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

Published in:
IEEE Signal Processing Letters

DOI (link to publication from Publisher):
10.1109/LSP.2016.2550534

Publication date:
2016

Document Version
Accepted author manuscript, peer reviewed version

Link to publication from Aalborg University

Citation for published version (APA):

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Users may download and print one copy of any publication from the public portal for the purpose of private study or research.

You may not further distribute the material or use it for any profit-making activity or commercial gain.

Users may freely distribute the URL identifying the publication in the public portal.

Take down policy
If you believe that this document breaches copyright please contact us at vbn@aub.aau.dk providing details, and we will remove access to the work immediately and investigate your claim.
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 $n \in \{1, \ldots, N\}$ 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 $\mathcal{V}_M^n$ of mobile agents at unknown positions and a set $\mathcal{V}_A^n$ of static anchors at known positions. An edge $(k, l) \in \mathcal{E}^n$ indicates the fact that agent or anchor $k$ transmits data to agent $l$ 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 $\mathcal{E}_{ML}^n$ of edges between certain agents, i.e., $(k, l) \in \mathcal{E}_{ML}^n$ implies $k, l \in \mathcal{V}_M^n$, and a set $\mathcal{E}_{MA}^n$ of edges between certain agents and anchors, i.e., $(k, l) \in \mathcal{E}_{MA}^n$ implies $k \in \mathcal{V}_M^n$ and $l \in \mathcal{V}_A^n$. Information exchange between agents is bidirectional, i.e., $(k, l) \in \mathcal{E}_{ML}^n$ implies $(l, k) \in \mathcal{E}_{ML}^n$. 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., $(k, l) \in \mathcal{E}_{MA}^n$ implies $(l, k) \notin \mathcal{E}_{MA}^n$.

Let the vector $x_k^n$ denote the state of agent $k \in \mathcal{V}_M^n$ at time $n \in \{1, \ldots, N\}$. Moreover, let $x^n \triangleq [x_k^n]_{k \in \mathcal{V}_M^n}$ and $x^{1:n} \triangleq [x^i]_{i=1}^n$. 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, $x_k^n = p_k^n \in \mathbb{R}^2$ 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., $k \in \mathcal{V}_M^n \cap \mathcal{V}_M^{n-1}$, then $p_k^n$ is assumed to evolve according to the Gaussian random walk model [18]

$$p_k^n = p_k^{n-1} + \sqrt{T} \nu_k^n.$$  

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

Final manuscript, April 3, 2016. 
B. Çakmak, T. Pedersen, and B. H. Fleury are with the Department of Electronic Systems, Aalborg University, Aalborg, Denmark (e-mail: {buc, troels, bfh}@es.aau.dk). D. N. Urup is with Danish Defence, Denmark (e-mail: dnu1984@gmail.com). F. Meyer is with CMRE, La Spezia, Italy (e-mail: flor- rian.meyer@cmre.nato.int). F. Hlawatsch is with the Institute of Telecommunications, TU Wien, Vienna, Austria (e-mail: franz.hlawatsch@nt.tuwien.ac.at). This work was supported by the European Commission in the framework of the FP7 Network of Excellence NEWCOM+ (grant 318306) and by the Austrian Science Fund (FWF) under grant P27370-N30.
Note that $v_k^n$ can be interpreted as a random velocity. In MM2, $x_k^n = (p_k^n | T^n)^T$, where $v_k^n \in \mathbb{R}^2$ is the 2-D velocity of agent $k$ at time $n$. For $k \in V_M^n \cup V_m^{n-1}$, $x_k^n$ is assumed to evolve according to the constant velocity model [18]

