Sigma Point Belief Propagation

Florian Meyer

joint work with Ondrej Hlinka and Franz Hlawatsch

Institute of Telecommunications

Vienna University of Technology

institute of telecommunications

TU WIEN

Outline

Outline

Sigma point basics

- Consider a random vector $\mathbf{x} \in \mathbb{R}^J$ whose mean $\boldsymbol{\mu}_{\mathbf{x}}$ and covariance matrix $\mathbf{C}_{\mathbf{x}}$ are known, and a transformed random vector $\mathbf{y} = H(\mathbf{x})$.

Sigma point basics

- Consider a random vector $\mathbf{x} \in \mathbb{R}^J$ whose mean $\boldsymbol{\mu}_{\mathbf{x}}$ and covariance matrix $\mathbf{C}_{\mathbf{x}}$ are known, and a transformed random vector $\mathbf{y} = H(\mathbf{x})$.
- Sigma points (SPs) [Julier et al., 1997] are used to calculate approximations of the mean $\boldsymbol{\mu}_{\mathbf{y}}$, covariance matrix $\mathbf{C}_{\mathbf{y}}$, and cross-covariance matrix $\mathbf{C}_{\mathbf{x},\mathbf{y}}$.

Sigma point basics

- Consider a random vector $\mathbf{x} \in \mathbb{R}^J$ whose mean $\boldsymbol{\mu}_\mathbf{x}$ and covariance matrix $\mathbf{C}_\mathbf{x}$ are known, and a transformed random vector $\mathbf{y} = H(\mathbf{x})$.
- Sigma points (SPs) [Julier et al., 1997] are used to calculate approximations of the mean $\boldsymbol{\mu}_\mathbf{y}$, covariance matrix $\mathbf{C}_\mathbf{y}$, and cross-covariance matrix $\mathbf{C}_{\mathbf{x},\mathbf{y}}$.
- The resulting approximations are at least as good as those obtained by linearizing $H(\cdot)$.
- Fundamental difference to Monte Carlo type methods: The SPs are not random samples but are calculated by means of a deterministic algorithm.

Calculation of sigma points

- For a J-dimensional random vector $\mathbf{x}$, SPs and corresponding weights $\{(\mathbf{x}^{(j)}, w^{(j)})\}_{j=0}^{2J}$ are calculated as [Julier et al., 1997]

$$\mathbf{x}^{(j)} = \begin{cases} \boldsymbol{\mu}_{\mathbf{x}}, & j = 0 \\ \boldsymbol{\mu}_{\mathbf{x}} + \sqrt{J+\kappa}\,\left(\mathbf{C}_{\mathbf{x}}^{1/2}\right)_j, & j = 1, \ldots, J \\ \boldsymbol{\mu}_{\mathbf{x}} - \sqrt{J+\kappa}\,\left(\mathbf{C}_{\mathbf{x}}^{1/2}\right)_j, & j = J+1, \ldots, 2J \end{cases}$$

$$w^{(j)} = \begin{cases} \frac{\kappa}{J+\kappa}, & j = 0 \\ \frac{1}{2(J+\kappa)}, & j = 1, \ldots, 2J. \end{cases}$$

Here, $\left(\mathbf{C}_{\mathbf{x}}^{1/2}\right)_j$ is the jth row or column of the matrix square root of $\mathbf{C}_{\mathbf{x}}$ and κ is a parameter controlling the spread of the SPs around the mean $\boldsymbol{\mu}_{\mathbf{x}}$.

Calculation of sigma points

- For a J-dimensional random vector $\mathbf{x}$, SPs and corresponding weights $\{(\mathbf{x}^{(j)}, w^{(j)})\}_{j=0}^{2J}$ are calculated as [Julier et al., 1997]

$$\mathbf{x}^{(j)} = \begin{cases} \boldsymbol{\mu}_{\mathbf{x}}, & j = 0 \\ \boldsymbol{\mu}_{\mathbf{x}} + \sqrt{J+\kappa}\left(\mathbf{C}_{\mathbf{x}}^{1/2}\right)_j, & j = 1, \ldots, J \\ \boldsymbol{\mu}_{\mathbf{x}} - \sqrt{J+\kappa}\left(\mathbf{C}_{\mathbf{x}}^{1/2}\right)_j, & j = J+1, \ldots, 2J \end{cases}$$

$$w^{(j)} = \begin{cases} \frac{\kappa}{J+\kappa}, & j = 0 \\ \frac{1}{2(J+\kappa)}, & j = 1, \ldots, 2J. \end{cases}$$

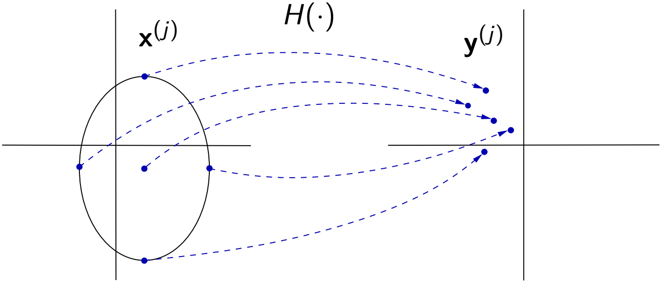
  Here, $\left(\mathbf{C}_{\mathbf{x}}^{1/2}\right)_j$ is the jth row or column of the matrix square root of $\mathbf{C}_{\mathbf{x}}$ and κ is a parameter controlling the spread of the SPs around the mean $\boldsymbol{\mu}_{\mathbf{x}}$.

- SPs have the property that the weighted sample mean $\tilde{\boldsymbol{\mu}}_{\mathbf{x}} = \sum_{j=0}^{2J} w^{(j)}\mathbf{x}^{(j)}$ and weighted sample covariance matrix $\tilde{\mathbf{C}}_{\mathbf{x}} = \sum_{j=0}^{2J} w^{(j)}(\mathbf{x}^{(j)} - \tilde{\boldsymbol{\mu}}_{\mathbf{x}})(\mathbf{x}^{(j)} - \tilde{\boldsymbol{\mu}}_{\mathbf{x}})^{\mathrm{T}}$ are exactly equal to $\boldsymbol{\mu}_{\mathbf{x}}$ and $\mathbf{C}_{\mathbf{x}}$, respectively.

