Lecture 5: Unscented Kalman filter and Particle Filtering

Simo Särkkä

Department of Biomedical Engineering and Computational Science
Helsinki University of Technology

April 21, 2009

Contents

Linearization Based Gaussian Approximation

- Problem: Determine the mean and covariance of y:

$$x \sim \mathrm{N}(\mu, \sigma^2)$$
$$y = \sin(x)$$

- Linearization based approximation:

$$y = \sin(\mu) + \frac{\partial \sin(\mu)}{\partial \mu}(x - \mu) + \ldots$$

which gives

$$\mathrm{E}[y] \approx \mathrm{E}[\sin(\mu) + \cos(\mu)(x - \mu)] = \sin(\mu)$$
$$\mathrm{Cov}[y] \approx \mathrm{E}[(\sin(\mu) + \cos(\mu)(x - \mu) - \sin(\mu))^2] = \cos^2(\mu)\,\sigma^2.$$

Principle of Unscented Transform [1/3]

- Form 3 sigma points as follows:

$$\begin{aligned} X_0 &= \mu \\ X_1 &= \mu + \sigma \\ X_2 &= \mu - \sigma. \end{aligned}$$

- We may now select some weights W_0, W_1, W_2 such that the original mean and (co)variance can be always recovered by

$$\begin{aligned} \mu &= \sum_i W_i \, x_i \\ \sigma^2 &= \sum_i W_i \, (X_i - \mu)^2. \end{aligned}$$

Principle of Unscented Transform [2/3]

- Use the same formula for approximating the distribution of $y = \sin(x)$ as follows:

$$\mu_y = \sum_i W_i \sin(X_i)$$

$$\sigma_y^2 = \sum_i W_i (\sin(X_i) - \mu_y)^2.$$

- For vectors $\mathbf{x} \sim \mathrm{N}(\mathbf{m}, \mathbf{P})$ the generalization of standard deviation σ is the Cholesky factor $\mathbf{L} = \sqrt{\mathbf{P}}$:

$$\mathbf{P} = \mathbf{L}\,\mathbf{L}^T.$$

- The sigma points can be formed using columns of $\mathbf{L}$ (here c is a suitable positive constant):

$$\mathbf{X}_0 = \mathbf{m}$$

$$\mathbf{X}_i = \mathbf{m} + c\,\mathbf{L}_i$$

$$\mathbf{X}_{n+i} = \mathbf{m} - c\,\mathbf{L}_i$$

Principle of Unscented Transform [3/3]

- For transformation $\mathbf{y} = \mathbf{g}(\mathbf{x})$ the approximation is:

$$\mu_y = \sum_i W_i \, \mathbf{g}(\mathbf{X}_i)$$

$$\Sigma_y = \sum_i W_i \, (\mathbf{g}(\mathbf{X}_i) - \mu_y) \, (\mathbf{g}(\mathbf{X}_i) - \mu_y)^T.$$

- Joint distribution of $\mathbf{x}$ and $\mathbf{y} = \mathbf{g}(\mathbf{x}) + \mathbf{q}$ is then given as

$$\mathrm{E}\left[\begin{pmatrix} \mathbf{x} \\ \mathbf{g}(\mathbf{x}) + \mathbf{q} \end{pmatrix} \,\middle|\, \mathbf{q} \right] \approx \sum_i W_i \begin{pmatrix} \mathbf{X}_i \\ \mathbf{g}(\mathbf{X}_i) \end{pmatrix} = \begin{pmatrix} \mathbf{m} \\ \mu_y \end{pmatrix}$$

$$\mathrm{Cov}\left[\begin{pmatrix} \mathbf{x} \\ \mathbf{g}(\mathbf{x}) + \mathbf{q} \end{pmatrix} \,\middle|\, \mathbf{q} \right]$$

$$\approx \sum_i W_i \begin{pmatrix} (\mathbf{X}_i - \mathbf{m}) \, (\mathbf{X}_i - \mathbf{m})^T & (\mathbf{X}_i - \mathbf{m}) \, (\mathbf{g}(\mathbf{X}_i) - \mu_y)^T \\ (\mathbf{g}(\mathbf{X}_i) - \mu_y) \, (\mathbf{X}_i - \mathbf{m})^T & (\mathbf{g}(\mathbf{X}_i) - \mu_y) \, (\mathbf{g}(\mathbf{X}_i) - \mu_y)^T \end{pmatrix}$$

Unscented Transform Approximation of Non-Linear Transforms [1/3]

