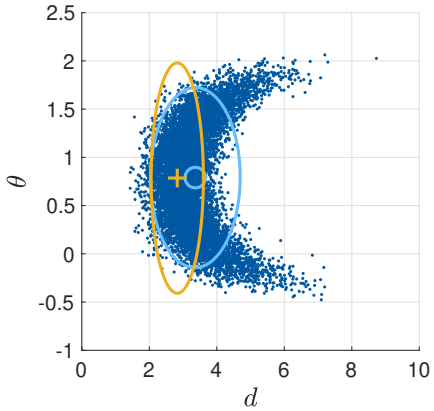
EKF for polar measurements

- An EKF would instead handle

$$\mathbf{g}(\mathbf{x}) = \begin{bmatrix} \sqrt{x_1^2 + x_2^2} \\ \arctan\left(\frac{x_2}{x_1}\right) \end{bmatrix}$$

by linearization:

$$\mathbf{y} \approx \mathbf{g}\left(\begin{bmatrix}2\\2\end{bmatrix}\right) + \mathbf{g}'\left(\begin{bmatrix}2\\2\end{bmatrix}\right)\left(\mathbf{x} - \begin{bmatrix}2\\2\end{bmatrix}\right)$$



REMARKS ON THE UT AND THE CUBATURE RULE

Unscented transform:

- $W_0 = 1 - n/3$ is negative for $n > 3$ $\Rightarrow$ Cov$\{\mathbf{y}\}$ can in become negative definite.
- UT variant with more design parameters but less intuitive.

Cubature rule:

- No theoretical risk of covariance becoming negative definite.
- No design parameters
- The spread of the σ-points increase with n, since $\mathcal{X}^{(i)} = \hat{\mathbf{x}} \pm \sqrt{n}\mathbf{P}_i^{1/2}$.

- Both the Unscented transform and the cubature rule computes the mean exactly for polynomials up to order 3, but the covariance is only exact when $\mathbf{g}(\mathbf{x})$ is of order 1.

SELF ASSESSMENT

Check all that apply!

- The number of σ-points used in an UT grows linearly with the dimension of $\mathbf{x}$.
- The number of σ-points in UT is determined by the dimensionality of $\mathbf{y} = \mathbf{g}(\mathbf{x})$.
- We need to evaluate $\mathbf{g}(\mathbf{x})$ at $4n + 2$ different points since need $2n + 1$ points to approximate the mean and later $2n + 1$ points to approximate the covariance.

SELF ASSESSMENT

Suppose that $n = 1$ and $p(x) = \mathcal{N}(x; 0, 1)$, the cubature rule is then simply

$$\mathbb{E}\{\mathbf{g}(x)\} \approx \mathbf{g}(-1)\frac{1}{2} + \mathbf{g}(1)\frac{1}{2}.$$

Compute the cubature rule approximation to $\mathbb{E}\{\mathbf{g}(x)\}$ when we further assume that

$$\mathbf{g}(x) = \begin{bmatrix} 1 + x + 5x^4 \\ 3x^2 + 4 \\ 2x^3 + x^5 \end{bmatrix}.$$

(You can also ask yourself which elements of $\mathbb{E}\{\mathbf{g}(x)\}$ that we have thus computed exactly.)

- $\begin{bmatrix} 3 & 3.5 & 1.5 \end{bmatrix}^T$
- $\begin{bmatrix} 6 & 7 & 3 \end{bmatrix}^T$
- $\begin{bmatrix} 7 & 7 & 3 \end{bmatrix}^T$
- $\begin{bmatrix} 6 & 7 & 0 \end{bmatrix}^T$