Unscented transformation

- SPs $\{\mathbf{y}^{(j)}\}_{j=0}^{2J}$ of $\mathbf{y}$ can be obtained by propagating each SP $\mathbf{x}^{(j)}$ through $H(\cdot)$ [Julier et al., 1997]:

$$\mathbf{y}^{(j)} = H\big(\mathbf{x}^{(j)}\big), \qquad j = 0, \ldots, 2J.$$



Calculation of mean and covariance

- From $\left\{\left(\mathbf{x}^{(j)}, \mathbf{y}^{(j)}, w^{(j)}\right)\right\}_{j=0}^{2J}$, one can calculate approximations of $\boldsymbol{\mu}_{\mathbf{y}}$, $\mathbf{C}_{\mathbf{y}}$ and $\mathbf{C}_{\mathbf{xy}}$ as

$$\tilde{\boldsymbol{\mu}}_{\mathbf{y}} = \sum_{j=0}^{2J} w^{(j)} \mathbf{y}^{(j)},$$

$$\tilde{\mathbf{C}}_{\mathbf{y}} = \sum_{j=0}^{2J} w^{(j)} (\mathbf{y}^{(j)} - \boldsymbol{\mu}_{\mathbf{y}})(\mathbf{y}^{(j)} - \mu_{\mathbf{y}})^{\mathrm{T}},$$

$$\tilde{\mathbf{C}}_{\mathbf{xy}} = \sum_{j=0}^{2J} w^{(j)} (\mathbf{y}^{(j)} - \boldsymbol{\mu}_{\mathbf{y}})(\mathbf{x}^{(j)} - \mu_{\mathbf{x}})^{\mathrm{T}}.$$

Outline

Bayesian estimation

- SPs can be used for Bayesian estimation of a random vector $\mathbf{x}$ from an observed vector

$$\mathbf{z} = \mathbf{y} + \mathbf{n}\,, \quad \text{with } \mathbf{y} = H(\mathbf{x})\,.$$

Here, the noise $\mathbf{n}$ is zero-mean and statistically independent of $\mathbf{x}$, and has known covariance matrix $\mathbf{C_n}$.

Bayesian estimation

- SPs can be used for Bayesian estimation of a random vector $\mathbf{x}$ from an observed vector

$$\mathbf{z} = \mathbf{y} + \mathbf{n}\,, \quad \text{with } \mathbf{y} = H(\mathbf{x})\,.$$

  Here, the noise $\mathbf{n}$ is zero-mean and statistically independent of $\mathbf{x}$, and has known covariance matrix $\mathbf{C_n}$.

- Bayesian estimation relies on the posterior probability density function (pdf)

$$f(\mathbf{x}|\mathbf{z}) \propto f(\mathbf{z}|\mathbf{x})f(\mathbf{x})\,,$$

  where $f(\mathbf{z}|\mathbf{x})$ is the likelihood function and $f(\mathbf{x})$ is the prior pdf.

Bayesian estimation

- SPs can be used for Bayesian estimation of a random vector $\mathbf{x}$ from an observed vector

$$\mathbf{z} = \mathbf{y} + \mathbf{n}\,, \quad \text{with } \mathbf{y} = H(\mathbf{x})\,.$$

Here, the noise $\mathbf{n}$ is zero-mean and statistically independent of $\mathbf{x}$, and has known covariance matrix $\mathbf{C}_{\mathbf{n}}$.

- Bayesian estimation relies on the posterior probability density function (pdf)

$$f(\mathbf{x}|\mathbf{z}) \propto f(\mathbf{z}|\mathbf{x})f(\mathbf{x})\,,$$

where $f(\mathbf{z}|\mathbf{x})$ is the likelihood function and $f(\mathbf{x})$ is the prior pdf.

- Closed-form calculation of $f(\mathbf{x}|\mathbf{z})$ is usually infeasible!

Bayesian estimation in the linear-Gaussian case

- A feasible special case is when $H(\cdot)$ is linear, i.e., $H(\mathbf{x}) = \mathbf{H}\mathbf{x}$ and $\mathbf{x}$ and $\mathbf{n}$ are Gaussian. Then $f(\mathbf{x}|\mathbf{z})$ is also Gaussian, and $\boldsymbol{\mu}_{\mathbf{x}|\mathbf{z}}$ and $\mathbf{C}_{\mathbf{x}|\mathbf{z}}$ can be calculated as

$$\boldsymbol{\mu}_{\mathbf{x}|\mathbf{z}} = \boldsymbol{\mu}_{\mathbf{x}} + \mathbf{K}(\mathbf{z} - \boldsymbol{\mu}_{\mathbf{y}}), \qquad \mathbf{C}_{\mathbf{x}|\mathbf{z}} = \mathbf{C}_{\mathbf{x}} - \mathbf{K}(\mathbf{C}_{\mathbf{y}} + \mathbf{C}_{\mathbf{n}})\mathbf{K}^{\mathsf{T}},$$

with

$$\begin{aligned} \mathbf{K} &= \mathbf{C}_{\mathbf{xy}}(\mathbf{C}_{\mathbf{y}} + \mathbf{C}_{\mathbf{n}})^{-1}, \\ \boldsymbol{\mu}_{\mathbf{y}} &= \mathbf{H}\boldsymbol{\mu}_{\mathbf{x}}, \\ \mathbf{C}_{\mathbf{y}} &= \mathbf{H}\mathbf{C}_{\mathbf{x}}\mathbf{H}^{\mathsf{T}}, \\ \mathbf{C}_{\mathbf{xy}} &= \mathbf{C}_{\mathbf{x}}\mathbf{H}^{\mathsf{T}}. \end{aligned}$$

Bayesian estimation with sigma points

- In the nonlinear case, $\boldsymbol{\mu}_{\mathbf{x}|\mathbf{z}}$ and $\mathbf{C}_{\mathbf{x}|\mathbf{z}}$ can be approximated by means of SPs [Julier et al., 1997].
- This is done by using the closed-form expressions of the linear-Gaussian case, in which $\boldsymbol{\mu}_{\mathbf{y}}$, $\mathbf{C}_{\mathbf{y}}$, and $\mathbf{C}_{\mathbf{xy}}$ are replaced by the SP approximations $\tilde{\boldsymbol{\mu}}_{\mathbf{y}}$, $\tilde{\mathbf{C}}_{\mathbf{y}}$, and $\tilde{\mathbf{C}}_{\mathbf{xy}}$.