Unscented transform

The unscented transform approximation to the joint distribution of $\mathbf{x}$ and $\mathbf{y} = \mathbf{g}(\mathbf{x}) + \mathbf{q}$ where $\mathbf{x} \sim \mathrm{N}(\mathbf{m}, \mathbf{P})$ and $\mathbf{q} \sim \mathrm{N}(\mathbf{0}, \mathbf{Q})$ is

$$\begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} \sim \mathrm{N}\left(\begin{pmatrix} \mathbf{m} \\ \boldsymbol{\mu}_U \end{pmatrix}, \begin{pmatrix} \mathbf{P} & \mathbf{C}_U \\ \mathbf{C}_U^T & \mathbf{S}_U \end{pmatrix} \right),$$

The sub-matrices are formed as follows:

1. Form the matrix of sigma points $\mathbf{X}$ as

$$\mathbf{X} = \begin{bmatrix} \mathbf{m} & \cdots & \mathbf{m} \end{bmatrix} + \sqrt{n + \lambda} \begin{bmatrix} 0 & \sqrt{\mathbf{P}} & -\sqrt{\mathbf{P}} \end{bmatrix},$$

[continues in the next slide...]

Unscented Transform Approximation of Non-Linear Transforms [2/3]

Unscented transform (cont.)

2. Propagate the sigma points through $\mathbf{g}(\cdot)$:

$$\mathbf{Y}_i = \mathbf{g}(\mathbf{X}_i), \quad i = 1 \ldots 2n+1,$$

3. The sub-matrices are then given as:

$$\begin{aligned}
\boldsymbol{\mu}_U &= \sum_i W_{i-1}^{(m)} \mathbf{Y}_i \\
\mathbf{S}_U &= \sum_i W_{i-1}^{(c)} (\mathbf{Y}_i - \boldsymbol{\mu}_U)(\mathbf{Y}_i - \boldsymbol{\mu}_U)^T + \mathbf{Q} \\
\mathbf{C}_U &= \sum_i W_{i-1}^{(c)} (\mathbf{X}_i - \mathbf{m})(\mathbf{Y}_i - \boldsymbol{\mu}_U)^T,
\end{aligned}$$

Unscented Transform Approximation of Non-Linear Transforms [3/3]

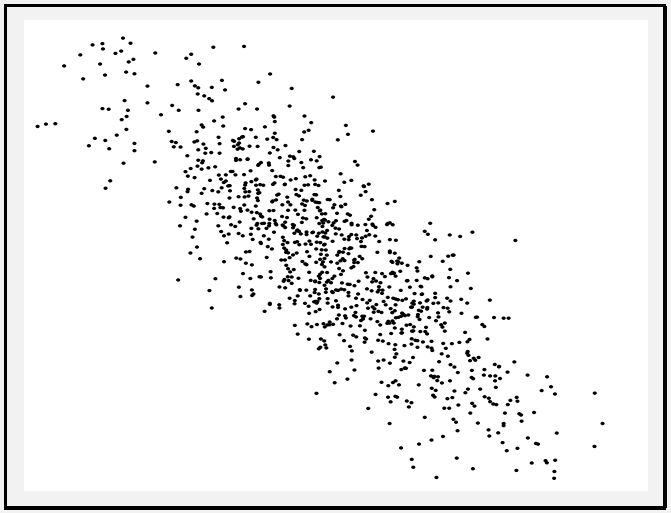
Unscented transform (cont.)

- λ is a scaling parameter defined as $\lambda = \alpha^2 (n + \kappa) - n$.
- α and κ determine the spread of the sigma points.
- Weights $W_i^{(m)}$ and $W_i^{(c)}$ are given as follows:

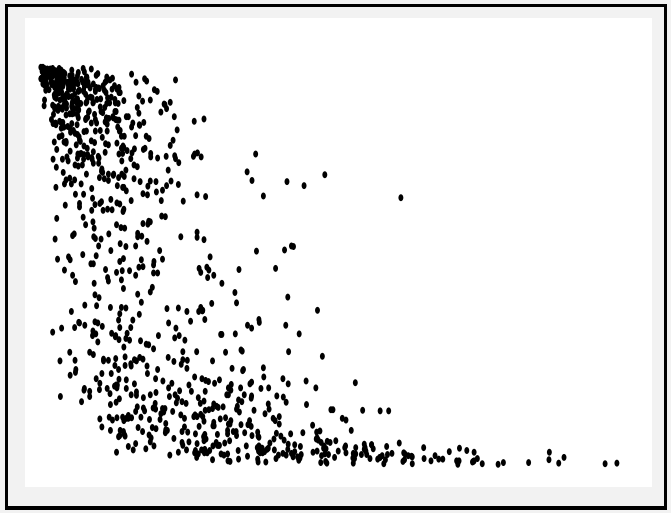
$$
\begin{aligned}
W_0^{(m)} &= \lambda/(n+\lambda) \\
W_0^{(c)} &= \lambda/(n+\lambda) + (1 - \alpha^2 + \beta) \\
W_i^{(m)} &= 1/\{2(n+\lambda)\}, \quad i = 1, \ldots, 2n \\
W_i^{(c)} &= 1/\{2(n+\lambda)\}, \quad i = 1, \ldots, 2n,
\end{aligned}
$$

- β can be used for incorporating prior information on the (non-Gaussian) distribution of $\mathbf{x}$.

