Aalborg Universitet

AALBORG UNIVERSITY

Cooperative Localization for Mobile Networks

A Distributed Belief Propagation – Mean Field Message Passing Algorithm

Cakmak, Burak; Urup, Daniel Nygaard; Meyer, Florian; Pedersen, Troels; Fleury, Bernard Henri; Hlawatsch, Franz

[Link to publication from Aalborg University](#)

Cooperative Localization for Mobile Networks: A Distributed Belief Propagation – Mean Field Message Passing Algorithm

Burak Çakmak, Daniel N. Urup, Florian Meyer, *Member, IEEE*, Troels Pedersen, *Member, IEEE*, Bernard H. Fleury, *Senior Member, IEEE*, and Franz Hlawatsch, *Fellow, IEEE*

Abstract—We propose a hybrid message passing method for distributed cooperative localization and tracking of mobile agents. Belief propagation and mean field message passing are employed for, respectively, the motion-related and measurement-related part of the factor graph. Using a Gaussian belief approximation, only three real values per message passing iteration have to be broadcast to neighboring agents. Despite these very low communication requirements, the estimation accuracy can be comparable to that of particle-based belief propagation.

Index Terms—Belief propagation, mean field approximation, cooperative localization, distributed estimation, information projection, Kullback-Leibler-divergence, mobile agent network.

I. INTRODUCTION

Cooperative localization is a powerful approach for mobile networks [1]–[5]. An attractive methodology for cooperative localization is sequential Bayesian estimation via message passing algorithms [6]. In particular, distributed belief propagation (BP) message passing algorithms were proposed in [2], [3], [7]–[11] to localize static or mobile agents. Feasible implementations involve certain approximations and use, e.g., particle methods [2], [3], [8]–[10] or the sigma point technique [11]. Each message transmitted between neighboring agents is a set of hundreds or more particles in the former case [2], [3], [8] and a mean and a covariance matrix, i.e., five real numbers in 2-D localization, in the latter case. For static agents, also message passing algorithms based on expectation propagation [12], [13] or the mean field (MF) approximation [14] were proposed. Similarly to sigma point BP [11], they use a Gaussian approximation and the transmitted messages consist of a mean and a covariance matrix.

In this letter, building on the theoretical framework in [15], we present a distributed hybrid BP–MF message passing method for cooperative localization and tracking of mobile agents. We employ BP and MF [15] for, respectively, the motion-related and measurement-related part of the underlying factor graph, and we use a Gaussian belief approximation. Each BP–MF iteration includes an information projection

[16] that is efficiently implemented by means of a Newton conjugate-gradient technique [17]. Our method can achieve an accuracy comparable to that of BP-based methods with the same communication cost as the MF method [14], i.e., three real numbers per transmitted message in 2-D localization.

This letter is organized as follows. The system model is described in Section II. The hybrid BP–MF scheme is developed in Section III, and the Gaussian belief approximation in Section IV. Section V presents simulation results.

II. SYSTEM MODEL

The mobile network at discrete time $n \in \{1, ..., N\}$ is described by a set of network nodes $\mathcal{V}^n$ and a set of edges $\mathcal{E}^n$ representing the communication/measurement links between the nodes. The set $\mathcal{V}^n$ is partitioned into a set $\mathcal{V}_\mathrm{M}^n$ of mobile agents at unknown positions and a set $\mathcal{V}_\mathrm{A}^n$ of static anchors at known positions. An edge $(k, l) \in \mathcal{E}^n$ indicates the fact that agent or anchor l transmits data to agent k and, concurrently, agent k acquires a noisy measurement of its distance to agent or anchor l. The edge set $\mathcal{E}^n$ is partitioned into a set $\mathcal{E}_\mathrm{M}^n$ of edges between certain agents, i.e., $(k, l) \in \mathcal{E}_\mathrm{M}^n$ implies $k, l \in \mathcal{V}_\mathrm{M}^n$, and a set $\mathcal{E}_\mathrm{MA}^n$ of edges between certain agents and anchors, i.e., $(k, l) \in \mathcal{E}_\mathrm{MA}^n$ implies $k \in \mathcal{V}_\mathrm{M}^n$ and $l \in \mathcal{V}_\mathrm{A}^n$. Information exchange between agents is bidirectional, i.e., $(k, l) \in \mathcal{E}_\mathrm{M}^n$ implies $(l, k) \in \mathcal{E}_\mathrm{M}^n$. We consider a distributed scenario where each agent knows only its own measurements. Since the anchors have exact knowledge of their own position, they do not need to acquire measurements and receive position information from neighboring nodes. Accordingly, anchors transmit position information to agents but not vice versa, i.e., $(k, l) \in \mathcal{E}_\mathrm{MA}^n$ implies $(l, k) \notin \mathcal{E}_\mathrm{MA}^n$.

Let the vector $\boldsymbol{x}_k^n$ denote the state of agent $k \in \mathcal{V}_\mathrm{M}^n$ at time $n \in \{1, ..., N\}$. Moreover, let $\boldsymbol{x}^n \triangleq \left[\boldsymbol{x}_k^n\right]_{k \in \mathcal{V}_\mathrm{M}^n}$ and $\boldsymbol{x}^{1:n} \triangleq \left[\boldsymbol{x}^i\right]_{i=1}^n$. While our approach applies to any linear-Gaussian motion model, we here consider specifically those two motion models (MMs) that are most frequently used in practice. In MM1, $\boldsymbol{x}_k^n = \boldsymbol{p}_k^n \in \mathbb{R}^2$ is the 2-D position of agent k at time n. If agent k belongs to the network at times n and $n-1$, i.e. $k \in \mathcal{V}_\mathrm{M}^n \cap \mathcal{V}_\mathrm{M}^{n-1}$, then $\boldsymbol{p}_k^n$ is assumed to evolve according to the Gaussian random walk model [18]

$$\boldsymbol{p}_k^n = \boldsymbol{p}_k^{n-1} + \sqrt{T}\boldsymbol{v}_k^n.$$

Here, T is the duration of one time step and $\boldsymbol{v}_k^n \in \mathbb{R}^2$ is zero-mean Gaussian driving noise with component variance σ_v^2.

