



# CHALMERS

## Chalmers Publication Library

### Hybrid Cooperative Positioning based on Distributed Belief Propagation

This document has been downloaded from Chalmers Publication Library (CPL). It is the author's version of a work that was accepted for publication in:

**IEEE Journal on Selected Areas in Communications (ISSN: 0733-8716)**

Citation for the published paper:

Caceres, M. ; Penna, F. ; Wymeersch, H. (2011) "Hybrid Cooperative Positioning based on Distributed Belief Propagation". IEEE Journal on Selected Areas in Communications, vol. 29(10), pp. 1948-1958.

<http://dx.doi.org/10.1109/JSAC.2011.111205>

Downloaded from: <http://publications.lib.chalmers.se/publication/151952>

Notice: Changes introduced as a result of publishing processes such as copy-editing and formatting may not be reflected in this document. For a definitive version of this work, please refer to the published source. Please note that access to the published version might require a subscription.

Chalmers Publication Library (CPL) offers the possibility of retrieving research publications produced at Chalmers University of Technology. It covers all types of publications: articles, dissertations, licentiate theses, masters theses, conference papers, reports etc. Since 2006 it is the official tool for Chalmers official publication statistics. To ensure that Chalmers research results are disseminated as widely as possible, an Open Access Policy has been adopted. The CPL service is administrated and maintained by Chalmers Library.

(article starts on next page)

# Hybrid Cooperative Positioning based on Distributed Belief Propagation

Mauricio A. Caceres, *Student Member, IEEE*, Federico Penna, *Student Member, IEEE*,  
Henk Wymeersch, *Member, IEEE*, Roberto Garelo, *Senior Member, IEEE*

**Abstract**—We propose a novel cooperative positioning algorithm that fuses information from satellites and terrestrial wireless systems, suitable for GPS-challenged scenarios. The algorithm is fully distributed over an unstructured network, does not require a fusion center, does not rely on fixed terrestrial infrastructure, and is thus suitable for ad-hoc deployment. The proposed message passing algorithm, named hybrid sum-product algorithm over a wireless network (H-SPAWN), is described and analyzed. A novel parametric message representation is introduced, to reduce computational and communication overhead. Through simulation, we show that H-SPAWN improves positioning availability and accuracy, and outperforms hybrid positioning algorithms based on conventional estimation techniques.

**Index Terms**—localization, GNSS, cooperative positioning, belief propagation, distributed signal processing

## I. INTRODUCTION

The global positioning system (GPS), developed in 1973 by the U.S. Department of Defense, has provided military support since the first Gulf war in a variety of scenarios, including soldier navigation, target tracking, reconnaissance, and missile guidance [1]. To cope with this problem, odometers and inertial sensors can be used to track the receiver position via dead reckoning when global navigation satellite systems (GNSS) is unavailable [3]. The main problem with dead reckoning is the error drift over time, so that the sensors are reliable for only very short periods of time. This makes them unsuitable for most military operations. Alternatively, the use of *hybrid positioning* has been proposed [4], where the receiver relies on terrestrial radio frequency (RF) signals, such as GSM/UMTS, DVB, and WiFi, in order to extract position-related measurements (e.g., signal strength, time of flight, or Doppler shift). However, such *systems of opportunity* rely on the presence of fixed terrestrial infrastructure, which is typically not available in hostile territory. To reduce the reliance on fixed infrastructure, peer-to-peer (P2P) *cooperative* methods have been proposed in [5]–[7], especially in the context of indoor wireless ad-hoc and sensor networks. Typically, GNSS data is not exploited by devices, thus these methods still require a limited local infrastructure to provide a geographical reference.

In order to provide reliable, GNSS-like navigation performance in hostile, GNSS-challenged environments without any fixed infrastructure, there is an urgent need to develop novel methods that can harness both the cooperative nature of the network as well as any available GNSS information. To address this need, we have introduced the concept of *hybrid cooperative positioning* [8], where a set of agents collaborate by sharing and fusing information coming from both GNSS and peers. Every agent is assumed to be equipped with a multifunction receiver comprising (i) a GNSS receiver; (ii) an RF system for short/medium distance peer-to-peer ranging (e.g., ultra wide-band (UWB) or WiFi); and (iii) a communication system (which may coincide with the ranging system). We will show how agents can help each-other in navigation by fusing information from GNSS and peers, in order to *increase positioning availability* for GNSS-denied agents and *increase positioning accuracy* for GNSS-challenged agents.