Linearization/UT Example



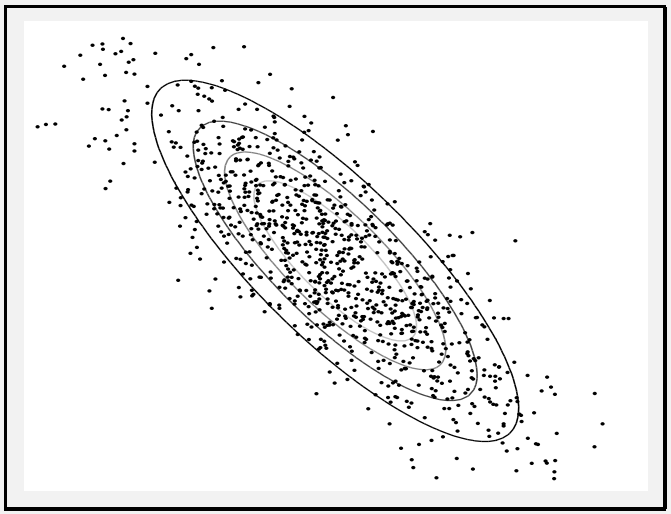
$\Longrightarrow$



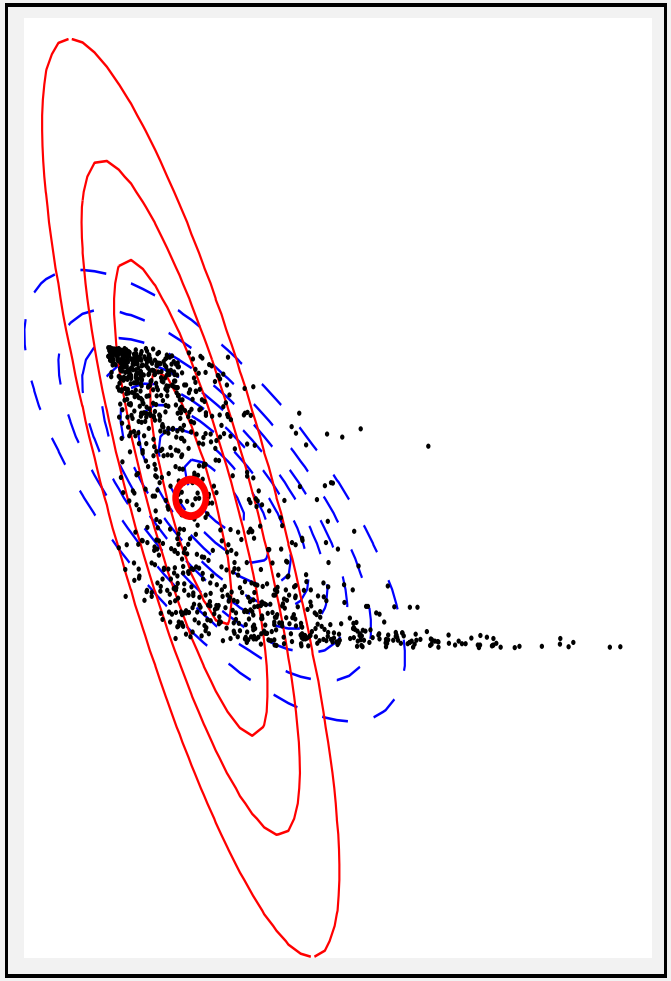
$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \sim \mathrm{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 2 & -2 \\ -2 & 3 \end{pmatrix} \right)$$

$$\begin{aligned} \frac{\mathrm{d}y_1}{\mathrm{d}t} &= \exp(-y_1), \quad & y_1(0) &= x_1 \\ \frac{\mathrm{d}y_2}{\mathrm{d}t} &= -\frac{1}{2} y_2^3, & y_2(0) &= x_2 \end{aligned}$$

