Sigma Point Belief Propagation

Florian Meyer

joint work with Ondrej Hlinka and Franz Hlawatsch

Institute of Telecommunications
Vienna University of Technology
Outline

1. Sigma Point Basics
2. Bayesian Estimation with Sigma Points
3. Belief Propagation
4. Sigma Point Belief Propagation
5. Simulation Results
Outline

1. Sigma Point Basics
2. Bayesian Estimation with Sigma Points
3. Belief Propagation
4. Sigma Point Belief Propagation
5. Simulation Results
Consider a random vector $\mathbf{x} \in \mathbb{R}^J$ whose mean $\mu_x$ and covariance matrix $\mathbf{C}_x$ are known, and a transformed random vector $\mathbf{y} = H(\mathbf{x})$. 
Sigma point basics

Consider a random vector $x \in \mathbb{R}^J$ whose mean $\mu_x$ and covariance matrix $C_x$ are known, and a transformed random vector $y = H(x)$.

Sigma points (SPs) [Julier et al., 1997] are used to calculate approximations of the mean $\mu_y$, covariance matrix $C_y$, and cross-covariance matrix $C_{x,y}$. 

Consider a random vector $\mathbf{x} \in \mathbb{R}^J$ whose mean $\mu_x$ and covariance matrix $C_x$ 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 $\mu_y$, covariance matrix $C_y$, and cross-covariance matrix $C_{x,y}$.

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]

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

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

Here, \( \left( \mathbf{C}_x^{1/2} \right)_j \) is the $j$th row or column of the matrix square root of $\mathbf{C}_x$ and $\kappa$ is a parameter controlling the spread of the SPs around the mean $\mu_x$. 
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]

\[
\begin{align*}
\mathbf{x}(j) &= \begin{cases}
\mu_x, & j = 0 \\
\mu_x + \sqrt{J + \kappa} \left( C_x^{1/2} \right)_j, & j = 1, \ldots, J \\
\mu_x - \sqrt{J + \kappa} \left( C_x^{1/2} \right)_j, & j = J + 1, \ldots, 2J 
\end{cases} \\
w(j) &= \begin{cases}
\frac{\kappa}{J + \kappa}, & j = 0 \\
\frac{1}{2(J + \kappa)}, & j = 1, \ldots, 2J.
\end{cases}
\end{align*}
\]

Here, $\left( C_x^{1/2} \right)_j$ is the $j$th row or column of the matrix square root of $C_x$ and $\kappa$ is a parameter controlling the spread of the SPs around the mean $\mu_x$.

- SPs have the property that the weighted sample mean $\tilde{\mu}_x = \sum_{j=0}^{2J} w(j) \mathbf{x}(j)$ and weighted sample covariance matrix $\tilde{C}_x = \sum_{j=0}^{2J} w(j) (\mathbf{x}(j) - \tilde{\mu}_x)(\mathbf{x}(j) - \tilde{\mu}_x)^T$ are exactly equal to $\mu_x$ and $C_x$, respectively.
SPs \( \{y^{(j)}\}_{j=0}^{2J} \) of \( y \) can be obtained by propagating each SP \( x^{(j)} \) through \( H(\cdot) \) [Julier et al., 1997]:

\[
y^{(j)} = H(x^{(j)}), \quad j = 0, \ldots, 2J.
\]
Calculation of mean and covariance

From \( \{(x(j), y(j), w(j))\}_{j=0}^{2J} \), one can calculate approximations of \( \mu_y \), \( C_y \) and \( C_{xy} \) as

\[
\tilde{\mu}_y = \sum_{j=0}^{2J} w(j) y(j),
\]

\[
\tilde{C}_y = \sum_{j=0}^{2J} w(j) (y(j) - \mu_y)(y(j) - \mu_y)^T,
\]

\[
\tilde{C}_{xy} = \sum_{j=0}^{2J} w(j) (y(j) - \mu_y)(x(j) - \mu_x)^T.
\]
1. Sigma Point Basics

2. Bayesian Estimation with Sigma Points

3. Belief Propagation

4. Sigma Point Belief Propagation

5. Simulation Results
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 $\mathbf{z} = \mathbf{y} + \mathbf{n}$, 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 $C_n$.