Bayesian estimation with sigma points

- In the nonlinear case, $\boldsymbol{\mu}_{\mathbf{x}|\mathbf{z}}$ and $\mathbf{C}_{\mathbf{x}|\mathbf{z}}$ can be approximated by means of SPs [Julier et al., 1997].
- This is done by using the closed-form expressions of the linear-Gaussian case, in which $\boldsymbol{\mu}_{\mathbf{y}}$, $\mathbf{C}_{\mathbf{y}}$, and $\mathbf{C}_{\mathbf{xy}}$ are replaced by the SP approximations $\tilde{\boldsymbol{\mu}}_{\mathbf{y}}$, $\tilde{\mathbf{C}}_{\mathbf{y}}$, and $\tilde{\mathbf{C}}_{\mathbf{xy}}$.
- SP approximations $\tilde{\boldsymbol{\mu}}_{\mathbf{x}|\mathbf{z}}$ and $\tilde{\mathbf{C}}_{\mathbf{x}|\mathbf{z}}$ are thus obtained as

$$\tilde{\boldsymbol{\mu}}_{\mathbf{x}|\mathbf{z}} = \boldsymbol{\mu}_{\mathbf{x}} + \tilde{\mathbf{K}}(\mathbf{z} - \tilde{\boldsymbol{\mu}}_{\mathbf{y}}) , \qquad \tilde{\mathbf{C}}_{\mathbf{x}|\mathbf{z}} = \mathbf{C}_{\mathbf{x}} - \tilde{\mathbf{K}}(\tilde{\mathbf{C}}_{\mathbf{y}} + \mathbf{C}_{\mathbf{n}})\tilde{\mathbf{K}}^{\mathsf{T}},$$

with $\tilde{\mathbf{K}} = \tilde{\mathbf{C}}_{\mathbf{xy}}(\tilde{\mathbf{C}}_{\mathbf{y}} + \mathbf{C}_{\mathbf{n}})^{-1}$.

Bayesian estimation with sigma points

- In the nonlinear case, $\boldsymbol{\mu}_{\mathbf{x}|\mathbf{z}}$ and $\mathbf{C}_{\mathbf{x}|\mathbf{z}}$ can be approximated by means of SPs [Julier et al., 1997].
- This is done by using the closed-form expressions of the linear-Gaussian case, in which $\boldsymbol{\mu}_{\mathbf{y}}$, $\mathbf{C}_{\mathbf{y}}$, and $\mathbf{C}_{\mathbf{xy}}$ are replaced by the SP approximations $\tilde{\boldsymbol{\mu}}_{\mathbf{y}}$, $\tilde{\mathbf{C}}_{\mathbf{y}}$, and $\tilde{\mathbf{C}}_{\mathbf{xy}}$.
- SP approximations $\tilde{\boldsymbol{\mu}}_{\mathbf{x}|\mathbf{z}}$ and $\tilde{\mathbf{C}}_{\mathbf{x}|\mathbf{z}}$ are thus obtained as

$$\tilde{\boldsymbol{\mu}}_{\mathbf{x}|\mathbf{z}} = \boldsymbol{\mu}_{\mathbf{x}} + \tilde{\mathbf{K}}(\mathbf{z} - \tilde{\boldsymbol{\mu}}_{\mathbf{y}}) , \qquad \tilde{\mathbf{C}}_{\mathbf{x}|\mathbf{z}} = \mathbf{C}_{\mathbf{x}} - \tilde{\mathbf{K}}(\tilde{\mathbf{C}}_{\mathbf{y}} + \mathbf{C}_{\mathbf{n}})\tilde{\mathbf{K}}^{\mathrm{T}},$$

  with $\tilde{\mathbf{K}} = \tilde{\mathbf{C}}_{\mathbf{xy}}(\tilde{\mathbf{C}}_{\mathbf{y}} + \mathbf{C}_{\mathbf{n}})^{-1}$.

- Sequential implementation of this algorithm $\Rightarrow$ SP filter

Outline

Factor graph

- Consider K state vectors $\mathbf{x}_k$, $k \in \{1, \dots, K\}$

Factor graph

- Consider K state vectors $\mathbf{x}_k$, $k \in \{1, \ldots, K\}$
- Assume that the posterior pdf $f(\mathbf{x}|\mathbf{z})$ factorizes as

$$f(\mathbf{x}|\mathbf{z}) \propto \left[\prod_{k=1}^{K} f(\mathbf{x}_k) \right] \prod_{(k',l) \in \mathcal{E}} f(\mathbf{z}_{k',l}|\mathbf{x}_{k'}, \mathbf{x}_l) \,.$$

Here, $\mathbf{z}_{k,l} = H(\mathbf{x}_k, \mathbf{x}_l) + \mathbf{v}_{k,l}$ are noisy “pairwise” observations.