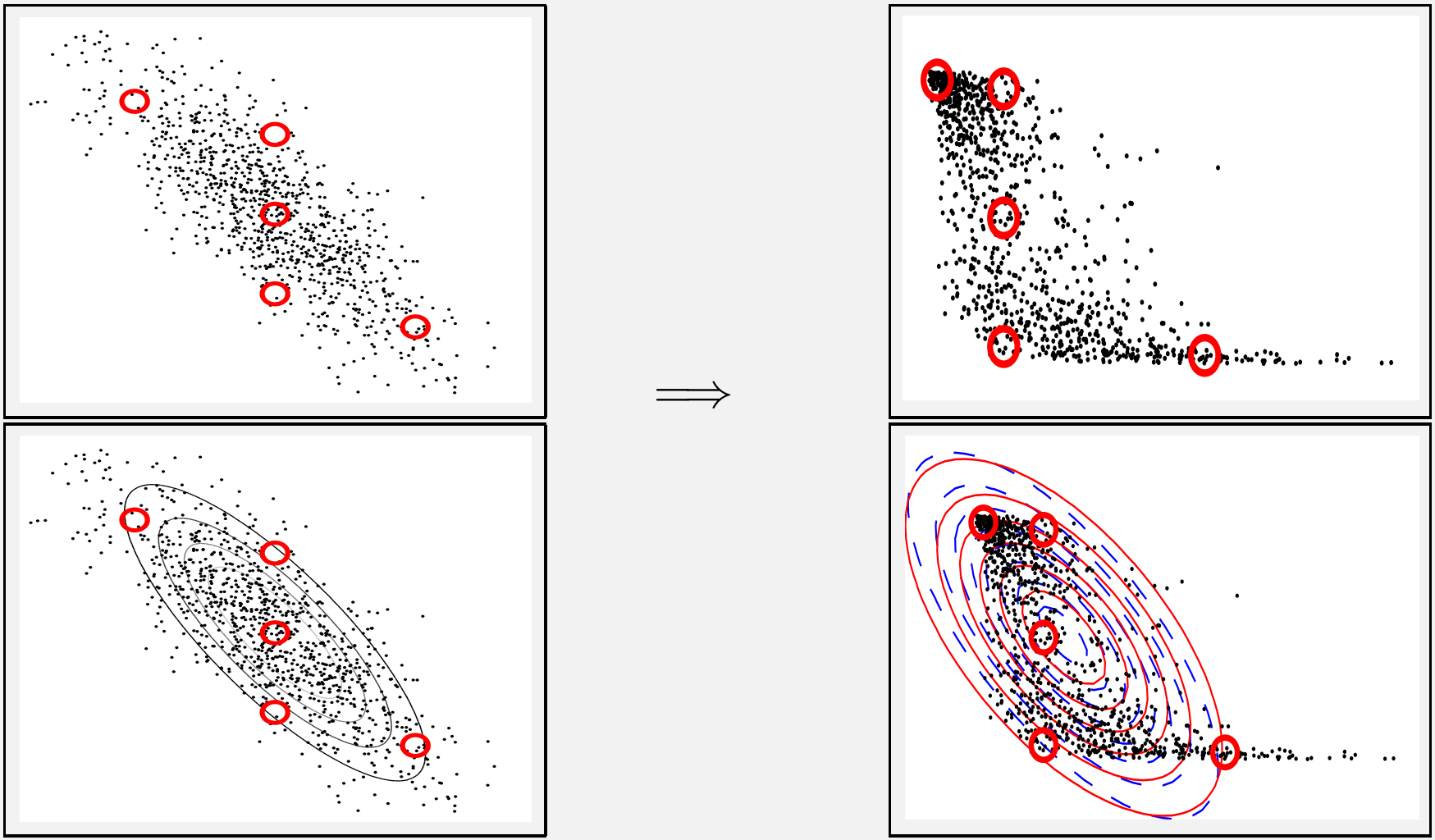
Linearization Approximation



$\Longrightarrow$



UT Approximation



Unscented Kalman Filter (UKF): Derivation [1/4]

- Assume that the filtering distribution of previous step is Gaussian

$$p(\mathbf{x}_{k-1} \mid \mathbf{y}_{1:k-1}) \approx \mathrm{N}(\mathbf{x}_{k-1} \mid \mathbf{m}_{k-1}, \mathbf{P}_{k-1})$$

- The joint distribution of $\mathbf{x}_k$ and $\mathbf{x}_{k-1} = \mathbf{f}(\mathbf{x}_{k-1}) + \mathbf{q}_{k-1}$ can be approximated with UT as Gaussian

$$p(\mathbf{x}_{k-1}, \mathbf{x}_k, \mid \mathbf{y}_{1:k-1}) \approx \mathrm{N}\left(\begin{bmatrix} \mathbf{x}_{k-1} \\ \mathbf{x}_k \end{bmatrix} \;\middle|\; \begin{pmatrix} \mathbf{m}'_1 \\ \mathbf{m}'_2 \end{pmatrix}, \begin{pmatrix} \mathbf{P}'_{11} & \mathbf{P}'_{12} \\ (\mathbf{P}'_{12})^T & \mathbf{P}'_{22} \end{pmatrix} \right),$$