- 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.
SPs can be used for Bayesian estimation of a random vector $\mathbf{x}$ from an observed vector $\mathbf{z} = \mathbf{y} + \mathbf{n}$, 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.

Closed-form calculation of $f(\mathbf{x}|\mathbf{z})$ is usually infeasible!
A feasible special case is when $H(\cdot)$ is linear, i.e., $H(x) = Hx$ and $x$ and $n$ are Gaussian. Then $f(x|z)$ is also Gaussian, and $\mu_{x|z}$ and $C_{x|z}$ can be calculated as

$$
\mu_{x|z} = \mu_x + K(z - \mu_y), \quad C_{x|z} = C_x - K(C_y + C_n)K^T,
$$

with

$$
K = C_{xy}(C_y + C_n)^{-1},
\mu_y = H\mu_x,
C_y = HC_xH^T,
C_{xy} = C_xH^T.
$$
In the **nonlinear case**, $\mu_{x|z}$ and $C_{x|z}$ 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 $\mu_y$, $C_y$, and $C_{xy}$ are replaced by the SP approximations $\tilde{\mu}_y$, $\tilde{C}_y$, and $\tilde{C}_{xy}$. 
Bayesian estimation with sigma points

- In the **nonlinear case**, $\mu_{x|z}$ and $C_{x|z}$ 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 $\mu_y$, $C_y$, and $C_{xy}$ are replaced by the SP approximations $\tilde{\mu}_y$, $\tilde{C}_y$, and $\tilde{C}_{xy}$.

- **SP approximations** $\tilde{\mu}_{x|z}$ and $\tilde{C}_{x|z}$ are thus obtained as

$$
\tilde{\mu}_{x|z} = \mu_x + \tilde{K}(z - \tilde{\mu}_y), \quad \tilde{C}_{x|z} = C_x - \tilde{K}(\tilde{C}_y + C_n)\tilde{K}^T,
$$

with $\tilde{K} = \tilde{C}_{xy}(\tilde{C}_y + C_n)^{-1}$. 
Bayesian estimation with sigma points

- In the **nonlinear case**, $\mu_{x|z}$ and $C_{x|z}$ 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 $\mu_y$, $C_y$, and $C_{xy}$ are replaced by the SP approximations $\tilde{\mu}_y$, $\tilde{C}_y$, and $\tilde{C}_{xy}$.

- SP approximations $\tilde{\mu}_{x|z}$ and $\tilde{C}_{x|z}$ are thus obtained as

$$
\tilde{\mu}_{x|z} = \mu_x + \tilde{K}(z - \tilde{\mu}_y), \quad \tilde{C}_{x|z} = C_x - \tilde{K}(\tilde{C}_y + C_n)\tilde{K}^T,
$$

with $\tilde{K} = \tilde{C}_{xy}(\tilde{C}_y + C_n)^{-1}$.

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

1. Sigma Point Basics
2. Bayesian Estimation with Sigma Points
3. Belief Propagation
4. Sigma Point Belief Propagation
5. Simulation Results
Consider $K$ state vectors $x_k, k \in \{1, \ldots, K\}$
Consider $K$ state vectors $x_k$, $k \in \{1, \ldots, K\}$

Assume that the posterior pdf $f(x|z)$ factorizes as

$$f(x|z) \propto \left[ \prod_{k=1}^{K} f(x_k) \right] \prod_{(k',l) \in E} f(z_{k',l}|x_{k'},x_l).$$

Here, $z_{k,l} = H(x_k,x_l) + v_{k,l}$ are noisy “pairwise” observations.
Consider $K$ state vectors $x_k$, $k \in \{1, \ldots, K\}$

Assume that the posterior pdf $f(x|z)$ factorizes as

$$f(x|z) \propto \left[ \prod_{k=1}^{K} f(x_k) \right] \prod_{(k',l) \in E} f(z_{k',l}|x_{k'},x_l).$$

Here, $z_{k,l} = H(x_k, x_l) + v_{k,l}$ are noisy “pairwise” observations.