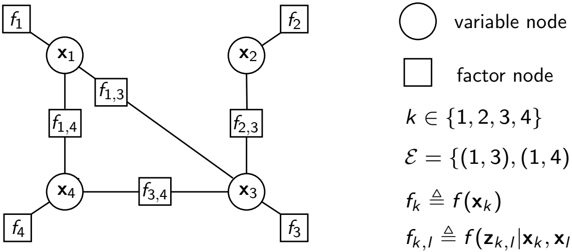
Factor graph

- Consider K state vectors $\mathbf{x}_k$, $k \in \{1, \ldots, K\}$
- Assume that the posterior pdf $f(\mathbf{x}|\mathbf{z})$ factorizes as

$$f(\mathbf{x}|\mathbf{z}) \propto \left[\prod_{k=1}^{K} f(\mathbf{x}_k) \right] \prod_{(k',l)\in\mathcal{E}} f(\mathbf{z}_{k',l}|\mathbf{x}_{k'}, \mathbf{x}_l) \, .$$

Here, $\mathbf{z}_{k,l} = H(\mathbf{x}_k, \mathbf{x}_l) + \mathbf{v}_{k,l}$ are noisy "pairwise" observations.

- Representation by factor graph:



$k \in \{1, 2, 3, 4\}$

$\mathcal{E} = \{(1, 3), (1, 4), (2, 3), (3, 4)\}$

$f_k \triangleq f(\mathbf{x}_k)$

$f_{k,l} \triangleq f(\mathbf{z}_{k,l}|\mathbf{x}_k, \mathbf{x}_l)$

Belief propagation

- To obtain an estimate of $\mathbf{x}_k$, the "marginal" posterior pdf $f(\mathbf{x}_k|\mathbf{z}) = \int f(\mathbf{x}|\mathbf{z})\,d\mathbf{x}^{\sim k}$ is needed.
- Direct calculation of $f(\mathbf{x}_k|\mathbf{z}) = \int f(\mathbf{x}|\mathbf{z})d\mathbf{x}^{\sim k}$ is typically infeasible.

Belief propagation

- To obtain an estimate of $\mathbf{x}_k$, the "marginal" posterior pdf $f(\mathbf{x}_k|\mathbf{z}) = \int f(\mathbf{x}|\mathbf{z})\, d\mathbf{x}^{\sim k}$ is needed.
- Direct calculation of $f(\mathbf{x}_k|\mathbf{z}) = \int f(\mathbf{x}|\mathbf{z}) d\mathbf{x}^{\sim k}$ is typically infeasible.
- An approximate marginal posterior ("belief") $b(\mathbf{x}_k) \approx f(\mathbf{x}_k|\mathbf{z})$ is obtained by running belief propagation (BP) message passing on the factor graph representing the joint posterior $f(\mathbf{x}|\mathbf{z})$.
- In the special case where the factor graph is a tree (no loops/cycles), BP is exact, i.e., $b(\mathbf{x}_k) = f(\mathbf{x}_k|\mathbf{z})$.

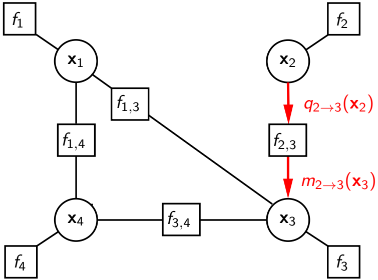
Belief propagation rules

- Let $\mathcal{N}_k$ denote the "neighbor" set of variable node $k \in \{1, \ldots, K\}$, which comprises all variable nodes $l \in \{1, \ldots, K\} \backslash \{k\}$ such that $(k, l) \in \mathcal{E}$.

Belief propagation rules

- Let $\mathcal{N}_k$ denote the "neighbor" set of variable node $k \in \{1, \ldots, K\}$, which comprises all variable nodes $l \in \{1, \ldots, K\} \setminus \{k\}$ such that $(k, l) \in \mathcal{E}$.
- Belief of variable $\mathbf{x}_k$:

$$b(\mathbf{x}_k) \propto f(\mathbf{x}_k) \prod_{l \in \mathcal{N}_k} m_{l \to k}(\mathbf{x}_k).$$



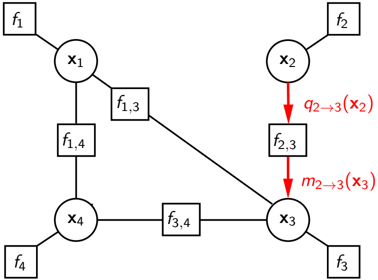
Belief propagation rules

- Let $\mathcal{N}_k$ denote the “neighbor” set of variable node $k \in \{1, \ldots, K\}$, which comprises all variable nodes $l \in \{1, \ldots, K\} \backslash \{k\}$ such that $(k, l) \in \mathcal{E}$.
- Belief of variable $\mathbf{x}_k$:

$$b(\mathbf{x}_k) \propto f(\mathbf{x}_k) \prod_{l \in \mathcal{N}_k} m_{l \to k}(\mathbf{x}_k).$$

- Message from variable node $\mathbf{x}_k$ to function node $f(\mathbf{z}_{k,l} | \mathbf{x}_k, \mathbf{x}_l)$:

$$q_{k \to l}(\mathbf{x}_k) = f(\mathbf{x}_k) \prod_{l' \in \mathcal{N}_k \backslash \{l\}} m_{l' \to k}(\mathbf{x}_k).$$



Belief propagation rules

- Let $\mathcal{N}_k$ denote the "neighbor" set of variable node $k \in \{1, \ldots, K\}$, which comprises all variable nodes $l \in \{1, \ldots, K\} \backslash \{k\}$ such that $(k, l) \in \mathcal{E}$.
- Belief of variable $\mathbf{x}_k$:

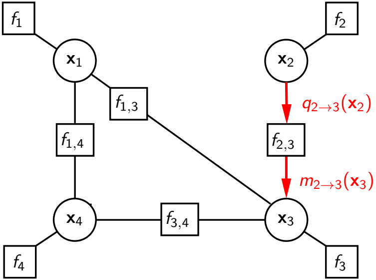
$$b(\mathbf{x}_k) \propto f(\mathbf{x}_k) \prod_{l \in \mathcal{N}_k} m_{l \to k}(\mathbf{x}_k).$$