Note that $\boldsymbol{v}_k^n$ can be interpreted as a random velocity. In MM2, $\boldsymbol{x}_k^n = \left[(\boldsymbol{p}_k^n)^{\mathrm{T}} \ (\boldsymbol{v}_k^n)^{\mathrm{T}}\right]^{\mathrm{T}}$, where $\boldsymbol{v}_k^n \in \mathbb{R}^2$ is the 2-D velocity of agent k at time n. For $k \in \mathcal{V}_{\mathrm{M}}^n \cap \mathcal{V}_{\mathrm{M}}^{n-1}$, $\boldsymbol{x}_k^n$ is assumed to evolve according to the constant velocity model [18]

$$\boldsymbol{x}_k^n = \boldsymbol{F}\boldsymbol{x}_k^{n-1} + \boldsymbol{G}\boldsymbol{a}_k^n. \tag{1}$$

Here, $\boldsymbol{a}_k^n \in \mathbb{R}^2$ is zero-mean Gaussian driving noise (a random acceleration) with component variance σ_a^2. Moreover, $\boldsymbol{F} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \otimes \mathbf{I}_2$ and $\boldsymbol{G} = \begin{bmatrix} T^2/2 \\ T \end{bmatrix} \otimes \mathbf{I}_2$, where $\otimes$ denotes the Kronecker product and $\mathbf{I}_m$ is the $m \times m$ identity matrix. Note that in both MM1 and MM2, the state-transition probability density function (pdf) $p(\boldsymbol{x}_k^n|\boldsymbol{x}_k^{n-1})$ is Gaussian. For agents that are part of the network at time n but not at time $n-1$, i.e., $k \in \mathcal{V}_{\mathrm{M}}^n \setminus \mathcal{V}_{\mathrm{M}}^{n-1}$, we set $p(\boldsymbol{x}_k^n|\boldsymbol{x}_k^{n-1}) = p(\boldsymbol{x}_k^n)$, where the prior pdf $p(\boldsymbol{x}_k^n)$ is Gaussian. Under common statistical independence assumptions on $\boldsymbol{v}_k^n$ or $\boldsymbol{a}_k^n$ [3], the joint prior pdf of all agent states up to time n is given by

$$p(\boldsymbol{x}^{1:n}) = \prod_{i=1}^n \prod_{k \in \mathcal{V}_{\mathrm{M}}^i} p(\boldsymbol{x}_k^i|\boldsymbol{x}_k^{i-1}). \tag{2}$$

If $(k,l) \in \mathcal{E}^n$, agent $k \in \mathcal{V}_{\mathrm{M}}^n$ acquires at time n a noisy measurement of its distance to agent or anchor l,

$$d_{k,l}^n = \|\boldsymbol{p}_k^n - \boldsymbol{p}_l^n\| + w_{k,l}^n. \tag{3}$$

The measurement error $w_{k,l}^n$ is assumed zero-mean Gaussian with variance σ_w^2. Note that the local likelihood function $p(d_{k,l}^n|\boldsymbol{p}_k^n, \boldsymbol{p}_l^n)$ is nonlinear in $\boldsymbol{p}_k^n$ and $\boldsymbol{p}_l^n$. Let $\boldsymbol{d}^{1:n} \triangleq \left[\boldsymbol{d}^i\right]_{i=1}^n$ with $\boldsymbol{d}^n \triangleq \left[d_{k,l}^n\right]_{(k,l) \in \mathcal{E}^n}$. Assuming that all $w_{k,l}^n$ are independent, the global likelihood function involving all measurements and all states up to time n factors according to

$$p(\boldsymbol{d}^{1:n}|\boldsymbol{x}^{1:n}) = \prod_{i=1}^n \prod_{(k,l) \in \mathcal{E}_{\mathrm{M}}^i} p(d_{k,l}^i|\boldsymbol{p}_k^i, \boldsymbol{p}_l^i) \prod_{(\kappa,\lambda) \in \mathcal{E}_{\mathrm{MA}}^i} p(d_{\kappa,\lambda}^i|\boldsymbol{p}_\kappa^i, \tilde{\boldsymbol{p}}_\lambda^i), \tag{4}$$

where $\tilde{\boldsymbol{p}}_\lambda^n$ denotes the (known) position of anchor $\lambda \in \mathcal{V}_{\mathrm{A}}^n$.

III. The Proposed Message Passing Scheme

The task of agent $k \in \mathcal{V}_{\mathrm{M}}^n$ is to estimate its state $\boldsymbol{x}_k^n$ from the total measurement vector $\boldsymbol{d}^{1:n}$, for $n \in \{1, \dots, N\}$. We will consider the minimum mean-square error (MMSE) estimator

$$\hat{\boldsymbol{x}}_k^n \triangleq \int \boldsymbol{x}_k^n \, p(\boldsymbol{x}_k^n|\boldsymbol{d}^{1:n}) \mathrm{d}\boldsymbol{x}_k^n, \quad k \in \mathcal{V}_{\mathrm{M}}^n. \tag{5}$$

Calculating the posterior pdf $p(\boldsymbol{x}_k^n|\boldsymbol{d}^{1:n})$ involved in (5) by direct marginalization of the joint posterior pdf $p(\boldsymbol{x}^{1:n}|\boldsymbol{d}^{1:n})$ is infeasible because of the excessive dimension of integration and because $\boldsymbol{d}^{1:n}$ is not locally available at the agents. Next, we develop a distributed message passing scheme that approximates $p(\boldsymbol{x}_k^n|\boldsymbol{d}^{1:n})$, $k \in \mathcal{V}_{\mathrm{M}}^n$, $n \in \{1, \dots, N\}$.