$$x_k^n = F x_k^{n-1} + G a_k^n. \quad (1)$$

Here, $a_k^n \in \mathbb{R}^2$ is zero-mean Gaussian driving noise (a random acceleration) with component variance $\sigma_a^2$. Moreover, $F = \begin{bmatrix} 1 & T \end{bmatrix} \otimes I_2$ and $G = \begin{bmatrix} T^2/2 \end{bmatrix} \otimes I_2$, where $\otimes$ denotes the Kronecker product and $I_m$ is the $m \times m$ identity matrix. Note that in both MM1 and MM2, the state-transition probability density function (pdf) $p(x_k^n | x_k^{n-1})$ is Gaussian. For agents that are part of the network at time $n$ but not at time $n-1$, i.e., $k \in V_M^n \setminus V_M^{n-1}$, we set $p(x_k^n | x_k^{n-1}) = p(x_k^n)$, where the prior pdf $p(x_k^n)$ is Gaussian. Under common statistical independence assumptions on $v_k^n$ or $a_k^n$ [3], the joint prior pdf of all agent states up to time $n$ is given by

$$p(x^{1:n}) = \prod_{i=1}^n \prod_{k \in V_i} p(x_k^n | x_k^{i-1}). \quad (2)$$

If $(k,l) \in \mathcal{E}^n$, agent $k \in V_M^n$ acquires at time $n$ a noisy measurement of its distance to agent or anchor $l$, $d_{k,l}^n = \|p_k^n - p_l^n\| + w_{k,l}^n. \quad (3)$

The measurement error $w_{k,l}^n$ is assumed zero-mean Gaussian with variance $\sigma_w^2$. Note that the local likelihood function $p(d_{k,l}^n | p_k^n, p_l^n)$ is nonlinear in $p_k^n$ and $p_l^n$. Let $d_{i,j}^n \equiv [d_{i,j}^n]_{i=1}^n$ with $d^n \equiv [d_{k,l}(k,l) \in \mathcal{E}^n]$. 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

$$p(d^{1:n} | x^{1:n}) = \prod_{i=1}^n \prod_{(k,l) \in \mathcal{E}_{i,k,i}^n} p(d_{k,l}^n | p_k^n, p_l^n) \prod_{(n,\lambda) \in \mathcal{E}_{M,\lambda}^n} p(d_{k,\lambda}^n | p_k^n, p_{\lambda}^n), \quad (4)$$

where $p_{\lambda}^n$ denotes the (known) position of anchor $\lambda \in V_M^n$.

III. THE PROPOSED MESSAGE PASSING SCHEME

The task of agent $k \in V_M^n$ is to estimate its state $x_k^n$ from the total measurement vector $d^{1:n}$, for $n \in \{1, \ldots, N\}$. We will consider the minimum mean-square error (MMSE) estimator

$$\hat{x}_k^n \triangleq \int x_k^n p(x_k^n | d^{1:n}) \, dx_k^n, \quad k \in V_M^n. \quad (5)$$

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

By Bayes’ rule, $p(x_k^n | d^{1:n}) \propto p(d^{1:n} | x_k^n) p(x_k^n)$, where $p(x_k^n)$ and $p(d^{1:n} | x_k^n)$ 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”) $q_k^n(x_k^n) \approx p(x_k^n | d^{1:n})$ for all $k \in V_M^n$. 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 $p(x_2^n | x_1^{n-1})$ and MF for the measurement-related factors $p(x_k^n | p_k^n, p_l^n)$, 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 $t \in \{1, \ldots, t^*\}$, beliefs $q_k^n(x_k^n)$ are calculated as

$$q_k^n(x_k^n) = \frac{1}{Z} m_{k-k}(x_k^n) \prod_{l \in N_k} m_{l-k}(p_k^n), \quad k \in V_M^n, \quad (6)$$

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

$$m_{k-k}(x_k^n) = \int q_l^{t-1}(x_k^n) p(x_k^n | x_l^{t-1}) \, dx_l^{t-1}, \quad k \in V_M^n \cap V_M^{n-1}, \quad (7)$$

and

$$m_{l-k}(p_k^n) = \exp \left( \int q_l^{t-1}(x_k^n) \ln p(d_{k,l}^n | p_k^n, p_l^n) \, dx_l^{t-1} \right). \quad (8)$$

(Note that $p_l^n \equiv \bar{p}_l^n$ if $l$ is an anchor.) This recursion is initialized with $q_k^n(x_k^n) = m_{k-k}(x_k^n)$.