- Form the sigma points $\mathbf{X}_i$ of $\mathbf{x}_{k-1} \sim \mathrm{N}(\mathbf{m}_{k-1}, \mathbf{P}_{k-1})$ and compute the transformed sigma points as $\hat{\mathbf{X}}_i = \mathbf{f}(\mathbf{X}_i)$.
- The expected values can now be expressed as:

$$\mathbf{m}'_1 = \mathbf{m}_{k-1}$$
$$\mathbf{m}'_2 = \sum_i W^{(m)}_{i-1} \hat{\mathbf{x}}_i$$

Unscented Kalman Filter (UKF): Derivation [2/4]

- The blocks of covariance can be expressed as:

$$\mathbf{P}'_{11} = \mathbf{P}_k$$
$$\mathbf{P}'_{12} = \sum_i W^{(c)}_{i-1} (\mathbf{X}_i - \mathbf{m}_{k-1}) (\hat{\mathbf{X}}_i - \mathbf{m}'_2)^T$$
$$\mathbf{P}'_{22} = \sum_i W^{(c)}_{i-1} (\hat{\mathbf{X}}_i - \mathbf{m}'_2) (\hat{\mathbf{X}}_i - \mathbf{m}'_2)^T + \mathbf{Q}_{k-1}$$

- The prediction mean and covariance of $\mathbf{x}_k$ are then $\mathbf{m}'_2$ and $\mathbf{P}'_{22}$, and thus we get

$$\mathbf{m}^-_k = \sum_i W^{(m)}_{i-1} \hat{\mathbf{X}}_i$$
$$\mathbf{P}^-_k = \sum_i W^{(c)}_{i-1} (\hat{\mathbf{X}}_i - \mathbf{m}^-_k) (\hat{\mathbf{X}}_i - \mathbf{m}^-_k)^T + \mathbf{Q}_{k-1}$$

Unscented Kalman Filter (UKF): Derivation [3/4]

- For the joint distribution of $\mathbf{x}_k$ and $\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{r}_k$ we similarly get

$$p(\mathbf{x}_k, \mathbf{y}_k, \mid \mathbf{y}_{1:k-1}) \approx \mathrm{N}\left(\begin{bmatrix} \mathbf{x}_k \\ \mathbf{y}_k \end{bmatrix} \;\middle|\; \begin{pmatrix} \mathbf{m}_1'' \\ \mathbf{m}_2'' \end{pmatrix}, \begin{pmatrix} \mathbf{P}_{11}'' & \mathbf{P}_{12}'' \\ (\mathbf{P}_{12}'')^T & \mathbf{P}_{22}'' \end{pmatrix} \right),$$

- If $\mathbf{X}_i^-$ are the sigma points of $\mathbf{x}_k \sim \mathrm{N}(\mathbf{m}_k^-, \mathbf{P}_k^-)$ and $\hat{\mathbf{Y}}_i = \mathbf{f}(\mathbf{X}_i^-)$, we get:

$$\begin{aligned}
\mathbf{m}_1'' &= \mathbf{m}_k^- \\
\mathbf{m}_2'' &= \sum_i W_{i-1}^{(m)} \, \hat{\mathbf{Y}}_i \\
\mathbf{P}_{11}'' &= \mathbf{P}_k^- \\
\mathbf{P}_{12}'' &= \sum_i W_{i-1}^{(c)} (\mathbf{X}_i^- - \mathbf{m}_k^-) \, (\hat{\mathbf{Y}}_i - \mathbf{m}_2'')^T \\
\mathbf{P}_{22}'' &= \sum_i W_{i-1}^{(c)} (\hat{\mathbf{Y}}_i - \mathbf{m}_2'') \, (\hat{\mathbf{Y}}_i - \mathbf{m}_2'')^T + \mathbf{R}_k
\end{aligned}$$

Unscented Kalman Filter (UKF): Derivation [4/4]

- Recall that if

$$\begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} \sim \mathrm{N}\left(\begin{pmatrix} \mathbf{a} \\ \mathbf{b} \end{pmatrix}, \begin{pmatrix} \mathbf{A} & \mathbf{C} \\ \mathbf{C}^T & \mathbf{B} \end{pmatrix} \right),$$

  then

$$\mathbf{x} \,|\, \mathbf{y} \sim \mathrm{N}(\mathbf{a} + \mathbf{C}\,\mathbf{B}^{-1}\,(\mathbf{y} - \mathbf{b}), \mathbf{A} - \mathbf{C}\,\mathbf{B}^{-1}\mathbf{C}^T).$$