By Bayes' rule, $p(\boldsymbol{x}^{1:n}|\boldsymbol{d}^{1:n}) \propto p(\boldsymbol{d}^{1:n}|\boldsymbol{x}^{1:n})p(\boldsymbol{x}^{1:n})$, where $p(\boldsymbol{x}^{1:n})$ and $p(\boldsymbol{d}^{1:n}|\boldsymbol{x}^{1:n})$ factor as in (2) and (4), respectively. This factorization underlies the proposed hybrid BP–MF message passing scheme, which provides approximate marginal posterior pdfs ("beliefs") $q_k(\boldsymbol{x}_k^n) \approx p(\boldsymbol{x}_k^n|\boldsymbol{d}^{1:n})$ for all $k \in \mathcal{V}_{\mathrm{M}}^n$. Our scheme is an instance of the general hybrid BP–MF message passing scheme presented in [15]. We use BP for the motion-related factors $p(\boldsymbol{x}_k^n|\boldsymbol{x}_k^{n-1})$ and MF for the measurement-related factors $p(d_{k,l}^n|\boldsymbol{p}_k^n, \boldsymbol{p}_l^n)$, and we suppress all messages sent backward in time (cf. [3]). We thus obtain the following iterative scheme at time n: In message passing iteration $t \in \{1, \dots, t^*\}$, beliefs $q_k^{[t]}(\boldsymbol{x}_k^n)$ are calculated as

$$q_k^{[t]}(\boldsymbol{x}_k^n) = \frac{1}{Z} m_{k \to k}(\boldsymbol{x}_k^n) \prod_{l \in \mathcal{N}_k^n} m_{l \to k}^{[t]}(\boldsymbol{p}_k^n), \quad k \in \mathcal{V}_{\mathrm{M}}^n, \tag{6}$$

where Z is a normalization constant and $\mathcal{N}_k^n \triangleq \{l \,|\, (k,l) \in \mathcal{E}^n\}$ is the set of agents and anchors communicating with agent k at time n (termed "neighbors"). The factors in (6) are obtained as

$$m_{k \to k}(\boldsymbol{x}_k^n) = \begin{cases} \int q_k^{[t^*]}(\boldsymbol{x}_k^{n-1}) p(\boldsymbol{x}_k^n|\boldsymbol{x}_k^{n-1}) \mathrm{d}\boldsymbol{x}_k^{n-1}, \\ \qquad\qquad\qquad k \in \mathcal{V}_{\mathrm{M}}^n \cap \mathcal{V}_{\mathrm{M}}^{n-1} \\ p(\boldsymbol{x}_k^n), \quad k \in \mathcal{V}_{\mathrm{M}}^n \setminus \mathcal{V}_{\mathrm{M}}^{n-1} \end{cases} \tag{7}$$

and

$$m_{l \to k}^{[t]}(\boldsymbol{p}_k^n) = \exp\left(\int q_l^{[t-1]}(\boldsymbol{x}_l^n) \ln p(d_{k,l}^n|\boldsymbol{p}_k^n, \boldsymbol{p}_l^n) \mathrm{d}\boldsymbol{x}_l^n\right). \tag{8}$$

(Note that $\boldsymbol{p}_l^n = \tilde{\boldsymbol{p}}_l^n$ if l is an anchor.) This recursion is initialized with $q_k^{[0]}(\boldsymbol{x}_k^n) = m_{k \to k}(\boldsymbol{x}_k^n)$.

In a distributed implementation, each agent k broadcasts its belief $q_k^{[t-1]}(\boldsymbol{x}_k^n)$ to its neighbors $l \in \mathcal{N}_k^n$ and receives the neighbor beliefs $q_l^{[t-1]}(\boldsymbol{x}_l^n)$, $l \in \mathcal{N}_k^n$. These beliefs are then used to calculate the messages $m_{l \to k}^{[t]}(\boldsymbol{p}_k^n)$, $l \in \mathcal{N}_k^n$ at agent k as in (8). These messages, in turn, are needed to calculate the updated belief $q_k^{[t]}(\boldsymbol{x}_k^n)$ at agent k according to (6). After t^* iterations, the final belief $q_k^{[t^*]}(\boldsymbol{x}_k^n)$ is used for state estimation, i.e. $q_k^{[t^*]}(\boldsymbol{x}_k^n)$ is substituted for $p(\boldsymbol{x}_k^n|\boldsymbol{d}^{1:n})$ in (5).

IV. Gaussian Belief Approximation

Inspired by [14, Section IV], we introduce an approximation of the message passing scheme (6)–(8) such that the beliefs are constrained to a certain class of Gaussian pdfs. This leads to a significant reduction of both interagent communication and computational complexity relative to a particle-based implementation. We first consider MM2. A more detailed derivation is provided in [19].