- Message from variable node $\mathbf{x}_k$ to function node $f(\mathbf{z}_{k,l}|\mathbf{x}_k, \mathbf{x}_l)$:

$$q_{k \to l}(\mathbf{x}_k) = f(\mathbf{x}_k) \prod_{l' \in \mathcal{N}_k \backslash \{l\}} m_{l' \to k}(\mathbf{x}_k).$$

- Message from function node $f(\mathbf{z}_{k,l}|\mathbf{x}_k, \mathbf{x}_l)$ to variable node $\mathbf{x}_k$:

$$m_{l \to k}(\mathbf{x}_k) = \int f(\mathbf{z}_{k,l}|\mathbf{x}_k, \mathbf{x}_l)\, q_{l \to k}(\mathbf{x}_l)\, d\mathbf{x}_l.$$



Outline

Motivation

- The SP filter is an attractive alternative to the Kalman filter and the particle filter in sequential Bayesian estimation problems.

Motivation

- The SP filter is an attractive alternative to the Kalman filter and the particle filter in sequential Bayesian estimation problems.
- In this work, we develop **sigma point belief propagation (SPBP)** by extending the SP filter to Bayesian inference corresponding to a general factorization structure.

Motivation

- The SP filter is an attractive alternative to the Kalman filter and the particle filter in sequential Bayesian estimation problems.
- In this work, we develop **sigma point belief propagation (SPBP)** by extending the SP filter to Bayesian inference corresponding to a general factorization structure.

Factorization Structure → / System Properties ↓	Sequential	General
Linear, Gaussian	Kalman filter [Kalman, 1960]	Gaussian BP [Weiss et al., 2001]
Nonlinear, non-Gaussian	Particle filter [Gordon et al., 1993]	Nonparametric BP [Sudderth et al., 2003]
Nonlinear, Gaussian	SP filter [Julier et al., 1997]	**SPBP**

Reformulation of BP rules

- Recall BP rules:

$$b(\mathbf{x}_k) \propto f(\mathbf{x}_k) \prod_{l \in \mathcal{N}_k} m_{l \to k}(\mathbf{x}_k)$$

$$q_{k \to l}(\mathbf{x}_k) = f(\mathbf{x}_k) \prod_{l' \in \mathcal{N}_k \setminus \{l\}} m_{l' \to k}(\mathbf{x}_k)$$

$$m_{l \to k}(\mathbf{x}_k) = \int f(\mathbf{z}_{k,l} | \mathbf{x}_k, \mathbf{x}_l)\, q_{l \to k}(\mathbf{x}_l)\, d\mathbf{x}_l \,.$$

Reformulation of BP rules

- Recall BP rules:

$$b(\mathbf{x}_k) \propto f(\mathbf{x}_k) \prod_{l \in \mathcal{N}_k} m_{l \to k}(\mathbf{x}_k)$$

$$q_{k \to l}(\mathbf{x}_k) = f(\mathbf{x}_k) \prod_{l' \in \mathcal{N}_k \setminus \{l\}} m_{l' \to k}(\mathbf{x}_k)$$

$$m_{l \to k}(\mathbf{x}_k) = \int f(\mathbf{z}_{k,l} | \mathbf{x}_k, \mathbf{x}_l) \, q_{l \to k}(\mathbf{x}_l) \, d\mathbf{x}_l \, .$$

- Equivalently,

$$b(\mathbf{x}_k) \propto \int f(\mathbf{x}_k) \prod_{l \in \mathcal{N}_k} \left[f(\mathbf{z}_{k,l} | \mathbf{x}_k, \mathbf{x}_l) \, q_{l \to k}(\mathbf{x}_l) d\mathbf{x}_l \right]$$

$$q_{k \to l}(\mathbf{x}_k) = \int f(\mathbf{x}_k) \prod_{l' \in \mathcal{N}_k \setminus \{l\}} \left[f(\mathbf{z}_{k,l'} | \mathbf{x}_k, \mathbf{x}_{l'}) \, q_{l' \to k}(\mathbf{x}_{l'}) d\mathbf{x}_{l'} \right] .$$

Reformulation of BP rules

- Let $\mathcal{N}_k = \{l_1, l_2, \ldots, l_{|\mathcal{N}_k|}\}$, and consider the "composite" vectors

$$\bar{\mathbf{x}}_k \triangleq \left(\mathbf{x}_k^{\mathsf{T}}\ \mathbf{x}_{l_1}^{\mathsf{T}}\ \mathbf{x}_{l_2}^{\mathsf{T}} \cdots \mathbf{x}_{l_{|\mathcal{N}_k|}}^{\mathsf{T}}\right)^{\mathsf{T}} \qquad (\mathbf{x}_k \text{ and its neighbor states}),$$

$$\bar{\mathbf{z}}_k \triangleq \left(\mathbf{z}_{k,l_1}^{\mathsf{T}}\ \mathbf{z}_{k,l_2}^{\mathsf{T}} \cdots \mathbf{z}_{k,l_{|\mathcal{N}_k|}}^{\mathsf{T}}\right)^{\mathsf{T}} \qquad (\text{all observations involving } \mathbf{x}_k).$$

Reformulation of BP rules

- Let $\mathcal{N}_k = \{l_1, l_2, \ldots, l_{|\mathcal{N}_k|}\}$, and consider the "composite" vectors

$$\bar{\mathbf{x}}_k \triangleq \left(\mathbf{x}_k^{\mathrm{T}}\ \mathbf{x}_{l_1}^{\mathrm{T}}\ \mathbf{x}_{l_2}^{\mathrm{T}} \cdots \mathbf{x}_{l_{|\mathcal{N}_k|}}^{\mathrm{T}}\right)^{\mathrm{T}} \qquad (\mathbf{x}_k \text{ and its neighbor states}),$$

$$\bar{\mathbf{z}}_k \triangleq \left(\mathbf{z}_{k,l_1}^{\mathrm{T}}\ \mathbf{z}_{k,l_2}^{\mathrm{T}} \cdots \mathbf{z}_{k,l_{|\mathcal{N}_k|}}^{\mathrm{T}}\right)^{\mathrm{T}} \quad (\text{all observations involving } \mathbf{x}_k).$$