- Thus we get the conditional mean and covariance:

$$\begin{aligned} \mathbf{m}_k &= \mathbf{m}_k^- + \mathbf{P}''_{12}\,(\mathbf{P}''_{22})^{-1}(\mathbf{y}_k - \mathbf{m}''_2) \\ \mathbf{P}_k &= \mathbf{P}_k^- - \mathbf{P}''_{12}\,(\mathbf{P}''_{22})^{-1}\,(\mathbf{P}''_{12})^T. \end{aligned}$$

Unscented Kalman Filter (UKF): Algorithm [1/3]

Unscented Kalman filter: Prediction step

1. Form the matrix of sigma points:

$$\mathbf{X}_{k-1} = \begin{bmatrix} \mathbf{m}_{k-1} & \cdots & \mathbf{m}_{k-1} \end{bmatrix} + \sqrt{n+\lambda} \begin{bmatrix} 0 & \sqrt{\mathbf{P}_{k-1}} & -\sqrt{\mathbf{P}_{k-1}} \end{bmatrix}.$$

2. Propagate the sigma points through the dynamic model:

$$\hat{\mathbf{X}}_{k,i} = \mathbf{f}(\mathbf{X}_{k-1,i}), \quad i = 1 \ldots 2n+1.$$

3. Compute the predicted mean and covariance:

$$\mathbf{m}_k^- = \sum_i W_{i-1}^{(m)} \hat{\mathbf{X}}_{k,i}$$

$$\mathbf{P}_k^- = \sum_i W_{i-1}^{(c)} (\hat{\mathbf{X}}_{k,i} - \mathbf{m}_k^-)(\hat{\mathbf{X}}_{k,i} - \mathbf{m}_k^-)^T + \mathbf{Q}_{k-1}.$$

Unscented Kalman Filter (UKF): Algorithm [2/3]

Unscented Kalman filter: Update step

1. Form the matrix of sigma points:

$$\mathbf{X}_k^- = \begin{bmatrix} \mathbf{m}_k^- & \cdots & \mathbf{m}_k^- \end{bmatrix} + \sqrt{n+\lambda} \begin{bmatrix} 0 & \sqrt{\mathbf{P}_k^-} & -\sqrt{\mathbf{P}_k^-} \end{bmatrix}.$$

2. Propagate sigma points through the measurement model:

$$\hat{\mathbf{Y}}_{k,i} = \mathbf{h}(\mathbf{X}_{k,i}^-), \quad i = 1 \ldots 2n+1.$$

3. Compute the following terms:

$$\begin{aligned}
\boldsymbol{\mu}_k &= \sum_i W_{i-1}^{(m)} \, \hat{\mathbf{Y}}_{k,i} \\
\mathbf{S}_k &= \sum_i W_{i-1}^{(c)} \, (\hat{\mathbf{Y}}_{k,i} - \boldsymbol{\mu}_k) \, (\hat{\mathbf{Y}}_{k,i} - \mu_k)^T + \mathbf{R}_k \\
\mathbf{C}_k &= \sum_i W_{i-1}^{(c)} \, (\mathbf{X}_{k,i}^- - \mathbf{m}_k^-) \, (\hat{\mathbf{Y}}_{k,i} - \boldsymbol{\mu}_k)^T.
\end{aligned}$$

Unscented Kalman Filter (UKF): Algorithm [3/3]

Unscented Kalman filter: Update step (cont.)

4. Compute the filter gain $\mathbf{K}_k$ and the filtered state mean $\mathbf{m}_k$ and covariance $\mathbf{P}_k$, conditional to the measurement $\mathbf{y}_k$:

$$\begin{aligned}
\mathbf{K}_k &= \mathbf{C}_k \, \mathbf{S}_k^{-1} \\
\mathbf{m}_k &= \mathbf{m}_k^- + \mathbf{K}_k \, [\mathbf{y}_k - \boldsymbol{\mu}_k] \\
\mathbf{P}_k &= \mathbf{P}_k^- - \mathbf{K}_k \, \mathbf{S}_k \, \mathbf{K}_k^T.
\end{aligned}$$

Unscented Kalman Filter (UKF): Example

- Recall the discretized pendulum model

$$\begin{pmatrix} x_k^1 \\ x_k^2 \end{pmatrix} = \underbrace{\begin{pmatrix} x_{k-1}^1 + x_{k-1}^2 \, \Delta t \\ x_{k-1}^2 - g \, \sin(x_{k-1}^1) \, \Delta t \end{pmatrix}}_{\mathbf{f}(\mathbf{x_{k-1}})} + \begin{pmatrix} 0 \\ q_{k-1} \end{pmatrix}$$

$$y_k = \underbrace{\sin(x_k^1)}_{\mathbf{h}(\mathbf{x_k})} + r_k,$$

- $\Rightarrow$ *Matlab demonstration*

Unscented Kalman Filter (UKF): Advantages

- No closed form derivatives or expectations needed.
- Not a local approximation, but based on values on a larger area.
- Functions **f** and **h** do not need to be differentiable.
- Theoretically, captures higher order moments of distribution than linearization.