A. Gaussian Belief Approximation for MM2

We constrain the beliefs to Gaussian pdfs by using the *information projection* approach [16], i.e., substituting for $q_k^{[t]}(\cdot)$ in (6)

$$\tilde{q}_k^{[t]}(\cdot) \triangleq \underset{g \in \mathcal{G}}{\arg\min} \, D\big[g \big\| q_k^{[t]}\big]. \tag{9}$$

Here, $D[g\|q] \triangleq \int g(\boldsymbol{x}) \ln \frac{g(\boldsymbol{x})}{q(\boldsymbol{x})} \mathrm{d}\boldsymbol{x}$ is the Kullback-Leibler divergence and $\mathcal{G}$ is the set of 4-D Gaussian pdfs $g(\boldsymbol{x}) = N(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{C})$ with covariance matrix of the form $\boldsymbol{C} = \begin{bmatrix} c_{\mathrm{p}} & c \\ c & c_{\mathrm{v}} \end{bmatrix} \otimes \mathbf{I}_2$. We will denote the mean and covariance matrix of $\tilde{q}_k^{[t]}(\boldsymbol{x}_k^n) = N\big(\boldsymbol{x}_k^n; (\boldsymbol{\mu}_k^n)^{[t]}, (\boldsymbol{C}_k^n)^{[t]}\big)$ defined in (9) as $(\boldsymbol{\mu}_k^n)^{[t]} = \begin{bmatrix} (\boldsymbol{\mu}_{\mathrm{p},k}^n)^{[t]} \\ (\boldsymbol{\mu}_{\mathrm{v},k}^n)^{[t]} \end{bmatrix}$ and $(\boldsymbol{C}_k^n)^{[t]} = \begin{bmatrix} (c_{\mathrm{p},k}^n)^{[t]} & (c_k^n)^{[t]} \\ (c_k^n)^{[t]} & (c_{\mathrm{v},k}^n)^{[t]} \end{bmatrix} \otimes \mathbf{I}_2$. Because direct computation of the minimizer (9) is infeasible, we resort

to an iterative method. To that end, we first derive an analytical expression of the objective function $D\big[g\big\|q_k^{[t]}\big]$ in (9), which we abbreviate by $F_k^{[t]}(\boldsymbol{\theta})$ with $\boldsymbol{\theta} \triangleq [\boldsymbol{\mu}^{\mathrm{T}}\ c_{\mathrm{p}}\ c_{\mathrm{v}}\ c]^{\mathrm{T}}$. Using the factorization in (6), this function can be expressed as

$$F_k^{[t]}(\boldsymbol{\theta}) = D[g\|m_{k\to k}] - \sum_{l\in\mathcal{N}_k^n} G_{k,l}^{[t]}(\boldsymbol{\mu}_{\mathrm{p}}, c_{\mathrm{p}}) + \gamma, \qquad (10)$$

where $\boldsymbol{\mu}_{\mathrm{p}}$ is the 2-D vector consisting of the first two entries of $\boldsymbol{\mu}$, γ is a constant, and

$$G_{k,l}^{[t]}(\boldsymbol{\mu}_{\mathrm{p}}, c_{\mathrm{p}}) \triangleq \int N(\boldsymbol{p}_k^n; \boldsymbol{\mu}_{\mathrm{p}}, c_{\mathrm{p}}\mathbf{I}_2) \ln m_{l\to k}^{[t]}(\boldsymbol{p}_k^n)\mathrm{d}\boldsymbol{p}_k^n. \quad (11)$$

To derive an expression of $D[g\|m_{k\to k}]$ in (10), we note that for $k \in \mathcal{V}_{\mathrm{M}}^n \cap \mathcal{V}_{\mathrm{M}}^{n-1}$, due to the Gaussian $\tilde{q}_k^{[t]}(\boldsymbol{x}_k^n)$ and the linear-Gaussian model (1), the message in (7) (in which $q_k^{[t^*]}(\boldsymbol{x}_k^{n-1})$ is replaced by $\tilde{q}_k^{[t^*]}(\boldsymbol{x}_k^{n-1})$) is also Gaussian, i.e., $m_{k\to k}(\boldsymbol{x}_k^n) = N(\boldsymbol{x}_k^n; \boldsymbol{\eta}_k^n, \boldsymbol{\Sigma}_k^n)$. By using (1) and standard Gaussian integral identities [20], we obtain for $k \in \mathcal{V}_{\mathrm{M}}^n \cap \mathcal{V}_{\mathrm{M}}^{n-1}$

$$\boldsymbol{\eta}_k^n = \boldsymbol{F}(\boldsymbol{\mu}_k^{n-1})^{[t^*]}, \quad \boldsymbol{\Sigma}_k^n = \boldsymbol{F}(\boldsymbol{C}_k^{n-1})^{[t^*]}\boldsymbol{F}^{\mathrm{T}} + \sigma_a^2\boldsymbol{G}\boldsymbol{G}^{\mathrm{T}}. \quad (12)$$

For $k \in \mathcal{V}_{\mathrm{M}}^n \setminus \mathcal{V}_{\mathrm{M}}^{n-1}$, $\boldsymbol{\eta}_k^n$ and $\boldsymbol{\Sigma}_k^n$ equal, respectively, the mean and covariance matrix of the Gaussian prior $p(\boldsymbol{x}_k^n) = N(\boldsymbol{x}_k^n; \boldsymbol{\eta}_k^n, \boldsymbol{\Sigma}_k^n)$. Accordingly, we obtain in either case [20]

$$\begin{aligned}D[g\|m_{k\to k}] = \frac{1}{2}\Big[&\mathrm{tr}\big((\boldsymbol{\Sigma}_k^n)^{-1}\boldsymbol{C}\big) - \ln\det(\boldsymbol{C})\\ &+ (\boldsymbol{\mu}-\boldsymbol{\eta}_k^n)^{\mathrm{T}}(\boldsymbol{\Sigma}_k^n)^{-1}(\boldsymbol{\mu}-\boldsymbol{\eta}_k^n)\Big] + \gamma',\end{aligned} \quad (13)$$