In this paper, we propose H-SPAWN, a fully distributed hybrid cooperative positioning algorithm, based on local exchange of messages between pairs of neighboring nodes. H-SPAWN is based on the sum-product algorithm (SPA) over a factor graph (FG) [9], and builds on the works from [6], [10]–[12] on RF-only non-cooperative positioning, and from [7], [13] on RF-only cooperative positioning. The main novelty of the proposed algorithm is to include GNSS information. The inclusion of GNSS adds a new dimension to the positioning problem, since the clock biases with respect to satellite clocks must be taken into account along with nodes' positions. We show that H-SPAWN is superior than conventional methods based on least-squares positioning [14] and Kalman filtering [4]. Our main contributions are as follows:

- Development of a mathematical framework for hybrid cooperative positioning, including a mapping between the network and a FG;
- Derivation of a distributed positioning algorithm (H-SPAWN), based on the SPA on the previously defined FG model;
- Introduction of an efficient parametric message representation to reduce computational complexity and communication overhead.

The paper is organized as follows: in Sec. II we formulate the problem and our assumptions. In Sec. III we derive the FG model. We then describe H-SPAWN in Sec. IV, and in Sec. V we evaluate its performance via simulation results. Concluding remarks are given in Sec. VI.

M.A. Caceres, F. Penna, and R. Garelo are with Politecnico di Torino, Italy. H. Wymeersch is with Chalmers University of Technology, Gothenburg, Sweden. E-mail: {mauricio.caceresduran, federico.penna, roberto.garelo}@polito.it, henkw@chalmers.se. This work was funded by the European Space Agency (ESA) project “Peer to Peer Positioning” (contract number 286/09/NL/AT). Part of this work was presented at the IEEE Global Communications Conference, 2010 [8].

## II. MATHEMATICAL FORMULATION

Consider a network composed by a set of agents  $\mathcal{M}$  of cardinality  $M$  and a set of satellites  $\mathcal{S}$  of cardinality  $S$ . Time is slotted  $t_0, t_1, \dots, t_k$  and will be simply denoted by the discrete time index  $k$ . Referring to a particular agent  $m \in \mathcal{M}$ , denote by  $\mathcal{M}_m^{(k)}$  the neighbors, i.e., the subset of peers agent  $m$  can communicate with at time  $k$ , and by  $\mathcal{S}_m^{(k)}$  the subset of satellites it can see. Position variables are denoted by  $\mathbf{p}_i^{(k)} = [x_i^{(k)} \ y_i^{(k)} \ z_i^{(k)}]^T$ , where  $x, y, z$  are earth-centered, earth-fixed (ECEF) coordinates of agent/satellite  $i$ , and the superscript T denotes transpose. The clock bias of node  $m$  is denoted by  $\delta_m^{(k)}$  and can be converted into distance units through the relation  $b_m^{(k)} = c \cdot \delta_m^{(k)}$ , where  $c$  is the speed of light. We thus define the state of node  $m$  as

$$\mathbf{x}_m^{(k)} \triangleq [\mathbf{p}_m^{(k)T} \ b_m^{(k)}]^T. \quad (1)$$

In the considered hybrid scenario, two types of measurements are performed by nodes:

(i) *Pseudoranges*: estimated distances from satellites

$$\rho_{s \rightarrow m}^{(k)} = \|\mathbf{p}_s^{(k)} - \mathbf{p}_m^{(k)}\| + b_m^{(k)} + v_{s \rightarrow m}^{(k)}, \quad (2)$$

(ii) *Ranges*: estimated distances between peers

$$r_{n \rightarrow m}^{(k)} = \|\mathbf{p}_n^{(k)} - \mathbf{p}_m^{(k)}\| + v_{n \rightarrow m}^{(k)}, \quad (3)$$

where the symbol  $\|\cdot\|$  denotes Euclidean distance,  $m, n \in \mathcal{M}$ ,  $s \in \mathcal{S}$ ,  $v_{s \rightarrow m}^{(k)}$  and  $v_{n \rightarrow m}^{(k)}$  are additive white Gaussian noise (AWGN) variables.