Unscented Kalman Filter (UKF): Disadvantage

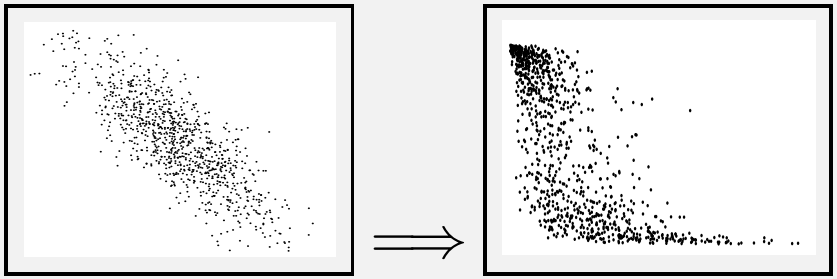
- Not a truly global approximation, based on a small set of trial points.
- Does not work well with nearly singular covariances, i.e., with nearly deterministic systems.
- Requires more computations than EKF or SLF, e.g., Cholesky factorizations on every step.
- Can only be applied to models driven by Gaussian noises.

Particle Filtering: Overview [1/3]

Demo: Kalman vs. Particle Filtering:

- ▸ Kalman filter animation
- ▸ Particle filter animation

Particle Filtering: Overview [2/3]



- The idea is to form a weighted particle presentation $(\mathbf{x}^{(i)}, w^{(i)})$ of the posterior distribution:

$$p(\mathbf{x}) \approx \sum_i w^{(i)} \, \delta(\mathbf{x} - \mathbf{x}^{(i)}).$$

- Particle filtering = Sequential importance sampling, with additional resampling step.
- Bootstrap filter (also called Condensation) is the simplest particle filter.

Particle Filtering: Overview [3/3]

- The efficiency of particle filter is determined by the selection of the importance distribution.
- The importance distribution can be formed by using e.g. EKF or UKF.
- Sometimes the optimal importance distribution can be used, and it minimizes the variance of the weights.
- Rao-Blackwellization: Some components of the model are marginalized in closed form $\Rightarrow$ hybrid particle/Kalman filter.

Bootstrap Filter: Principle

- State density representation is set of samples $\{\mathbf{x}_k^{(i)} \ : \ i = 1, \ldots, N\}$.
- Bootstrap filter performs optimal filtering update and prediction steps using Monte Carlo.
- Prediction step performs prediction for each particle separately.
- Equivalent to integrating over the distribution of previous step (as in Kalman Filter).
- Update step is implemented with weighting and resampling.

Bootstrap Filter: Algorithm

Bootstrap Filter

1. Generate sample from predictive density of each old sample point $\mathbf{x}_{k-1}^{(i)}$:

$$\tilde{\mathbf{x}}_k^{(i)} \sim p(\mathbf{x}_k \mid \mathbf{x}_{k-1}^{(i)}).$$

2. Evaluate and normalize weights for each new sample point $\tilde{\mathbf{x}}_k^{(i)}$:

$$w_k^{(i)} = p(\mathbf{y}_k \mid \tilde{\mathbf{x}}_k^{(i)}).$$

3. Resample by selecting new samples $\mathbf{x}_k^{(i)}$ from set $\{\tilde{\mathbf{x}}_k^{(i)}\}$ with probabilities proportional to $w_k^{(i)}$.

Sequential Importance Resampling: Principle

- State density representation is set of weighted samples $\{(\mathbf{x}_k^{(i)}, w_k^{(i)}) \ : \ i = 1, \ldots, N\}$ such that for arbitrary function $\mathbf{g}(\mathbf{x}_k)$, we have

$$\mathrm{E}[\mathbf{g}(\mathbf{x}_k) \,|\, \mathbf{y}_{1:k}] \approx \sum_i w_k^{(i)} \, \mathbf{g}(\mathbf{x}_k^{(i)}).$$

- On each step, we first draw samples from an importance distribution $\pi(\cdot)$, which is chosen suitably.
- The prediction and update steps are combined and consist of computing new weights from the old ones $w_{k-1}^{(i)} \rightarrow w_k^{(i)}$.
- If the "sample diversity" i.e the effective number of different samples is too low, do resampling.