where γ' is a constant. Furthermore, one can express $G_{k,l}^{[t]}(\boldsymbol{\mu}_{\mathrm{p}}, c_{\mathrm{p}})$ in (11) via an expectation of $-(d_{k,l}^n - \|\boldsymbol{z}_{k,l}^n\|)^2/\sigma_w^2$, where $\boldsymbol{z}_{k,l}^n$ is a 2-D Gaussian random vector with mean $\boldsymbol{\mu}_{\mathrm{p}} - (\boldsymbol{\mu}_{\mathrm{p},l}^n)^{[t-1]}$ and variance $c_{\mathrm{p}} + (c_{\mathrm{p},l}^n)^{[t-1]}$. For $l \in \mathcal{V}_{\mathrm{A}}^n$, in particular, $(\boldsymbol{\mu}_{\mathrm{p},l}^n)^{[t-1]} = \tilde{\boldsymbol{p}}_l^n$ and $(c_{\mathrm{p},l}^n)^{[t-1]} = 0$. By using expressions of the first-order and second-order moments of the Rician pdf [21], one obtains [19]

$$\begin{aligned}&G_{k,l}^{[t]}(\boldsymbol{\mu}_{\mathrm{p}}, c_{\mathrm{p}})\\ &= -\frac{d_\mu^2 + 2c_{\mathrm{p}}}{2\sigma_w^2} + \frac{d_{k,l}^n}{\sigma_w^2}\sqrt{\frac{\pi C}{2}}\, M\left(-\frac{1}{2}; 1; -\frac{d_\mu^2}{2C}\right) + \gamma'',\end{aligned} \quad (14)$$

where $d_\mu \triangleq \|\boldsymbol{\mu}_{\mathrm{p}} - (\boldsymbol{\mu}_{\mathrm{p},l}^n)^{[t-1]}\|$, $C \triangleq c_{\mathrm{p}} + (c_{\mathrm{p},l}^n)^{[t-1]}$, $M(\cdot;\cdot;\cdot)$ denotes the confluent hypergeometric function of the first kind [22], and γ'' is a constant.

B. Iterative Minimization Algorithm for MM2

To derive an iterative algorithm for computing an approximation of $(\boldsymbol{\theta}_k^n)^{[t]} = \big[(\boldsymbol{\mu}_k^n)^{[t]\mathrm{T}}\ (c_{\mathrm{p},k}^n)^{[t]}\ (c_{\mathrm{v},k}^n)^{[t]}\ (c_k^n)^{[t]}\big]^{\mathrm{T}}$, i.e., of the minimizer of (10), we set the gradient of $F_k^{[t]}(\boldsymbol{\theta})$ to zero. This yields the following system of non-linear fixed-point equations $\boldsymbol{\theta} = (\chi_k^n)^{[t]}(\boldsymbol{\theta})$, whereof $(\boldsymbol{\theta}_k^n)^{[t]}$ is a solution:

$$\boldsymbol{\mu} = \boldsymbol{\eta}_k^n + \boldsymbol{\Sigma}_k^n \sum_{l\in\mathcal{N}_k^n} \frac{\partial G_{k,l}^{[t]}(\boldsymbol{\mu}_{\mathrm{p}}, c_{\mathrm{p}})}{\partial\boldsymbol{\mu}}, \qquad (15)$$

$$c_{\mathrm{p}} = \frac{c^2}{c_{\mathrm{v}}} + \left(\frac{J_{k,11}^n + J_{k,22}^n}{2} - \sum_{l\in\mathcal{N}_k^n} \frac{\partial G_{k,l}^{[t]}(\boldsymbol{\mu}_{\mathrm{p}}, c_{\mathrm{p}})}{\partial c_{\mathrm{p}}}\right)^{-1}, \quad (16)$$

$$c_{\mathrm{v}} = \frac{c^2}{c_{\mathrm{p}}} + \frac{2}{J_{k,33}^n + J_{k,44}^n}, \qquad (17)$$

$$c = \frac{1 + \sqrt{1 + (J_{k,13}^n + J_{k,24}^n)^2 c_{\mathrm{p}} c_{\mathrm{v}}}}{J_{k,13}^n + J_{k,24}^n}, \qquad (18)$$

with $J_{k,ij}^n \triangleq \big[(\boldsymbol{\Sigma}_k^n)^{-1}\big]_{ij}$. The partial derivatives in (15) and (16) can be calculated using the relation $\frac{\mathrm{d}M(-1/2;1;x)}{\mathrm{d}x} = -M(1/2;2;x)/2$ [22], where $M(-1/2;1;x)$ can be computed efficiently via an approximation [23, Section 4.5].

A Newton conjugate-gradient method [17, Chapter 7.1] is now applied to (15)–(18) to solve the system $\boldsymbol{\theta} = (\chi_k^n)^{[t]}(\boldsymbol{\theta})$ in $j_{\max}$ steps, starting from an initial value $\boldsymbol{\theta}_0$. The method iteratively computes $\boldsymbol{\theta}_{j+1} = (\mathbf{I}_7 - \boldsymbol{\Psi}_j)\boldsymbol{\theta}_j + \boldsymbol{\Psi}_j(\chi_k^n)^{[t]}(\boldsymbol{\theta}_j)$, where $\boldsymbol{\Psi}_j$ is the inverse of the Hessian matrix of $F_k^{[t]}(\boldsymbol{\theta})$ at $\boldsymbol{\theta}_j$. The Hessian matrix is approximated via the conjugate gradient, which requires only $F_k^{[t]}(\boldsymbol{\theta})$ and its gradient [17]. While the algorithm's convergence has not been proven so far, it is suggested by our simulations. The algorithm may produce a local minimum of $F_k^{[t]}(\boldsymbol{\theta})$, since this function is not convex in general. Therefore, the algorithm is run several times with different values of $\boldsymbol{\theta}_0$, and the result yielding the smallest value of $F_k^{[t]}(\boldsymbol{\theta})$ is retained. In our simulations, we used the generic routine `scipy.optimize.fmin_tnc` [24].