Notice that pseudorange measurements are affected by the additional unknown  $b_m^{(k)}$ , that is one of the variables to be estimated.<sup>1</sup> We introduce the following set notation to group together different nodes' variables:  $\mathbb{X}_{\mathcal{M}}^{(k)} \triangleq \{\mathbf{x}_m^{(k)} | \forall m \in \mathcal{M}\}$ ;  $\rho_{\mathcal{M}}^{(k)} \triangleq \{\rho_{s \rightarrow m}^{(k)} | \forall s \in \mathcal{S}_m^{(k)}\}$ ;  $\mathbb{P}_{\mathcal{M}}^{(k)} \triangleq \{\mathbf{p}_m^{(k)} | \forall m \in \mathcal{M}\}$ ;  $\mathbf{r}_m^{(k)} \triangleq \{r_{n \rightarrow m}^{(k)} | \forall n \in \mathcal{M}_m^{(k)}\}$ ;  $\mathbb{I}_{\mathcal{M}}^{(k)} \triangleq \{\mathbf{r}_m^{(k)} | \forall m \in \mathcal{M}\}$ .

The localization problem can be formulated as follows: every agent  $m$  needs to determine its *a posteriori* distribution of  $\mathbf{x}_m^{(k)}$ , at each time slot  $k$ , given all the available measurements:

$$p(\mathbf{x}_m^{(k)} | \mathbb{P}_{\mathcal{M}}^{(1:k)}, \mathbb{I}_{\mathcal{M}}^{(1:k)}), \quad \forall m \in \mathcal{M}, \quad (4)$$

where the superscript  $(1:k)$  denotes all variables from time step 1 to  $k$ , e.g.  $\mathbb{I}_{\mathcal{M}}^{(1:k)} \triangleq \{\mathbb{I}_{\mathcal{M}}^{(1)}, \dots, \mathbb{I}_{\mathcal{M}}^{(k)}\}$ .

We will make the following assumptions, which are reasonable approximations in many practical scenarios.

**A1:** Mobility is modeled as a Markov process, mutually independent for every node:<sup>2</sup>

$$p(\mathbb{X}_{\mathcal{M}}^{(k)} | \mathbb{X}_{\mathcal{M}}^{(k-1)}, \dots, \mathbb{X}_{\mathcal{M}}^{(0)}) = \prod_{m \in \mathcal{M}} p(\mathbf{x}_m^{(k)} | \mathbf{x}_m^{(k-1)}). \quad (5)$$

<sup>1</sup>Peer-to-peer range measurements can be performed by methods that avoid synchronization, like round-trip-time [15] or received signal strength measurements [16].

<sup>2</sup>It is assumed that each node has some internal information about its own mobility model, expressed by a p.d.f.  $p(\mathbf{x}_m^{(k)} | \mathbf{x}_m^{(k-1)})$ . However, this distribution may differ from the true mobility statistics, which depend on the users in the network. See for example simulations in Sec. V-B.

**A2:** Measurement likelihood depends only on the current state and can be split into two factors, since range and pseudorange measurements are independent:

$$p(\mathbb{P}_{\mathcal{M}}^{(k)}, \mathbb{I}_{\mathcal{M}}^{(k)} | \mathbb{X}_{\mathcal{M}}^{(0:k)}) = p(\mathbb{P}_{\mathcal{M}}^{(k)} | \mathbb{X}_{\mathcal{M}}^{(k)}) p(\mathbb{I}_{\mathcal{M}}^{(k)} | \mathbb{X}_{\mathcal{M}}^{(k)}). \quad (6)$$

**A3:** Pseudorange measurement noise samples are independent with variance known by each node:

$$v_{s \rightarrow m}^{(k)} \sim \mathcal{N}(0, \sigma_{s \rightarrow m}^2). \quad (7)$$

**A4:** Range measurement noise samples are independent, with symmetric link variance and known by both nodes:

$$v_{m \rightarrow n}^{(k)}, v_{n \rightarrow m}^{(k)} \sim \mathcal{N}(0, \sigma_{m \rightarrow n}^2). \quad (8)$$