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

$$q_k^n(\cdot) \triangleq \arg \min_{g \in \mathcal{G}} D[q_k^n || g_k^n]. \quad (9)$$

Here, $D[q || g] \triangleq \int q(x) \ln \frac{q(x)}{g(x)} \, dx$ is the Kullback-Leibler divergence and $\mathcal{G}$ is the set of 4-D Gaussian pdfs $g(x) = N(x; \mu, C)$ with covariance matrix of the form $C = \begin{bmatrix} \mu_1 & C_{12} \\ C_{12} & C_2 \end{bmatrix} \otimes I_2$. We will denote the mean and covariance matrix of $q_k^n(x_k^n) = N(x_k^n; \mu_k^n, C_k^n)$ defined in (9) as $\mu_k^n = \begin{bmatrix} \mu_{k,1}^n \\ \mu_{k,2}^n \end{bmatrix}$ and $C_k^n = \begin{bmatrix} C_{11}^n & C_{12}^n \\ C_{21}^n & C_{22}^n \end{bmatrix} \otimes I_2$. 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 $D[\theta]\ [q_k^t]$ in (9), which we abbreviate by $F_k^{[t]}(\theta)$ with $\theta \triangleq [\mu^T, c_p, c_v]^T$. Using the factorization in (6), this function can be expressed as

$$F_k^{[t]}(\theta) = D[\theta]\ [m_{k-k}] - \sum_{l \in N_k^c} G_{k,l}^{[t]}(\mu_p, c_p, \gamma) + \gamma',$$

(10)

where $\gamma'$ is a constant, and

$$G_{k,l}^{[t]}(\mu_p, c_p, \gamma) \triangleq \int N(p_c^l; \mu_l, c_l) \ln m_{l-k}^{[t]}(p_c^l) dp_c^l.$$  

(11)

To derive an expression of $D[\theta]\ [m_{k-k}]$ in (10), we note that for $k \in V_{M}^\delta \cap V_{M}^{n-1}$, due to the Gaussian $q_k^{[t]}(x_k^n)$ and the linear-Gaussian model (1), the message in (7) (in which $q_k^{[t]}(x_k^{n-1})$ is replaced by $q_k^{[t]}(x_k^{n-1})$) is also Gaussian, i.e., $m_{k-k}^{[t]}(x_k^n) = N(x_k^n; \mu_k^n, \Sigma_k^n)$. By using (1) and standard Gaussian identity [20], we obtain in either case [20]

$$\eta_k^n = F(\mu_k^{n-1}|^t), \quad \Sigma_k^n = F(C_{k-k}^{n-1}|^t) F^T + \sigma_k^2 GG^T.$$  

(12)

For $k \in V_{M}^\delta \cap V_{M}^{n-1}$, $\eta_k^n$ and $\Sigma_k^n$ equal, respectively, the mean and covariance matrix of the Gaussian prior $p(x_k^n) = N(x_k^n; \mu_k^n, \Sigma_k^n)$. Accordingly, we obtain in either case [20]

$$D[\theta]\ [m_{k-k}] = \frac{1}{2}\left[ \text{tr}(\Sigma_k^n) + 1 - \ln|\Sigma_k^n| \right] + \gamma'.$$