Sequential Importance Resampling: Algorithm

Sequential Importance Resampling

1. Draw new point $\mathbf{x}_k^{(i)}$ for each point in the sample set $\{\mathbf{x}_{k-1}^{(i)}, i = 1, \ldots, N\}$ from the importance distribution:

$$\mathbf{x}_k^{(i)} \sim \pi(\mathbf{x}_k \mid \mathbf{x}_{k-1}^{(i)}, \mathbf{y}_{1:k}), \qquad i = 1, \ldots, N.$$

2. Calculate new weights

$$w_k^{(i)} = w_{k-1}^{(i)} \frac{p(\mathbf{y}_k \mid \mathbf{x}_k^{(i)})\, p(\mathbf{x}_k^{(i)} \mid \mathbf{x}_{k-1}^{(i)})}{\pi(\mathbf{x}_k^{(i)} \mid \mathbf{x}_{k-1}^{(i)}, \mathbf{y}_{1:k})}, \qquad i = 1, \ldots, N.$$

   and normalize them to sum to unity.

3. If the effective number of particles is too low, perform resampling.

Effective Number of Particles and Resampling

The estimate for the effective number of particles can be computed as:

$$n_{\text{eff}} \approx \frac{1}{\sum_{i=1}^{N} \left(w_k^{(i)} \right)^2},$$

Resampling

1. Interpret each weight $w_k^{(i)}$ as the probability of obtaining the sample index i in the set $\{\mathbf{x}_k^{(i)} \mid i = 1, \ldots, N\}$.
2. Draw N samples from that discrete distribution and replace the old sample set with this new one.
3. Set all weights to the constant value $w_k^{(i)} = 1/N$.

Constructing the Importance Distribution

- Use the dynamic model as the importance distribution $\Rightarrow$ Bootstrap filter.
- Use the optimal importance distribution

$$\pi(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \mathbf{y}_{1:k}) = p(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \mathbf{y}_{1:k}).$$

- Approximate the optimal importance distribution by UKF $\Rightarrow$ unscented particle filter.
- Instead of UKF also EKF or SLF can be, for example, used.
- Simulate availability of optimal importance distribution $\Rightarrow$ auxiliary SIR (ASIR) filter.

Rao-Blackwellized Particle Filtering: Principle [1/2]

- Consider a conditionally Gaussian model of the form

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

- The model is of the form

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

- The full model is non-linear and non-Gaussian in general.
- But given the values $\mathbf{s}_k$ the model is Gaussian and thus Kalman filter equations can be used.

Rao-Blackwellized Particle Filtering: Principle [1/2]

- The idea of the Rao-Blackwellized particle filter:
  - Use Monte Carlo sampling to the values $\mathbf{s}_k$
  - Given these values, compute distribution of $\mathbf{x}_k$ with Kalman filter equations.
  - Result is a Mixture Gaussian distribution, where each particle consist of value $\mathbf{s}_k^{(i)}$, associated weight $w_k^{(i)}$ and the mean and covariance conditional to the history $\mathbf{s}_{1:k}^{(i)}$
- The explicit RBPF equations can be found in the lecture notes.
- If the model is almost conditionally Gaussian, it is also possible to use EKF, SLF or UKF instead of the linear KF.

Particle Filter: Advantages

- No restrictions in model – can be applied to non-Gaussian models, hierarchical models etc.
- Global approximation.
- Approaches the exact solution, when the number of samples goes to infinity.
- In its basic form, very easy to implement.
- Superset of other filtering methods – Kalman filter is a Rao-Blackwellized particle filter with one particle.

Particle Filter: Disadvantages

- Computational requirements much higher than of the Kalman filters.
- Problems with nearly noise-free models, especially with accurate dynamic models.
- Good importance distributions and efficient Rao-Blackwellized filters quite tricky to implement.
- Very hard to find programing errors (i.e., to debug).

Summary

- Unscented transform (UT) approximates transformations of Gaussian variables by propagating sigma points through the non-linearity.
- In UT the mean and covariance are approximated as linear combination of the sigma points.
- The unscented Kalman filter uses unscented transform for computing the approximate means and covariance in non-linear filtering problems.
- Particle filters use weighted set of samples (particles) for approximating the filtering distributions.
- Sequential importance resampling (SIR) is the general framework and bootstrap filter is a simple special case of it.
- In Rao-Blackwellized particle filters a part of the state is sampled and part is integrated in closed form with Kalman filter.

Matlab Demo

[Tracking of pendulum with EKF, SLF, UKF and BF]