## III. BAYESIAN INFERENCE ON FACTOR GRAPH

The goal of a Bayesian positioning approach is to determine the marginals (4) recursively at each time slot  $k$ . We assume to be given a prior distribution  $p(\mathbf{x}_m^{(0)})$ ,  $\forall m \in \mathcal{M}$ , which may be uniform if no *a priori* information is available. Marginals are given by

$$p(\mathbf{x}_m^{(k)} | \mathbb{P}_{\mathcal{M}}^{(1:k)}, \mathbb{I}_{\mathcal{M}}^{(1:k)}) = \int p(\mathbb{X}_{\mathcal{M}}^{(k-1:k)} | \mathbb{P}_{\mathcal{M}}^{(1:k)}, \mathbb{I}_{\mathcal{M}}^{(1:k)}) \partial \mathbb{X}_{\mathcal{M} \setminus m}^{(k-1:k)}, \quad (9)$$

where the subscript  $\mathcal{M} \setminus m$  denotes all variables in  $\mathbb{X}_{\mathcal{M}}^{(k-1:k)}$  except  $\mathbf{x}_m^{(k)}$ . By **A1-A4**, the joint *a posteriori* distribution becomes

$$p(\mathbb{X}_{\mathcal{M}}^{(k-1:k)} | \mathbb{P}_{\mathcal{M}}^{(1:k)}, \mathbb{I}_{\mathcal{M}}^{(1:k)}) = p(\mathbb{P}_{\mathcal{M}}^{(k)} | \mathbb{X}_{\mathcal{M}}^{(k)}) p(\mathbb{I}_{\mathcal{M}}^{(k)} | \mathbb{X}_{\mathcal{M}}^{(k)}) \times \prod_{m \in \mathcal{M}} p(\mathbf{x}_m^{(k)} | \mathbf{x}_m^{(k-1)}) p(\mathbf{x}_m^{(k-1)} | \mathbb{P}_{\mathcal{M}}^{(1:k-1)}, \mathbb{I}_{\mathcal{M}}^{(1:k-1)}). \quad (10)$$

The first two factors represent the likelihood of range and pseudorange measurements, respectively, while the subsequent factors account for temporal evolution of each node's state variable according to a given mobility model. The last factor in (10) is the *a posteriori* distribution of agent  $m$  at the previous time slot. Hence, the *a posteriori* distributions can be computed recursively. Written in this form, the above probability function can be mapped on a FG, which allows to compute efficiently the marginals (4) by applying the SPA [9]. A FG is a probabilistic graphical model that represents statistical dependencies among variables, and is used to perform Bayesian inference. Due to GNSS bias, there are two options to construct the FG: (a) considering position and bias variables jointly, or (b) treating them separately. Both options are next analyzed and discussed.

### A. Joint Position-Bias Model

Given as input  $p(\mathbf{x}_m^{(k-1)} | \mathbb{P}_{\mathcal{M}}^{(1:k-1)}, \mathbb{I}_{\mathcal{M}}^{(1:k-1)}) \forall m$ , i.e., the result of inference process at the previous time slot, the remaining part of (10) can be decomposed as

$$\prod_{m=1}^M \left[ f_m(\mathbf{x}_m^{(k)}, \mathbf{x}_m^{(k-1)}) \prod_{s \in \mathcal{S}_m} g_{s,m}(\mathbf{x}_m^{(k)}) \prod_{\substack{n \in \mathcal{M}_m \\ n < m}} h_{n,m}(\mathbf{x}_m^{(k)}, \mathbf{x}_n^{(k)}) \right], \quad (11)$$