- We can now write

$$b(\mathbf{x}_k) \propto \int f(\mathbf{x}_k) \prod_{l \in \mathcal{N}_k} \left[f(\mathbf{z}_{k,l}|\mathbf{x}_k, \mathbf{x}_l)\, q_{l\to k}(\mathbf{x}_l)\, d\mathbf{x}_l \right]$$

$$\propto \int f(\bar{\mathbf{z}}_k|\bar{\mathbf{x}}_k)\, f(\bar{\mathbf{x}}_k)\, d\bar{\mathbf{x}}_k^{\sim k},$$

  with

$$f(\bar{\mathbf{x}}_k) \propto f(\mathbf{x}_k) \prod_{l \in \mathcal{N}_k} q_{l\to k}(\mathbf{x}_l) \quad (\text{composite prior}),$$

$$f(\bar{\mathbf{z}}_k|\bar{\mathbf{x}}_k) = \prod_{l \in \mathcal{N}_k} f(\mathbf{z}_{k,l}|\mathbf{x}_k, \mathbf{x}_l) \qquad (\text{composite likelihood}).$$

Reformulation of BP rules

- Let $\mathcal{N}_k = \{l_1, l_2, \ldots, l_{|\mathcal{N}_k|}\}$, and consider the "composite" vectors

$$\bar{\mathbf{x}}_k \triangleq \left(\mathbf{x}_k^{\mathsf{T}}\ \mathbf{x}_{l_1}^{\mathsf{T}}\ \mathbf{x}_{l_2}^{\mathsf{T}} \cdots \mathbf{x}_{l_{|\mathcal{N}_k|}}^{\mathsf{T}}\right)^{\mathsf{T}} \qquad (\mathbf{x}_k \text{ and its neighbor states}),$$

$$\bar{\mathbf{z}}_k \triangleq \left(\mathbf{z}_{k,l_1}^{\mathsf{T}}\ \mathbf{z}_{k,l_2}^{\mathsf{T}} \cdots \mathbf{z}_{k,l_{|\mathcal{N}_k|}}^{\mathsf{T}}\right)^{\mathsf{T}} \qquad (\text{all observations involving } \mathbf{x}_k).$$

- We can now write

$$b(\mathbf{x}_k) \propto \int f(\mathbf{x}_k) \prod_{l \in \mathcal{N}_k} \left[f(\mathbf{z}_{k,l}|\mathbf{x}_k, \mathbf{x}_l)\, q_{l\to k}(\mathbf{x}_l)\, d\mathbf{x}_l\right]$$
$$\propto \int f(\bar{\mathbf{z}}_k|\bar{\mathbf{x}}_k)\, f(\bar{\mathbf{x}}_k)\, d\bar{\mathbf{x}}_k^{\sim k},$$

with

$$f(\bar{\mathbf{x}}_k) \propto f(\mathbf{x}_k) \prod_{l \in \mathcal{N}_k} q_{l\to k}(\mathbf{x}_l) \quad (\text{composite prior}),$$

$$f(\bar{\mathbf{z}}_k|\bar{\mathbf{x}}_k) = \prod_{l \in \mathcal{N}_k} f(\mathbf{z}_{k,l}|\mathbf{x}_k, \mathbf{x}_l) \qquad (\text{composite likelihood}).$$

- Note that $f(\bar{\mathbf{z}}_k|\bar{\mathbf{x}}_k)$ corresponds to the composite observation model

$$\bar{\mathbf{z}}_k = \bar{\mathbf{y}}_k + \bar{\mathbf{v}}_k\,, \quad \text{with } \bar{\mathbf{y}}_k = H(\bar{\mathbf{x}}_k)\,,$$

where $H(\bar{\mathbf{x}}_k) \triangleq \left(\left(G(\mathbf{x}_k, \mathbf{x}_{l_1})\right)^{\mathsf{T}} \cdots \left(G(\mathbf{x}_k, \mathbf{x}_{l_{|\mathcal{N}_k|}})\right)^{\mathsf{T}}\right)^{\mathsf{T}}$ and $\bar{\mathbf{v}}_k \triangleq \left(\mathbf{v}_{k,l_1}^{\mathsf{T}} \cdots \mathbf{v}_{k,l_{|\mathcal{N}_k|}}^{\mathsf{T}}\right)^{\mathsf{T}}$.

Sigma point BP

- We can finally express $b(\mathbf{x}_k)$ as the result of a marginalization:

$$b(\mathbf{x}_k) = \int b(\bar{\mathbf{x}}_k)\, d\bar{\mathbf{x}}_k^{\sim k},$$

with the composite belief

$$b(\bar{\mathbf{x}}_k) \propto f(\bar{\mathbf{z}}_k|\bar{\mathbf{x}}_k)\, f(\bar{\mathbf{x}}_k).$$

Sigma point BP

- We can finally express $b(\mathbf{x}_k)$ as the result of a marginalization:

$$b(\mathbf{x}_k) = \int b(\bar{\mathbf{x}}_k)\, d\bar{\mathbf{x}}_k^{\sim k},$$

with the composite belief

$$b(\bar{\mathbf{x}}_k) \propto f(\bar{\mathbf{z}}_k|\bar{\mathbf{x}}_k) f(\bar{\mathbf{x}}_k).$$

- This expression of $b(\bar{\mathbf{x}}_k)$ has the same form as the expression $f(\mathbf{x}|\mathbf{z}) \propto f(\mathbf{z}|\mathbf{x})f(\mathbf{x})$ that occurred in Bayesian estimation $\Rightarrow$ SPs can again be used for calculating an approximate mean and covariance of $b(\bar{\mathbf{x}}_k)$ and, in turn, of $b(\mathbf{x}_k)$.