(13)

where $\gamma'$ is a constant. Furthermore, one can express $G_{k,l}^{[t]}(\mu_p, c_p, \gamma)$ in (11) via an expectation of $-(d_{k,l}^2 - \|z_{k,l}^t\|^2)/\sigma_w^2$, where $z_{k,l}^t$ is a 2-D Gaussian random vector with mean $\mu_p - (\mu_p^{[l-1]}|$ and variance $c_p + (c_p^{[l-1]}|t-1)$. For $l \in V_{M}^\delta$, in particular, $(\mu_p^{[l]}|t-1) = p_l^t$ and $(c_p^{[l]}|t-1) = 0$. By using expressions of the first-order and second-order moments of the Rician pdf [21], we obtain [19]

$$G_{k,l}^{[t]}(\mu_p, c_p) = -\frac{d_{k,l}^2 + 2c_p}{2\sigma_w^2} + \frac{d_{k,l}^2}{\sigma_w^2} \sqrt{\frac{\pi}{2}} M\left(-\frac{1}{2}; 1; -\frac{d_{k,l}^2}{2}\right) + \gamma',$$

(14)

where $d_{k,l} \triangleq \|\mu_p - (\mu_p^{[l-1]}|t-1)\|$, $C \triangleq c_p + (c_p^{[l-1]}|t-1), M(*)$ denotes the confluent hypergeometric function of the first kind [22], and $\gamma'$ is a constant.

B. Iterative Minimization Algorithm for MM2

To derive an iterative algorithm for computing an approximation of $(\theta_k^n)^{[t]} = [(\mu_k^n)^{[t]}|T], (c_p^n)^{[t]}|T], (c_v^n)^{[t]}|T]$ and $\eta_k^n$, i.e., of the minimizer of (10), we set the gradient of $F_k^{[t]}(\theta)$ to zero. This yields the following system of non-linear fixed-point equations $\theta = (\chi_k^n)^{[t]}|T$, whereof $(\theta_k^n)^{[t]}$ is a solution:

$$\mu = \eta_k^n + \sum_{l \in N_k^c} \frac{\partial G_{k,l}^{[t]}(\mu_p, c_p)}{\partial \mu},$$

(15)

$$c_p = \frac{c_v^2}{c_v} + \frac{2}{J_{k,13} + J_{k,44} - \sum_{l \in N_k^c} \frac{\partial G_{k,l}^{[t]}(\mu_p, c_p)}{\partial c_p}}.$$  

(16)

with $J_{k,ij} \triangleq \begin{bmatrix} \Sigma_k^n \end{bmatrix}_{ij}$. The partial derivatives in (15) and (16) can be calculated using the relation $\frac{dM(-1/2,i,j)}{dz} = -M(1/2; 2; x)/22$, where $M(-1/2; 2; x)$ 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 $\theta = (\chi_k^n)^{[t]}|T$ in $\max$ steps, starting from an initial value $\theta_0$. The method iteratively computes $\theta_{t+1} = (I - \Psi_t)\theta_t + \Psi_t(\chi_k^n)^{[t]}|T$, where $\Psi_t$ is the inverse of the Hessian matrix of $F_k^{[t]}(\theta)$ at $\theta_t$. The Hessian matrix is approximated via the conjugate gradient, which requires only $F_k^{[t]}(\theta)$ 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 $F_k^{[t]}(\theta)$, since this function is not convex in general. Therefore, the algorithm is run several times with different values of $\theta_0$, and the result yielding the smallest value of $F_k^{[t]}(\theta)$ is retained. In our simulations, we used the generic routine scipyc.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 $\mu = \mu_k$ and $C = c_k I_2$, and the Gaussian belief approximation reads $q_k^{[t]}(p_c^t) = N(p_c^t(\mu_k, c_k I_2))$. The objective function $F_k^{[t]}(\theta)$ (with $\theta \triangleq [\mu_k^T, c_k]^T$) is still given by (10) together with (13) and (14); however, the expressions (12) are replaced by

$$\eta_k^n = (\mu_k^{n-1}|t-1), \quad \Sigma_k^n = (C_k^{n-1}|t-1) + \sigma_k^2 I_2.$$  

(19)

where $(\mu_k^{n-1}|t-1) = (\mu_k^{n-1}|t-1)$ and $(C_k^{n-1}|t-1) = (c_k^{n-1}|t-1) I_2$. Finally, fixed point equations in $\mu_k$ and $c_k$ are obtained by setting to zero the gradient of $F_k^{[t]}(\theta)$, 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.