C. Gaussian Belief Approximation for MM1

The results in Sections IV-A and IV-B can be used with minor changes also for MM1. We here have $\boldsymbol{\mu} = \boldsymbol{\mu}_{\mathrm{p}}$ and $\boldsymbol{C} = c_{\mathrm{p}}\mathbf{I}_2$, and the Gaussian belief approximation reads $\tilde{q}_k^{[t]}(\boldsymbol{p}_k^n) = N\big(\boldsymbol{p}_k^n; (\boldsymbol{\mu}_{\mathrm{p},k}^n)^{[t]}, (c_{\mathrm{p},k}^n)^{[t]}\mathbf{I}_2\big)$. The objective function $F_k^{[t]}(\boldsymbol{\theta})$ (with $\boldsymbol{\theta} \triangleq [\boldsymbol{\mu}_{\mathrm{p}}^{\mathrm{T}}\ c_{\mathrm{p}}]^{\mathrm{T}}$) is still given by (10) together with (13) and (14); however, the expressions (12) are replaced by

$$\boldsymbol{\eta}_k^n = (\boldsymbol{\mu}_k^{n-1})^{[t^*]}, \quad \boldsymbol{\Sigma}_k^n = (\boldsymbol{C}_k^{n-1})^{[t^*]} + T\sigma_v^2\mathbf{I}_2, \qquad (19)$$

where $(\boldsymbol{\mu}_k^{n-1})^{[t^*]} = (\boldsymbol{\mu}_{\mathrm{p},k}^{n-1})^{[t^*]}$ and $(\boldsymbol{C}_k^{n-1})^{[t^*]} = (c_{\mathrm{p},k}^{n-1})^{[t^*]}\mathbf{I}_2$. Finally, fixed point equations in $\boldsymbol{\mu}_{\mathrm{p}}$ and c_{p} are obtained by setting to zero the gradient of $F_k^{[t]}(\boldsymbol{\theta})$, and an iterative belief approximation algorithm is again based on these equations.

D. Distributed Cooperative Localization Algorithm

The results of the previous subsections lead to a distributed algorithm for cooperative localization in which only parameters of Gaussian pdfs have to be communicated. At time n, agent k performs the following operations:

1. Mobility update: For $k \in \mathcal{V}_{\mathrm{M}}^n \cap \mathcal{V}_{\mathrm{M}}^{n-1}$, $\boldsymbol{\eta}_k^n$ and $\boldsymbol{\Sigma}_k^n$ are calculated from $(\boldsymbol{\mu}_k^{n-1})^{[t^*]}$ and $(\boldsymbol{C}_k^{n-1})^{[t^*]}$ as in (12) (for MM2) or as in (19) (for MM1). For $k \in \mathcal{V}_{\mathrm{M}}^n \setminus \mathcal{V}_{\mathrm{M}}^{n-1}$, $\boldsymbol{\eta}_k^n$ and $\boldsymbol{\Sigma}_k^n$ are the mean and covariance matrix of the Gaussian prior pdf $p(\boldsymbol{x}_k^n)$, which are assumed already available at agent k.

2. Iterative message passing: The message passing iterations are initialized ($t = 0$) with $(\boldsymbol{\mu}_k^n)^{[0]} = \boldsymbol{\eta}_k^n$ and $(\boldsymbol{C}_k^n)^{[0]} = \boldsymbol{\Sigma}_k^n$. At iteration $t \in \{1,\dots,t^*\}$, agent k broadcasts $(\boldsymbol{\mu}_{\mathrm{p},k}^n)^{[t-1]}$ and $(c_{\mathrm{p},k}^n)^{[t-1]}$ and receives from the neighbors $(\boldsymbol{\mu}_{\mathrm{p},l}^n)^{[t-1]}$ and
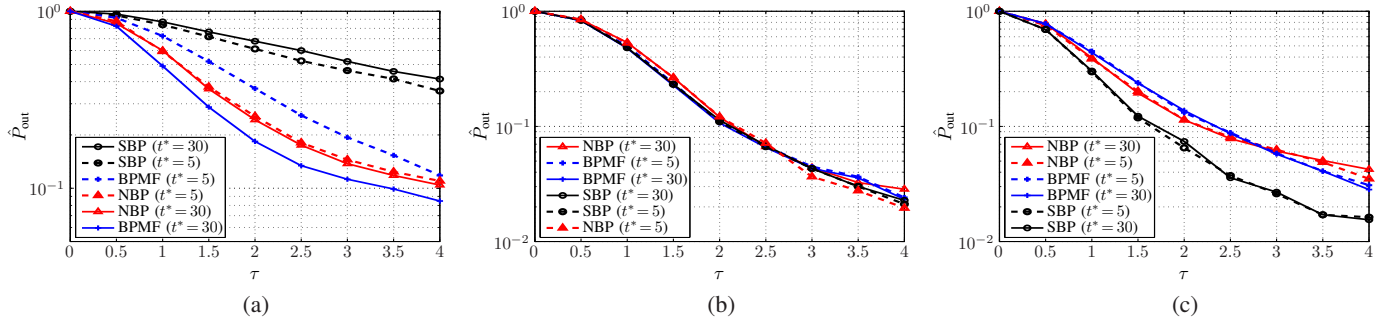
Fig. 1. Average outage probability versus outage threshold: (a) at $n = 1$ for both MMs, (b) at $n = 30$ for MM1, and (c) at $n = 30$ for MM2.

$(c_{\mathrm{p},l}^n)^{[t-1]}$, $l \in \mathcal{N}_k^n$. Note that the anchors $(l \in \mathcal{N}_k^n \cap \mathcal{V}_{\mathrm{A}}^n)$ broadcast their true position, so that $(\boldsymbol{\mu}_{\mathrm{p},l}^n)^{[t-1]} = \tilde{\boldsymbol{p}}_l^n$ and $(c_{\mathrm{p},l}^n)^{[t-1]} = 0$. Then, new parameters $(\boldsymbol{\mu}_k^n)^{[t]}$ and $(\boldsymbol{C}_k^n)^{[t]}$ are calculated using the iterative belief approximation algorithm. After the last iteration $(t = t^*)$, an approximation of the MMSE state estimate $\hat{\boldsymbol{x}}_k^n$ in (5) is obtained as $(\boldsymbol{\mu}_k^n)^{[t^*]}$. This equals the result of (5) with $p(\boldsymbol{x}_k^n | \boldsymbol{d}^{1:n})$ replaced by $\tilde{q}_k^{[t^*]}(\boldsymbol{x}_k^n)$).