Sigma point BP

- We can finally express $b(\mathbf{x}_k)$ as the result of a marginalization:

$$b(\mathbf{x}_k) = \int b(\bar{\mathbf{x}}_k)\, d\bar{\mathbf{x}}_k^{\sim k},$$

with the composite belief

$$b(\bar{\mathbf{x}}_k) \propto f(\bar{\mathbf{z}}_k | \bar{\mathbf{x}}_k) f(\bar{\mathbf{x}}_k).$$

- This expression of $b(\bar{\mathbf{x}}_k)$ has the same form as the expression $f(\mathbf{x}|\mathbf{z}) \propto f(\mathbf{z}|\mathbf{x}) f(\mathbf{x})$ that occurred in Bayesian estimation $\Rightarrow$ SPs can again be used for calculating an approximate mean and covariance of $b(\bar{\mathbf{x}}_k)$ and, in turn, of $b(\mathbf{x}_k)$.

- More specifically, $\tilde{\boldsymbol{\mu}}_{b(\bar{\mathbf{x}}_k)}$ and $\tilde{\mathbf{C}}_{b(\bar{\mathbf{x}}_k)}$ can be obtained by using the closed-form expressions of the linear-Gaussian case, in which $\boldsymbol{\mu}_{\bar{\mathbf{y}}_k}$, $\mathbf{C}_{\bar{\mathbf{y}}_k}$, and $\mathbf{C}_{\bar{\mathbf{x}}_k \bar{\mathbf{y}}_k}$ are replaced by the SP approximations $\tilde{\boldsymbol{\mu}}_{\bar{\mathbf{y}}_k}$, $\tilde{\mathbf{C}}_{\bar{\mathbf{y}}_k}$, and $\tilde{\mathbf{C}}_{\bar{\mathbf{x}}_k \bar{\mathbf{y}}_k}$.

Sigma point BP

SPBP algorithm

An SP-based approximate calculation of the mean $\boldsymbol{\mu}_{b(\mathbf{x}_k)}$ and covariance matrix $\mathbf{C}_{b(\mathbf{x}_k)}$ of $b(\mathbf{x}_k)$ can be obtained by performing the following two steps:

1. Use the "Bayesian estimation" SP scheme to calculate $\tilde{\boldsymbol{\mu}}_{b(\bar{\mathbf{x}}_k)} \approx \boldsymbol{\mu}_{b(\bar{\mathbf{x}}_k)}$ and $\tilde{\mathbf{C}}_{b(\bar{\mathbf{x}}_k)} \approx \mathbf{C}_{b(\bar{\mathbf{x}}_k)}$ representing $b(\bar{\mathbf{x}}_k)$ from $\boldsymbol{\mu}_{\bar{\mathbf{x}}_k}$ and $\mathbf{C}_{\bar{\mathbf{x}}_k}$ representing $f(\bar{\mathbf{x}}_k)$.

Sigma point BP

SPBP algorithm

An SP-based approximate calculation of the mean $\boldsymbol{\mu}_{b(\mathbf{x}_k)}$ and covariance matrix $\mathbf{C}_{b(\mathbf{x}_k)}$ of $b(\mathbf{x}_k)$ can be obtained by performing the following two steps:

1. Use the "Bayesian estimation" SP scheme to calculate $\tilde{\boldsymbol{\mu}}_{b(\bar{\mathbf{x}}_k)} \approx \boldsymbol{\mu}_{b(\bar{\mathbf{x}}_k)}$ and $\tilde{\mathbf{C}}_{b(\bar{\mathbf{x}}_k)} \approx \mathbf{C}_{b(\bar{\mathbf{x}}_k)}$ representing $b(\bar{\mathbf{x}}_k)$ from $\boldsymbol{\mu}_{\bar{\mathbf{x}}_k}$ and $\mathbf{C}_{\bar{\mathbf{x}}_k}$ representing $f(\bar{\mathbf{x}}_k)$.
2. Obtain $\tilde{\boldsymbol{\mu}}_{b(\mathbf{x}_k)}$ and $\tilde{\mathbf{C}}_{b(\mathbf{x}_k)}$ by extracting from $\tilde{\boldsymbol{\mu}}_{b(\bar{\mathbf{x}}_k)}$ and $\tilde{\mathbf{C}}_{b(\bar{\mathbf{x}}_k)}$ the elements related to $\mathbf{x}_k$. This corresponds to the marginalization

$$b(\mathbf{x}_k) = \int b(\bar{\mathbf{x}}_k)\, d\bar{\mathbf{x}}_k^{\sim k}.$$

Outline

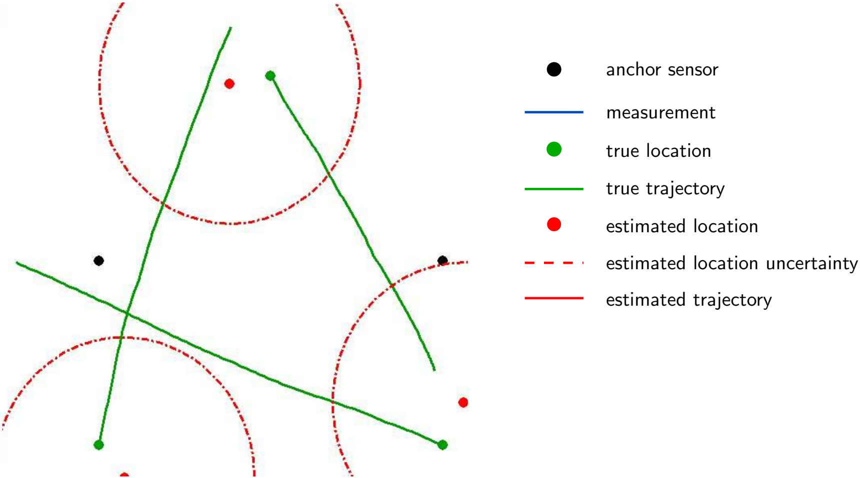
Cooperative self-localization

- We simulated a decentralized, cooperative, dynamic localization scenario [Wymeersch et al., 2009] using a network of three mobile sensors and two anchor sensors.
- The state $\mathbf{x}_{k,i}$ of mobile sensor $k \in \{1, 2, 3\}$ at time $i \in \{0, 1, \ldots, 50\}$ consists of the sensor's location and velocity and corresponds to a variable node in the factor graph.