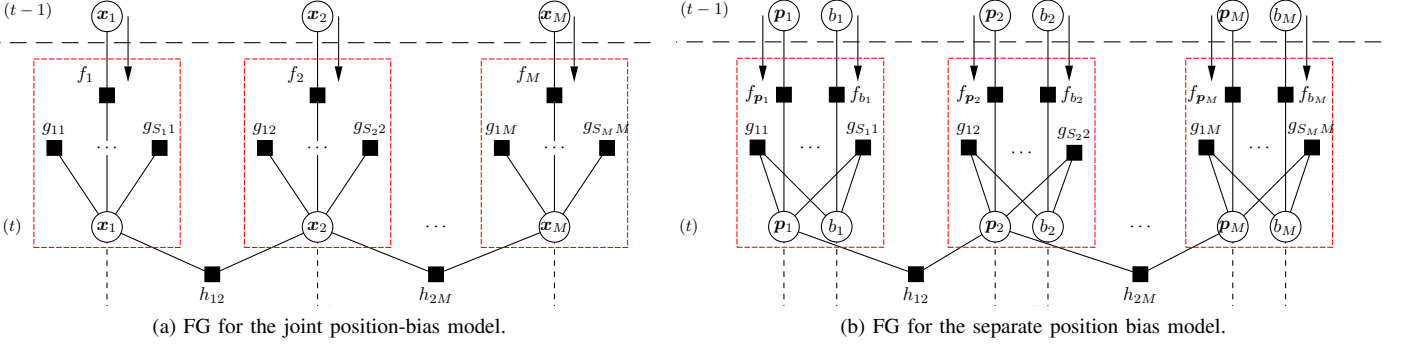


Figure 1. Factor graphs for hybrid cooperative positioning, where  $f$  represents the mobility model,  $g$  the pseudorange likelihood, and  $h$  the peer-to-peer measurement likelihood. Dashed boxes represent physical nodes, i.e., messages inside a box are computed internally by a node. On the other hand, factors connected to pairs of nodes imply packet exchanges.

where

- $f_m(\mathbf{x}_m^{(k)}, \mathbf{x}_m^{(k-1)}) \equiv p(\mathbf{x}_m^{(k)} | \mathbf{x}_m^{(k-1)})$  represents mobility.
- $g_{s,m}(\mathbf{x}_m^{(k)}) \equiv p(\rho_{s \rightarrow m} | \mathbf{x}_m^{(k)})$  represents the pseudorange measurement likelihood given the state of node  $m$ . Note that positions of the satellites do not appear as variables, since they are known exactly by nodes.<sup>3</sup>
- $h_{n,m}(\mathbf{x}_m^{(k)}, \mathbf{x}_n^{(k)}) \equiv p(r_{n \rightarrow m} | \mathbf{x}_m^{(k)}, \mathbf{x}_n^{(k)})$  represents the peer-to-peer range measurement likelihood given the positions of nodes  $m$  and  $n$ .

The resulting FG is depicted in Fig. 1a, where the vertices on top receive messages  $p(\mathbf{x}_m^{(k-1)} | \mathcal{P}_{\mathcal{M}}^{(1:k-1)}, \mathcal{I}_{\mathcal{M}}^{(1:k-1)})$  as inputs from the previous time slot.

Factors corresponding to the same node in the physical network are grouped into dashed boxes. Therefore, we have established a connection between vertices of the FG, i.e., vari-

<sup>3</sup>Satellite positions are computed through the *ephemeris* (i.e., orbital information) that is part of every message sent by GPS satellites.

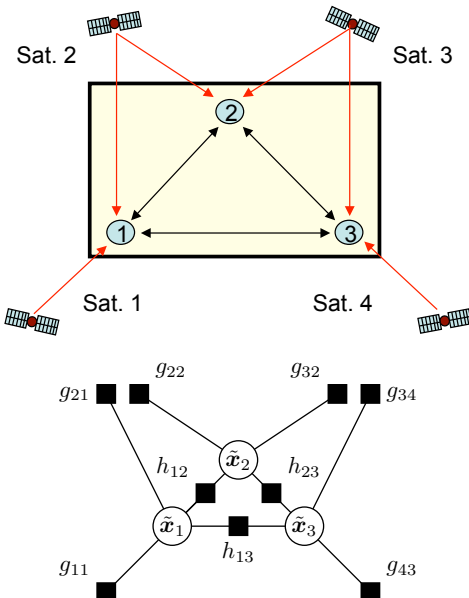


Figure 2. Example of GNSS-cooperative network and corresponding factor graph (joint position-bias variable representation).

ables to be estimated, and nodes in the network, i.e., devices that perform computations and send messages to neighbors. Then, probabilistic messages defined by the SPA algorithm (see next section) correspond to real packets exchanged among physical nodes: in this way the resulting algorithm is fully distributed. Fig. 2 shows a network configuration example and its corresponding FG.

#### B. Separate Position-Bias Model

In the alternative model, separate temporal factors exist for position and bias; range factors involve position variables only, while pseudorange factors are connected to both position and bias variables. Given as input  $p(\mathbf{x}_m^{(k-1)} | \mathcal{P}_{\mathcal{M}}^{(1:k-1)}, \mathcal{I}_{\mathcal{M}}^{(1:k-1)}) \forall m$ , the factorization of the remaining part of (10) is

$$\prod_{m=1}^M \left[ f_{p_m}(\mathbf{p}_m^{(k)}, \mathbf{p}_m^{(k-1)}) f_{b_m}(b_m^{(k)}, b_m^{(k-1)}) \times \prod_{s \in \mathcal{S}_m} g_{s,m}(\mathbf{p}_m^{(k)}, b_m^{(k)}) \prod_{\substack{n \in \mathcal{M}_m \\ n < m}} h_{n,m}(\mathbf{p}_m^{(k)}, \mathbf{p}_n^{(k)}) \right]. \quad (12)$$