The iterative belief approximation algorithm uses $\boldsymbol{\eta}_k^n$ and $\boldsymbol{\Sigma}_k^n$, which are locally available at agent k, and $(\boldsymbol{\mu}_{\mathrm{p},l}^n)^{[t-1]}$ and $(c_{\mathrm{p},l}^n)^{[t-1]}$, $l \in \mathcal{N}_k^n$, which were received from the neighbors of agent k. Therefore, at each message passing iteration t, each agent k must broadcast to its neighbors $l \in \mathcal{N}_k^n$ only three real values, namely, two for $(\boldsymbol{\mu}_{\mathrm{p},k}^n)^{[t-1]}$ and one for $(c_{\mathrm{p},k}^n)^{[t-1]}$.

V. SIMULATION RESULTS

We consider a region of interest (ROI) of size $120\,\mathrm{m} \times 120\,\mathrm{m}$ with the same $|\mathcal{V}_{\mathrm{M}}^n| = 41$ agents and $|\mathcal{V}_{\mathrm{A}}^n| = 18$ anchors at all $N = 30$ simulated time steps n. The anchors are regularly placed within the ROI. To avoid boundary effects, agents leaving the ROI reenter it at the respective opposite side. Agents and anchors have a communication radius of $20\,\mathrm{m}$; thereby, each agent communicates with one or two anchors. The agents measure distances according to (3) with $\sigma_w = 1\,\mathrm{m}$. For generating the agent trajectories, we set $T = 1\,\mathrm{s}$, $\sigma_v = \sqrt{1.5}\,\mathrm{m/s}$, and $\sigma_a = \sqrt{0.03}\,\mathrm{m/s^2}$. The initial agent positions are uniformly drawn on the ROI and, for MM2, the initial agent velocities are drawn from a Gaussian pdf with mean $[0 \; 0]^{\mathrm{T}}$ and covariance matrix $0.6 \cdot \mathbf{I}_2$. For initializing the various algorithms, the prior pdf for $\boldsymbol{p}_k^0$ is chosen Gaussian with mean $\boldsymbol{\mu}_{\mathrm{p},k}^0$ and covariance matrix $900 \cdot \mathbf{I}_2$. Here, if agent k is adjacent to one anchor l, then $\boldsymbol{\mu}_{\mathrm{p},k}^0$ is uniformly drawn from a circle of radius $d_{k,l}^0$ around the true anchor position $\tilde{\boldsymbol{p}}_l^0$, and if agent k is adjacent to two anchors l and l', then $\boldsymbol{\mu}_{\mathrm{p},k}^0$ is chosen as $(\tilde{\boldsymbol{p}}_l^0 + \tilde{\boldsymbol{p}}_{l'}^0)/2$. For MM2, the pdf for $\boldsymbol{v}_k^0$ is chosen Gaussian with mean $[0 \; 0]^{\mathrm{T}}$ and covariance matrix $0.6 \cdot \mathbf{I}_2$.

We compare the proposed hybrid BP–MF method as stated in Section IV-D (abbreviated BPMF) with nonparametric BP (NBP) and sigma point BP (SBP). NBP [8] is an extension of the particle-based BP method of [2] to mobile agents, and SBP [11] is a low-complexity sigma-point-based BP scheme in which, similarly to BPMF, only Gaussian parameters are communicated. Our simulation of NBP uses 800 particles. For simulating BPMF, we perform the fixed-point iteration (with 30 iteration steps) multiple times with different initial

values $\boldsymbol{\theta}_0$. More specifically, 20 initial values of $\boldsymbol{\mu}$ are drawn from $m_{k \to k}(\boldsymbol{x}_k^n)$, 20 are drawn from $\tilde{q}_k^{[t^*]}(\boldsymbol{x}_k^{n-1})$, and, for each adjacent anchor l, 20 are uniformly drawn from an annulus of radius $d_{k,l}^n$ and radial width $3\sigma_w$ around $\tilde{\boldsymbol{p}}_l^n$ [2]. Furthermore, the initial values of c_{p} and, for MM2, of c_{v} and c are always equal to the respective parameters of $\tilde{q}_k^{[t^*]}(\boldsymbol{x}_k^{n-1})$. Our measure of performance is the outage probability $P_{\mathrm{out}} \triangleq \mathrm{Pr}\big[\|\hat{\boldsymbol{p}}_k^n - \tilde{\boldsymbol{p}}_k^n\| > \tau\big]$, where $\tilde{\boldsymbol{p}}_k^n$ is the true position of agent k at time n, $\hat{\boldsymbol{p}}_k^n$ is a corresponding estimate, and $\tau > 0$ is a threshold.

Fig. 1 shows the simulated outage probability $\hat{P}_{\mathrm{out}}$, averaged over 30 simulation trials, of the three methods versus the outage threshold τ. It is seen that, at $n = 1$, BPMF outperforms NBP and SBP for $t^* = 30$; in particular, SBP performs poorly. Since BPMF and SBP use a Gaussian approximation, one may conclude that in the case of a noninformative prior (which is in force at $n = 1$), the Gaussian approximation degrades the performance of a pure BP scheme like SBP more than that of the proposed hybrid BP–MF scheme. At $n = 30$, for MM1, BPMF performs as NBP and SBP. However, for MM2, where the state can be predicted more accurately from the previous time, SBP outperforms both BPMF and NBP. Indeed, as previously observed in [11], SBP works very well when informative prior knowledge is available. We expect that NBP would be similarly accurate if more particles were used; however, the complexity of SBP grows quadratically with the number of particles. It is also seen that for both MMs, contrary to BPMF, the performance of NBP and SBP at $n = 30$ does not improve when t^* is increased beyond 5. We note that in less dense networks, where beliefs can be multimodal, NBP can be expected to outperform SBP and BPMF.