Representation by factor graph:

- Variable nodes $x_k$ for $k \in \{1, 2, 3, 4\}$
- Factor nodes $f_k, f_{k,l}$
- Edge set $E = \{(1,3), (1,4), (2,3), (3,4)\}$
- $f_k \triangleq f(x_k)$
- $f_{k,l} \triangleq f(z_{k,l}|x_k, x_l)$
To obtain an estimate of $x_k$, the "marginal" posterior pdf
\[ f(x_k|z) = \int f(x|z) dx^{\sim k} \]
is needed.

Direct calculation of $f(x_k|z) = \int f(x|z) dx^{\sim k}$ is typically infeasible.
To obtain an estimate of $x_k$, the “marginal” posterior pdf $f(x_k|z) = \int f(x|z) dx^{\sim k}$ is needed.

Direct calculation of $f(x_k|z) = \int f(x|z) dx^{\sim k}$ is typically infeasible.

An approximate marginal posterior (‘‘belief’’) $b(x_k) \approx f(x_k|z)$ is obtained by running belief propagation (BP) message passing on the factor graph representing the joint posterior $f(x|z)$.

In the special case where the factor graph is a tree (no loops/cycles), BP is exact, i.e., $b(x_k) = f(x_k|z)$. 
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 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 $x_k$:

  $$b(x_k) \propto f(x_k) \prod_{l \in \mathcal{N}_k} m_{l \rightarrow k}(x_k).$$
Belief propagation rules

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

- **Belief** of variable $x_k$:
  
  $$b(x_k) \propto f(x_k) \prod_{l \in \mathcal{N}_k} m_{l \to k}(x_k).$$

- **Message** from variable node $x_k$ to function node $f(z_k,l|x_k,x_l)$:
  
  $$q_{k \to l}(x_k) = f(x_k) \prod_{l' \in \mathcal{N}_k \setminus \{l\}} m_{l' \to k}(x_k).$$
Belief propagation rules

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

- **Belief** of variable $x_k$:
  \[
  b(x_k) \propto f(x_k) \prod_{l \in \mathcal{N}_k} m_{l \rightarrow k}(x_k).
  \]

- **Message** from variable node $x_k$ to function node $f(z_{k,l}|x_k, x_l)$:
  \[
  q_{k \rightarrow l}(x_k) = f(x_k) \prod_{l' \in \mathcal{N}_k \setminus \{l\}} m_{l' \rightarrow k}(x_k).
  \]

- **Message** from function node $f(z_{k,l}|x_k, x_l)$ to variable node $x_k$:
  \[
  m_{l \rightarrow k}(x_k) = \int f(z_{k,l}|x_k, x_l) q_{l \rightarrow k}(x_l) \, dx_l.
  \]
Outline

1. Sigma Point Basics
2. Bayesian Estimation with Sigma Points
3. Belief Propagation
4. Sigma Point Belief Propagation
5. Simulation Results
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.

<table>
<thead>
<tr>
<th>Factorization Structure</th>
<th>Sequential</th>
<th>General</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nonlinear, non-Gaussian</td>
<td>Particle filter [Gordon et al., 1993]</td>
<td>Nonparametric BP [Sudderth et al., 2003]</td>
</tr>
<tr>
<td>Nonlinear, Gaussian</td>
<td>SP filter [Julier et al., 1997]</td>
<td>SPBP</td>
</tr>
</tbody>
</table>
Reformulation of BP rules

- Recall BP rules:

\[ b(x_k) \propto f(x_k) \prod_{l \in \mathcal{N}_k} m_{l \rightarrow k}(x_k) \]

\[ q_{k \rightarrow l}(x_k) = f(x_k) \prod_{l' \in \mathcal{N}_k \backslash \{l\}} m_{l' \rightarrow k}(x_k) \]

\[ m_{l \rightarrow k}(x_k) = \int f(z_{k,l} | x_k, x_l) q_{l \rightarrow k}(x_l) \, dx_l. \]
Reformulation of BP rules

- Recall BP rules:

\[ b(x_k) \propto f(x_k) \prod_{l \in \mathcal{N}_k} m_{l \to k}(x_k) \]

\[ q_{k \to l}(x_k) = f(x_k) \prod_{l' \in \mathcal{N}_k \setminus \{l\}} m_{l' \to k}(x_k) \]

\[ m_{l \to k}(x_k) = \int f(z_{k,l} | x_k, x_l) q_{l \to k}(x_l) \, dx_l. \]