This model is represented by the FG in Fig. 1b. Information from the previous time slot is passed by messages  $p(\mathbf{p}_m^{(k-1)} | \mathcal{P}_{\mathcal{M}}^{(1:k-1)}, \mathcal{I}_{\mathcal{M}}^{(1:k-1)})$  and  $p(b_m^{(k-1)} | \mathcal{P}_{\mathcal{M}}^{(1:k-1)})$  sent by vertices at level  $k-1$ .

### IV. ALGORITHM DESCRIPTION

In this section we present the proposed H-SPAWN algorithm. We first provide a high-level description of the algorithm and we describe how the message passing scheme can be implemented over a wireless network. Then, we focus on message computation, introducing a parametric implementation strategy to efficiently update and exchange H-SPAWN messages.

#### A. Summary of the Algorithm

H-SPAWN is defined by applying SPA message update rules [9] over the factor graphs of Fig. 1a or 1b. Denoting by  $\xi_m$  a generic state variable (i.e.,  $\mathbf{x}_m$ ,  $\mathbf{p}_m$ , or  $b_m$ ), messages can be classified as: (i) *temporal messages*  $\eta_{f_m \rightarrow \xi_m}$ , representing the evolution of state variable  $m$  from time  $k-1$  to time  $k$ ; (ii)

---

**Algorithm 1:** H-SPAWN – joint position-bias version
 

---

**input :** Initial beliefs  $\hat{p}(\mathbf{x}_m^{(0)}) \forall m$   
**output:** Updated beliefs  $\hat{p}(\mathbf{x}_m^{(k)}) \forall m$

```

1 for timestep  $k = 1$  to  $K$  do
2    $\forall m \in \mathcal{M}$ : compute temporal message  $\eta_{f_m \rightarrow \mathbf{x}_m^{(k)}}$  using (13)
3    $\forall m \in \mathcal{M}$ : broadcast predicted distribution as initial P2P message:  $\eta_{\mathbf{x}_m^{(k)} \rightarrow h_{n,m}} = \eta_{f_m \rightarrow \mathbf{x}_m^{(k)}}$ 
4    $\forall m \in \mathcal{M}$ : collect ranges  $r_{n \rightarrow m} \forall n \in \mathcal{M}_m^{(k)}$  and pseudoranges  $\rho_{s \rightarrow m} \forall s \in \mathcal{S}_m^{(k)}$ 
5   for iteration  $i = 1$  to  $I$  do
6     for nodes  $m \in \mathcal{M}$  in parallel do
7       Receive messages  $\eta_{\mathbf{x}_n^{(k)} \rightarrow h_{n,m}}$  from all neighbors  $n \in \mathcal{M}_m^{(k)}$ 
8       Compute local factor-to-variable messages given the likelihood of received measurements from peers and from satellites:  $\eta_{g_{s,m} \rightarrow \mathbf{x}_m^{(k)}} \forall s \in \mathcal{S}_m^{(k)}$ ;  $\eta_{h_{n,m} \rightarrow \mathbf{x}_m^{(k)}}$  using (19) and (16)
9       Compute and broadcast outgoing P2P messages  $\eta_{\mathbf{x}_m^{(k)} \rightarrow h_{n,m}} \forall n \in \mathcal{M}_m^{(k)}$  using (22)
10      Update beliefs  $\hat{p}(\mathbf{x}_m^{(k)})$  using (24)
11    end
12  end
13 end

```

---

messages from satellite factors  $\eta_{g_{s,m} \rightarrow \xi_m}$ , associated to GNSS measurements from satellites  $s \in \mathcal{S}_m^{(k)}$ ; (iii) *messages from peer-to-peer factors*  $\eta_{h_{n,m} \rightarrow \xi_m}$ , representing the information received by node  $m$  from neighbors; (iv) *messages to peer-to-peer factors*  $\eta_{\xi_m \rightarrow h_{n,m}}$ , whereby node  $m$  communicates its position message to neighbors  $n \in \mathcal{M}_m^{(k)}$ ; (v) *messages to satellite factors*  $\eta_{\mathbf{p}_m^{(k)} \rightarrow g_{s,m}}$  and  $\eta_{\mathbf{b}_m^{(k)} \rightarrow g_{s,m}}$ , used only in the separate model to link bias and position variables together.