The communication requirements, in terms of number of real values broadcast per message passing iteration t by each agent k to adjacent agents $l \in \mathcal{N}_k^n$, are 3 for BPMF, 5 for SBP, and 1600 for NBP.

VI. CONCLUSION

The proposed algorithm for cooperative localization and tracking combines the advantages of existing BP and MF methods: its accuracy is similar to that of particle-based BP although only three real values per message passing iteration are broadcast by each agent, instead of hundreds of particles. Our simulations showed that the algorithm performs particularly well relative to pure BP-based methods when the prior information on the agent positions is imprecise.

References

[1] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero III, R. L. Moses, and N. S. Correal, "Locating the nodes: Cooperative localization in wireless sensor networks," *IEEE Signal Process. Mag.*, vol. 22, no. 4, pp. 54–69, Jul. 2005.

[2] 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 Commun.*, vol. 23, no. 4, pp. 809–819, Apr. 2005.

[3] H. Wymeersch, J. Lien, and M. Z. Win, "Cooperative localization in wireless networks," *Proc. IEEE*, vol. 97, no. 2, pp. 427–450, Feb. 2009.

[4] S. Mazuelas, A. Bahillo, R. M. Lorenzo, P. Fernandez, F. A. Lago, E. Garcia, J. Blas, and E. J. Abril, "Robust indoor positioning provided by real-time RSSI values in unmodified WLAN networks," *IEEE J. Sel. Topics Signal Process.*, vol. 3, no. 5, pp. 821–831, Oct. 2009.

[5] D. Dardari, E. Falletti, and M. Luise, *Satellite and Terrestrial Radio Positioning Techniques*. Oxford, UK: Academic Press, 2012.

[6] H.-A. Loeliger, "An introduction to factor graphs," *IEEE Signal Process. Mag.*, vol. 21, no. 1, pp. 28–41, Jan. 2004.

[7] M. A. Caceres, F. Penna, H. Wymeersch, and R. Garello, "Hybrid cooperative positioning based on distributed belief propagation," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 10, pp. 1948–1958, Dec. 2011.

[8] J. Lien, J. Ferner, W. Srichavengsup, H. Wymeersch, and M. Z. Win, "A comparison of parametric and sample-based message representation in cooperative localization," *Int. J. Navig. Observ.*, 2012.

[9] V. Savic and S. Zazo, "Reducing communication overhead for cooperative localization using nonparametric belief propagation," *IEEE Wireless Commun. Lett.*, vol. 1, no. 4, pp. 308–311, 2012.

[10] F. Meyer, O. Hlinka, H. Wymeersch, E. Riegler, and F. Hlawatsch, "Distributed localization and tracking of mobile networks including noncooperative objects," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 2, no. 1, pp. 57–71, Mar. 2016.

[11] F. Meyer, O. Hlinka, and F. Hlawatsch, "Sigma point belief propagation," *IEEE Signal Process. Lett.*, vol. 21, no. 2, pp. 145–149, Feb. 2014.

[12] M. Welling and J. Lim, "A distributed message passing algorithm for sensor localization," in *Proc. ICANN-2007*, Limassol, Cyprus, 2007, pp. 767–775.

[13] S. Van de Velde, H. Wymeersch, and H. Steendam, "Comparison of message passing algorithms for cooperative localization under NLOS conditions," in *Proc. IEEE WPNC-12*, Dresden, Germany, 2012.

[14] C. Pedersen, T. Pedersen, and B. H. Fleury, "A variational message passing algorithm for sensor self-localization in wireless networks," in *Proc. IEEE ISIT-11*, Saint Petersburg, Russia, Aug. 2011, pp. 2158–2162.

[15] E. Riegler, G. E. Kirkelund, C. N. Manchon, M. Badiu, and B. H. Fleury, "Merging belief propagation and the mean field approximation: A free energy approach," *IEEE Trans. Inf. Theory*, vol. 59, no. 1, pp. 588–602, Jan. 2013.

[16] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. Cambridge, MA, USA: MIT Press, 2009.

[17] J. Nocedal and S. Wright, *Numerical Optimization*. New York, NY: Springer, 2006.

[18] Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. New York, NY, USA: Wiley, 2001.

[19] B. Çakmak, D. N. Urup, F. Meyer, T. Pedersen, B. H. Fleury, and F. Hlawatsch, "Cooperative localization for mobile networks: A distributed belief propagation – mean field message passing algorithm (supplementary material)," 2016, Available online: http://arxiv.org/abs/1512.07782/anc/supplementaryMaterial.pdf.

[20] K. B. Petersen and M. S. Pedersen, *The Matrix Cookbook*. Copenhagen, Denmark: Technical University of Denmark, 2008.

[21] S. O. Rice, "Mathematical analysis of random noise," *Bell System Technical Journal*, vol. 23, no. 3, pp. 282–332, 1944.

[22] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. Washington D.C., USA: National Bureau of Standards, 1964.

[23] D. N. Urup, "Distributed localization in dynamic cooperative wireless sensor networks using the mean field approximation," Master's thesis, Aalborg University, Denmark, Oct. 2014.

[24] E. Jones, T. Oliphant, and P. Peterson, "Open source scientific tools for Python," 2016, Available online: http://www.scipy.org/.