- Equivalently,

\[ b(x_k) \propto \int f(x_k) \prod_{l \in \mathcal{N}_k} \left[ f(z_{k,l} | x_k, x_l) q_{l \to k}(x_l) \, dx_l \right] \]

\[ q_{k \to l}(x_k) = \int f(x_k) \prod_{l' \in \mathcal{N}_k \setminus \{l\}} \left[ f(z_{k,l'} | x_k, x_{l'}) q_{l' \to k}(x_{l'}) \, dx_{l'} \right]. \]
Reformulation of BP rules

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

$$\bar{\mathbf{x}}_k \triangleq (\mathbf{x}_k^\top \mathbf{x}_{l_1}^\top \mathbf{x}_{l_2}^\top \cdots \mathbf{x}_{l_{|\mathcal{N}_k|}}^\top)^\top$$  

($\mathbf{x}_k$ and its neighbor states),

$$\bar{\mathbf{z}}_k \triangleq (\mathbf{z}_{k,l_1}^\top \mathbf{z}_{k,l_2}^\top \cdots \mathbf{z}_{k,l_{|\mathcal{N}_k|}}^\top)^\top$$  

(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

$$\bar{x}_k \triangleq (x_k^T x_{l_1}^T x_{l_2}^T \cdots x_{l_{|\mathcal{N}_k|}}^T)^T \quad (x_k \text{ and its neighbor states}),$$

$$\bar{z}_k \triangleq (z_{k,l_1}^T z_{k,l_2}^T \cdots z_{k,l_{|\mathcal{N}_k|}}^T)^T \quad (\text{all observations involving } x_k).$$

We can now write

$$b(x_k) \propto \int f(x_k) \prod_{l \in \mathcal{N}_k} \left[ f(z_{k,l}|x_k, x_l) q_{l \rightarrow k}(x_l) \, dx_l \right]$$

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

with

$$f(\bar{x}_k) \propto f(x_k) \prod_{l \in \mathcal{N}_k} q_{l \rightarrow k}(x_l) \quad (\text{composite prior}),$$

$$f(\bar{z}_k|\bar{x}_k) = \prod_{l \in \mathcal{N}_k} f(z_{k,l}|x_k, x_l) \quad (\text{composite likelihood}).$$
Reformulation of BP rules

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

\[ \bar{x}_k \triangleq (x^T_k x^T_{l_1} x^T_{l_2} \cdots x^T_{l_{|\mathcal{N}_k|}})^T \quad \text{(}x_k\text{ and its neighbor states)}, \]

\[ \bar{z}_k \triangleq (z^T_{k,l_1} z^T_{k,l_2} \cdots z^T_{k,l_{|\mathcal{N}_k|}})^T \quad \text{(all observations involving } x_k). \]

We can now write

\[
    b(x_k) \propto \int f(x_k) \prod_{l \in \mathcal{N}_k} \left[ f(z_{k,l}|x_k, x_l) q_{l \rightarrow k}(x_l) \, dx_l \right] \\
    \propto \int f(\bar{z}_k|\bar{x}_k) f(\bar{x}_k) \, d\bar{x}^\sim_k,
\]

with

\[
    f(\bar{x}_k) \propto f(x_k) \prod_{l \in \mathcal{N}_k} q_{l \rightarrow k}(x_l) \quad \text{(composite prior)},
\]

\[
    f(\bar{z}_k|\bar{x}_k) = \prod_{l \in \mathcal{N}_k} f(z_{k,l}|x_k, x_l) \quad \text{(composite likelihood)}.\]

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

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

where $H(\bar{x}_k) \triangleq \left( (G(x_k, x_{l_1}))^T \cdots (G(x_k, x_{l_{|\mathcal{N}_k|}}))^T \right)^T$ and $\bar{v}_k \triangleq (v^T_{k,l_1} \cdots v^T_{k,l_{|\mathcal{N}_k|}})^T$. 
We can finally express $b(x_k)$ as the result of a marginalization:

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

with the composite belief

$$b(\bar{x}_k) \propto f(\bar{z}_k|\bar{x}_k) f(\bar{x}_k).$$
We can finally express $b(x_k)$ as the result of a marginalization:

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

with the composite belief

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

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

\[
b(x_k) = \int b(\bar{x}_k) d\bar{x}_k^k, \]

with the composite belief

\[
b(\bar{x}_k) \propto f(\bar{z}_k|\bar{x}_k) f(\bar{x}_k). \]

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

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

SPBP algorithm

An SP-based approximate calculation of the mean $\mu_{b(x_k)}$ and covariance matrix $C_{b(x_k)}$ of $b(x_k)$ can be obtained by performing the following two steps:

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

An SP-based approximate calculation of the mean $\mu_{b(x_k)}$ and covariance matrix $C_{b(x_k)}$ of $b(x_k)$ can be obtained by performing the following two steps:

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

2. Obtain $\tilde{\mu}_{b(x_k)}$ and $\tilde{C}_{b(x_k)}$ by extracting from $\tilde{\mu}_{b(\bar{x}_k)}$ and $\tilde{C}_{b(\bar{x}_k)}$ the elements related to $x_k$. This corresponds to the marginalization

$$b(x_k) = \int b(\bar{x}_k) d\bar{x}_k^k.$$
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 $x_{k,i}$ 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

- anchor sensor
- measurement
- true location
- true trajectory
- estimated location
- estimated location uncertainty
- estimated trajectory
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.
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:**

<table>
<thead>
<tr>
<th>Time step</th>
<th>RMSE 0.4</th>
<th>RMSE 0.6</th>
<th>RMSE 0.8</th>
<th>RMSE 1.0</th>
<th>RMSE 1.2</th>
<th>RMSE 1.4</th>
<th>RMSE 1.6</th>
<th>RMSE 1.8</th>
</tr>
</thead>
<tbody>
<tr>
<td>NBP-2 (250)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NBP-1 (250)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NBP-2 (500)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NBP-1 (500)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NBP-2 (1000)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NBP-1 (1000)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SPBP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Communication and computation requirements:**

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Communication between sensors</th>
<th>Runtime on Xeon X5650</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPBP</td>
<td>500</td>
<td>0.61</td>
</tr>
<tr>
<td>NBP-1 (250)</td>
<td>50000</td>
<td>1.53</td>
</tr>
<tr>
<td>NBP-2 (250)</td>
<td>50000</td>
<td>2.01</td>
</tr>
<tr>
<td>NBP-1 (500)</td>
<td>100000</td>
<td>5.16</td>
</tr>
<tr>
<td>NBP-2 (500)</td>
<td>100000</td>
<td>7.27</td>
</tr>
<tr>
<td>NBP-1 (1000)</td>
<td>200000</td>
<td>19.57</td>
</tr>
<tr>
<td>NBP-2 (1000)</td>
<td>200000</td>
<td>28.10</td>
</tr>
</tbody>
</table>
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:

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

Communication and computation requirements:

<table>
<thead>
<tr>
<th></th>
<th>Communication between sensors [real values]</th>
<th>Runtime on Xeon X5650 [seconds]</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPBP</td>
<td>500</td>
<td>0.61</td>
</tr>
<tr>
<td>NBP-1 (250)</td>
<td>50000</td>
<td>1.53</td>
</tr>
<tr>
<td>NBP-2 (250)</td>
<td>50000</td>
<td>2.01</td>
</tr>
<tr>
<td>NBP-1 (500)</td>
<td>100000</td>
<td>5.16</td>
</tr>
<tr>
<td>NBP-2 (500)</td>
<td>100000</td>
<td>7.27</td>
</tr>
<tr>
<td>NBP-1 (1000)</td>
<td>200000</td>
<td>19.57</td>
</tr>
<tr>
<td>NBP-2 (1000)</td>
<td>200000</td>
<td>28.10</td>
</tr>
</tbody>
</table>
The proposed sigma point belief propagation (SPBP) algorithm extends the SP filter to Bayesian inference corresponding to a general factorization structure.
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.
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!