Thanks to the identity between variables and physical nodes, H-SPAWN admits a natural distributed implementation: messages of type (i), (ii), (iii), and (v) are computed internally by each node, whereas messages of type (iv) are sent in broadcast and received by all nodes that are close enough at time  $k$  (neighbor sets  $\mathcal{M}_m$  may vary at different time slots). Beliefs  $\hat{p}(\xi_m)$  (estimated marginal probabilities) are then computed as the product of all incoming messages towards  $\xi_m$ .

The resulting algorithm and a possible message scheduling are summarized by Alg. 1, referring to joint position-bias model. For the separate model, all messages involving variables  $\mathbf{x}_m^{(k)}$  should be replaced by two separate messages for  $\mathbf{p}_m^{(k)}$  and for  $\mathbf{b}_m^{(k)}$ ; in addition, between lines 8 and 9, messages  $\eta_{\mathbf{p}_m^{(k)} \rightarrow g_{s,m}}$  and  $\eta_{\mathbf{b}_m^{(k)} \rightarrow g_{s,m}} \forall s \in \mathcal{S}_m^{(k)}$  must be updated. Observe that the algorithm works on two time scales: in the *iteration* scale, messages are updated using the same measurement data; in the *timestep* scale, new measurements are received.

### B. Parametric BP: Joint Position-Bias Model

We now illustrate how the messages defined in the previous section are computed in practice. For each type of message, we

first express the exact update rule according to the SPA. Since these expressions are numerically complex (due to integrals and multiplications in 3-D or 4-D spaces), a *parametric belief propagation (BP)* strategy can be adopted to avoid direct computation of messages. The parametric approach approximates all messages by known “distribution families”. In this way, message passing in BP reduces to simply passing the *parameters* that characterize such distributions. The advantage of parametric BP over a sample-based message representation is enormous in terms of numerical complexity: the number of samples needed to represent messages would be prohibitive for any practical wireless network.

We first consider the FG in Fig. 1a where position and bias variables are considered jointly; then, the separate case is analyzed in Sec. IV-C.

1) *Temporal messages:* The SPA update rule for temporal messages over the FG of Fig. 1a is

$$\eta_{f_m \rightarrow \mathbf{x}_m^{(k)}}(\mathbf{x}_m^{(k)}) \propto \int f_m(\mathbf{x}_m^{(k)}, \mathbf{x}_m^{(k-1)}) \times p(\mathbf{x}_m^{(k-1)} | \mathcal{P}_{\mathcal{M}}^{(1:k-1)}, \mathbb{I}_{\mathcal{M}}^{(1:k-1)}) \partial \mathbf{x}_m^{(k-1)}. \quad (13)$$

The above expression has the following meaning: each node propagates the beliefs of  $\mathbf{x}_m = [\mathbf{p}_m^T, \mathbf{b}_m^T]^T$  from time  $k-1$  to time  $k$ , according to some internal mobility model (which updates the position  $\mathbf{p}_m$ ) and clock drift model (which updates the bias  $\mathbf{b}_m$ ) jointly represented by factor  $f_m$ . Temporal messages can be interpreted as a *prediction step* in the localization process.

Adopting a parametric approach, we approximate temporal messages as  $(D+1)$ -variate Gaussian probability density functions (p.d.f.s), with mean  $\boldsymbol{\mu}_{\mathbf{x}_m^{(k)}}$  derived from  $\boldsymbol{\mu}_{\mathbf{x}_m^{(k-1)}}$  using mobility and clock drift models, and covariance  $\boldsymbol{\Sigma}_{\mathbf{x}_m^{(k)}} \succeq \boldsymbol{\Sigma}_{\mathbf{x}_m^{(k-1)}}$  (e.g., linearly increasing with the time elapsed between  $t_{k-1}$  and  $t_k$ ) which accounts for prediction uncertainty:

$$\eta_{f_m \rightarrow \mathbf{x}_m^{(k)}}(\mathbf{x}_m^{(k)}) \approx \mathcal{P}_{\mathcal{N}}(\mathbf{x}_m^{(k)}; \boldsymbol{\mu}_{\mathbf{x}_m^{(k)}}, \boldsymbol{\Sigma}_{\mathbf{x}_m^{(k)}}), \quad (14)$$