Cooperative self-localization

- We simulated a decentralized, cooperative, dynamic localization scenario [Wymeersch et al., 2009] using a network of three mobile sensors and two anchor sensors.
- The state $\mathbf{x}_{k,i}$ of mobile sensor $k \in \{1, 2, 3\}$ at time $i \in \{0, 1, \ldots, 50\}$ consists of the sensor's location and velocity and corresponds to a variable node in the factor graph.
- Each mobile sensor communicates with all other sensors, performs distance measurements relative to all other sensors, and estimates its own state.
- The communication graph coincides with the factor graph; this facilitates a distributed processing.

Simulation scenario



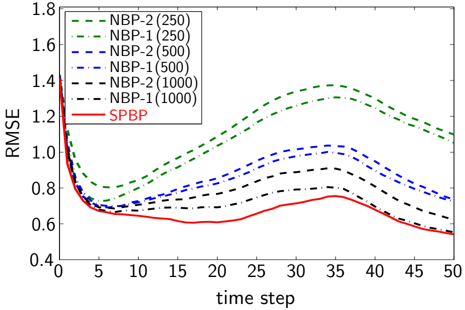
Simulation results

- We compare the proposed SPBP algorithm with two nonparametric (random particle based) BP methods, referred to as NBP-1 [Ihler et al., 2005] and NBP-2 [Savic et al., 2013]; both simulated with 250, 500 and 1000 particles.

Simulation results

- We compare the proposed SPBP algorithm with two nonparametric (random particle based) BP methods, referred to as NBP-1 [Ihler et al., 2005] and NBP-2 [Savic et al., 2013]; both simulated with 250, 500 and 1000 particles.

Root-mean-square error (RMSE) of location and velocity versus time:



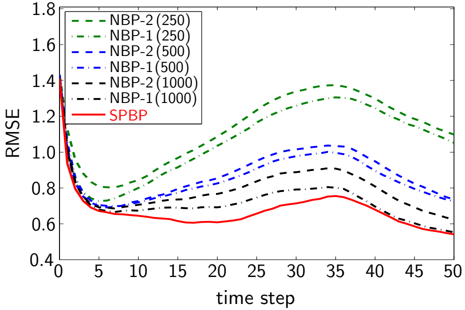
Communication and computation requirements:

	Communication between sensors [real values]	Runtime on Xeon X5650 [seconds]
SPBP	500	0.61
NBP-1 (250)	50000	1.53
NBP-2 (250)	50000	2.01
NBP-1 (500)	100000	5.16
NBP-2 (500)	100000	7.27
NBP-1 (1000)	200000	19.57
NBP-2 (1000)	200000	28.10

Simulation results

- We compare the proposed SPBP algorithm with two nonparametric (random particle based) BP methods, referred to as NBP-1 [Ihler et al., 2005] and NBP-2 [Savic et al., 2013]; both simulated with 250, 500 and 1000 particles.

Root-mean-square error (RMSE) of location and velocity versus time:



Communication and computation requirements:

	Communication between sensors [real values]	Runtime on Xeon X5650 [seconds]
SPBP	500	0.61
NBP-1 (250)	50000	1.53
NBP-2 (250)	50000	2.01
NBP-1 (500)	100000	5.16
NBP-2 (500)	100000	7.27
NBP-1 (1000)	200000	19.57
NBP-2 (1000)	200000	28.10

- The proposed SPBP algorithm outperforms NBP-1 and NBP-2 although it requires significantly less communications and computations.

Conclusion

- The proposed sigma point belief propagation (SPBP) algorithm extends the SP filter to Bayesian inference corresponding to a general factorization structure.

Conclusion

- The proposed sigma point belief propagation (SPBP) algorithm extends the SP filter to Bayesian inference corresponding to a general factorization structure.
- SPBP is well suited to certain decentralized inference problems in wireless sensor networks because of its low communication requirements.

Conclusion

- The proposed sigma point belief propagation (SPBP) algorithm extends the SP filter to Bayesian inference corresponding to a general factorization structure.
- SPBP is well suited to certain decentralized inference problems in wireless sensor networks because of its low communication requirements.
- In a cooperative self-localization scenario, SPBP can outperform nonparametric BP although it requires significantly less communications and computations.

Thank you!

References

[Kalman, 1960] R. E. Kalman, "A new approach to linear filtering and prediction problems," *J. Basic Eng.*, 1960

[Gordon et al., 1993] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *Proc. IEE Radar Signal Processing*, 1993

[Julier et al., 1997] S. J. Julier and J. K. Uhlmann, "A new extension of the Kalman filter to nonlinear systems," in *Proc. AeroSense-97*, 1997

[Weiss et al., 2001] Y. Weiss and W. T. Freeman, "Correctness of belief propagation in Gaussian graphical models of arbitrary topology," *Neural Computation*, 2001

[Sudderth et al., 2003] E. B. Sudderth, A. T. Ihler, W. T. Freeman, and A. S. Willsky, "Nonparametric belief propagation," in *Proc. CVPR*, 2003

[Ihler et al., 2005] A. T. Ihler, J. W. Fisher, R. L. Moses, and A. S. Willsky, "Nonparametric belief propagation for self-localization of sensor networks," *IEEE J. Sel. Areas Comm.*, 2005

[Wymeersch et al., 2009] H. Wymeersch, J. Lien, and M. Z. Win, "Cooperative localization in wireless networks," *Proc. IEEE*, 2009

[Savic et al., 2013] V. Savic and S. Zazo, "Cooperative localization in mobile networks using nonparametric variants of belief propagation," *Ad Hoc Networks*, 2013