The prediction and update step in the Unscented KF and the Cubature KF

Sensor fusion & nonlinear filtering

Lars Hammarstrand

GAUSSIAN PREDICTION BY MOMENT MATCHING

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

Gaussian filter prediction

$$\hat{\mathbf{x}}_{k|k-1} = \mathbb{E}\{\mathbf{x}_k|\mathbf{y}_{1:k-1}\} = \int \mathbf{f}(\mathbf{x}_{k-1})\mathcal{N}(\mathbf{x}_{k-1}; \hat{\mathbf{x}}_{k-1|k-1}, \mathbf{P}_{k-1|k-1})\, d\mathbf{x}_{k-1}$$

$$\mathbf{P}_{k|k-1} = \mathrm{Cov}\{\mathbf{x}_k|\mathbf{y}_{1:k-1}\} = \mathbf{Q}_{k-1} + \int (\mathbf{f}(\mathbf{x}_{k-1}) - \hat{\mathbf{x}}_{k|k-1})(\cdot)^T \mathcal{N}(\mathbf{x}_{k-1}; \hat{\mathbf{x}}_{k-1|k-1}, \mathbf{P}_{k-1|k-1})\, d\mathbf{x}_{k-1}$$

- **Objective:** show how the unscented transform and the cubature rule can approximate $\hat{\mathbf{x}}_{k|k-1}$ and $\mathbf{P}_{k|k-1}$.

PREDICTION IN UKF

1. Form a set of $2n+1$ σ-points

$$\mathcal{X}_{k-1}^{(0)} = \hat{\mathbf{x}}_{k-1|k-1},$$

$$\mathcal{X}_{k-1}^{(i)} = \hat{\mathbf{x}}_{k-1|k-1} + \sqrt{\frac{n}{1-W_0}}\left(\mathbf{P}_{k-1|k-1}^{1/2}\right)_i, \quad i = 1, 2, \ldots, n,$$

$$\mathcal{X}_{k-1}^{(i+n)} = \hat{\mathbf{x}}_{k-1|k-1} - \sqrt{\frac{n}{1-W_0}}\left(\mathbf{P}_{k-1|k-1}^{1/2}\right)_i, \quad i = 1, 2, \ldots, n,$$

$$W_i = \frac{1-W_0}{2n}, \quad i = 1, 2, \ldots, 2n.$$

2. Compute the predicted moments

$$\hat{\mathbf{x}}_{k|k-1} \approx \sum_{i=0}^{2n} \mathbf{f}(\mathcal{X}_{k-1}^{(i)}) W_i$$

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

PREDICTION IN CKF

1. Form a set of $2n$ σ-points

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

2. Compute the predicted moments

$$\hat{\mathbf{x}}_{k|k-1} \approx \sum_{i=1}^{2n} \mathbf{f}(\mathcal{X}_{k-1}^{(i)}) W_i$$

$$\mathbf{P}_{k|k-1} \approx \mathbf{Q}_{k-1} + \sum_{i=1}^{2n} (\mathbf{f}(\mathcal{X}_{k-1}^{(i)}) - \hat{\mathbf{x}}_{k|k-1})(\cdot)^T W_i.$$

SELF ASSESSMENT

If a σ-point method approximates

$$\mathbb{E}\{\mathbf{h}(\mathbf{x})\} = \int \mathbf{h}(\mathbf{x})\mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}, \mathbf{P})\, d\mathbf{x} \approx \sum_i W_i \mathbf{h}(\mathcal{X}^{(i)})$$

then $\mathbb{E}\{\mathbf{h}(\mathbf{x})(\mathbf{f}(\mathbf{x}) + \mathbf{x})\}$ would be approximated as

- $\sum_i \mathbf{h}(\mathcal{X}^{(i)})W_i \times \left(\sum_i W_i \mathbf{f}(\mathcal{X}^{(i)}) + W_i \mathcal{X}^{(i)}\right)$
- $\sum_i \mathbf{h}(\mathcal{X}^{(i)})W_i \times \left(\sum_i W_i \mathbf{f}(\mathcal{X}^{(i)}) + \sum_i W_i \mathcal{X}^{(i)}\right)$
- $\mathbf{h}(\sum_i \mathcal{X}^{(i)} W_i) \times \left(\mathbf{f}(\sum_i W_i \mathcal{X}^{(i)}) + \sum_i W_i \mathcal{X}^{(i)}\right)$
- $\sum_i W_i \mathbf{h}(\mathcal{X}^{(i)})(\mathbf{f}(\mathcal{X}^{(i)}) + \mathcal{X}^{(i)})$

Check the correct answer.