1. Mobility update: For $k \in V_{M}^\delta \cap V_{M}^{n-1}$, $\eta_k^n$ and $\Sigma_k^n$ are calculated from $C_{k-k}^{n-1}|t-1)$ as in (12) for (MM2) or as in (19) for (MM1). For $k \in V_{M}^\delta \cap V_{M}^{n-1}$, $\eta_k^n$ and $\Sigma_k^n$ are the mean and covariance matrix of the Gaussian prior pdf $p(x_k^n)$, which are assumed already available at agent $k$.

2. Iterative message passing: The message passing iterations are initialized ($t = 0$) with $\mu_k^{[0]} = \eta_k^n$ and $C_k^{[0]} = \Sigma_k^n$. At iteration $t \in \{1, \ldots, t\}$, agent $k$ broadcasts $(\mu_k^{[t-1]}|t-1)$ and $(c_k^{[t-1]}|t-1)$ and receives from the neighbors $(\mu_p^{[t-1]}|t-1)$ and
We consider a region of interest (ROI) of size 120m \times 120m with the same \(|Y_M^0| = 41\) agents and \(|Y_M^1| = 18\) 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 at the respective opposite side. Agents and anchors have a communication radius of 20m; thereby, each agent communicates with one or two anchors. The agents measure distances according to (3) with \(\sigma_w = 1\) m. For generating the agent trajectories, we set \(T = 1\) s, \(\sigma_v = \sqrt{1.5}\) m/s, and \(\sigma_w = \sqrt{0.05}\) m/s². 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 \([0 \ 0]^T\) and covariance matrix 0.6 \(I_2\). For initializing the various algorithms, the prior pdf for \(p_k\) is chosen Gaussian with mean \(\mu_{p,k}^0\) and covariance matrix 900 \(I_2\). Here, if agent \(k\) is adjacent to one anchor \(l\), then \(\mu_{p,k}^0\) is uniformly drawn from a circle of radius \(d_{l,k}^0\) around the true anchor position \(\bar{p}_k\), and if agent \(k\) is adjacent to two anchors \(l\) and \(l'\), then \(\mu_{p,k}^0\) is chosen as \((\bar{p}_k + \bar{p}_{k,l})/2\). For MM2, the pdf for \(v_k\) is chosen Gaussian with mean \([0 \ 0]^T\) and covariance matrix 0.6 \(I_2\).

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 \(\theta_0\). More specifically, 20 initial values of \(\mu\) are drawn from \(\mathcal{N}_{k,n} \sim \mathcal{N}(\mu_{p,k}^0, \Sigma_{p,k}^0)\), 20 are drawn from \(\mathcal{N}(\bar{p}_k, \Sigma_{p,k})\), and, for each adjacent anchor \(l\), 20 are uniformly drawn from an annulus of radius \(d_{l,k}^0\) and radial width \(3\sigma_w\) around \(\bar{p}_l\) [2]. Furthermore, the initial values of \(c_k\) and, for MM2, of \(\epsilon_k\) and \(c\) are always equal to the respective parameters of \(\mathcal{N}(\bar{x}_k, \Sigma_{k})\). Our measure of performance is the outage probability \(P_{\text{out}} = \Pr(\|\bar{x}_k - \bar{p}_l\| > \tau)\), where \(\bar{p}_l\) is the true position of agent \(k\) at time \(t\), \(\bar{p}_l\) is a corresponding estimate, and \(\tau > 0\) is a threshold.

Fig. 1 shows the simulated outage probability \(P_{\text{out}}\) averaged over 30 simulation trials, of the three methods versus the outage threshold \(\tau\). 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 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 \(l \in \mathcal{N}_k\), 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