GAUSSIAN UPDATE BY MOMENT MATCHING

$$\begin{cases} \mathbf{x}_k | \mathbf{y}_{1:k-1} \sim \mathcal{N}(\hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1}) \\ \mathbf{y}_k = h(\mathbf{x}_k) + \mathbf{r}_k, \end{cases} \Rightarrow \begin{cases} \hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{P}_{xy}\mathbf{S}_k^{-1}(\mathbf{y}_k - \hat{\mathbf{y}}_{k|k-1}) \\ \mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{P}_{xy}\mathbf{S}_k^{-1}\mathbf{P}_{xy}^T. \end{cases}$$

Gaussian filter update

$$\hat{\mathbf{y}}_{k|k-1} = \int \mathbf{h}(\mathbf{x}_k)\mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1})\, d\mathbf{x}_k$$

$$\mathbf{P}_{xy} = \int (\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})(\mathbf{h}(\mathbf{x}_k) - \hat{\mathbf{y}}_{k|k-1})^T \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1}) d\mathbf{x}_k$$

$$\mathbf{S}_k = \mathbf{R}_k + \int (\mathbf{h}(\mathbf{x}_k) - \hat{\mathbf{y}}_{k|k-1})(\cdot)^T \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1})\, d\mathbf{x}_k$$

- Objective: show how $\hat{\mathbf{y}}_{k|k-1}$, $\mathbf{P}_{xy}$ and $\mathbf{S}_k$ can be computed using the unscented transform and the cubature rule.

UPDATE IN UKF

1. Form a set of $2n+1$ σ-points

$$\mathcal{X}_k^{(0)} = \hat{\mathbf{x}}_{k|k-1}, \quad W_i = \frac{1-W_0}{2n}, \; i > 1,$$

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

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

$N(x_k; \hat{x}_{k|k-1}, P_{k|k-1})$

2. Compute the desired moments

$$\hat{\mathbf{y}}_{k|k-1} \approx \sum_{i=0}^{2n} \mathbf{h}(\mathcal{X}_k^{(i)}) W_i$$

$$\mathbf{P}_{xy} \approx \sum_{i=0}^{2n} \left(\mathcal{X}_k^{(i)} - \hat{\mathbf{x}}_{k|k-1}\right) \left(\mathbf{h}\left(\mathcal{X}_k^{(i)}\right) - \hat{\mathbf{y}}_{k|k-1}\right)^T W_i$$

$$\mathbf{S}_k \approx \mathbf{R}_k + \sum_{i=0}^{2n} \left(\mathbf{h}\left(\mathcal{X}_k^{(i)}\right) - \hat{\mathbf{y}}_{k|k-1}\right) (\cdot)^T W_i$$

UPDATE IN CKF

1. Form a set of $2n$ σ-points

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

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

$$W_i = \frac{1}{2n}, \qquad i = 1, 2, \ldots, 2n.$$

2. Compute the desired moments

$$\hat{\mathbf{y}}_{k|k-1} \approx \sum_{i=1}^{2n} \mathbf{h}(\mathcal{X}_k^{(i)}) W_i$$

$$\mathbf{P}_{xy} \approx \sum_{i=1}^{2n} \left(\mathcal{X}_k^{(i)} - \hat{\mathbf{x}}_{k|k-1}\right)(\mathbf{h}(\mathcal{X}_k^{(i)}) - \hat{\mathbf{y}}_{k|k-1})^T W_i$$

$$\mathbf{S}_k \approx \mathbf{R}_k + \sum_{i=1}^{2n} (\mathbf{h}(\mathcal{X}_k^{(i)}) - \hat{\mathbf{y}}_{k|k-1})(\cdot)^T W_i$$

SELF ASSESSMENT

If a σ-point method is exact for any polynomial up to degree 9 and $\mathbf{h}(\mathbf{x}_k)$ is a polynomial of degree 3, then the σ-point method would help us to:

- Prove that the posterior distribution, $p(\mathbf{x}_k | \mathbf{y}_{1:k})$ is Gaussian.
- Compute the cross covariance $\mathbf{P}_{xy}$ exactly.
- Compute the posterior mean, $\hat{\mathbf{x}}_{k|k}$, exactly.
- Compute the predicted measurement $\hat{\mathbf{y}}_{k|k-1}$ exactly.

Check all that apply.