with

$$\mathcal{P}_{\mathcal{N}}(\mathbf{x}; \boldsymbol{\mu}_{\mathbf{x}}, \boldsymbol{\Sigma}_{\mathbf{x}}) \triangleq \frac{1}{Z} \exp \left[ -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_{\mathbf{x}})^T \boldsymbol{\Sigma}_{\mathbf{x}}^{-1} (\mathbf{x} - \boldsymbol{\mu}_{\mathbf{x}}) \right], \quad (15)$$

where  $Z$  is a normalization factor that may be chosen arbitrarily (messages and beliefs in BP do not necessarily need to be normalized). The shape of the  $\mathcal{N}$  distribution is a *hyper-ellipsoid*. A section of such distribution is depicted in Fig. 3a.

2) *Messages from satellite factors:* Due to the single connection of satellite factors, satellite messages simply propagate the function received from factor  $g_{s,m}$ :

$$\eta_{g_{s,m} \rightarrow \mathbf{x}_m^{(k)}} = g_{s,m}(\mathbf{x}_m^{(k)}), \quad (16)$$

i.e., the likelihood of measurement  $\rho_{s \rightarrow m}^{(k)}$  with respect to the position of satellite  $s$  and node  $m$ . Messages of this type can be represented in a parametric form as

$$\eta_{g_{s,m} \rightarrow \mathbf{x}_m^{(k)}}(\mathbf{x}_m^{(k)}) = \mathcal{P}_{\mathcal{V}}(\mathbf{x}_m^{(k)}; \rho_{s \rightarrow m}^{(k)}, \mathbf{p}_s^{(k)}, \sigma_{s \rightarrow m}^2), \quad (17)$$

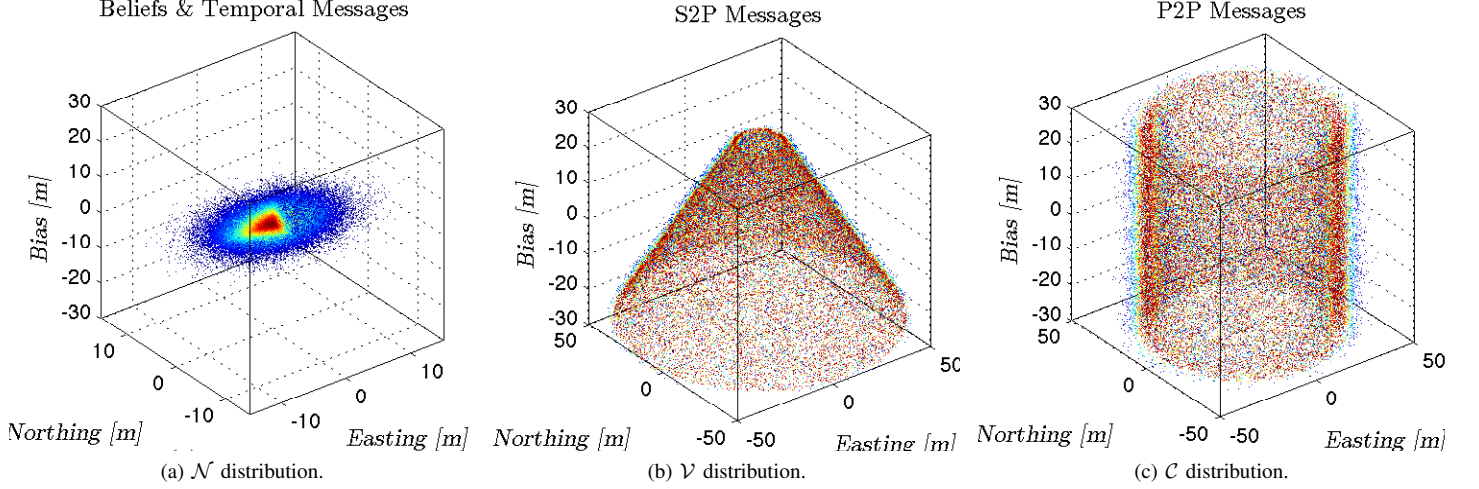


Figure 3. Distribution families for the joint position-bias model. Horizontal sections.

where  $\rho_{s \rightarrow m}^{(k)}$  is the value of the pseudorange (2) measured at the current time slot  $k$ ,  $\sigma_{s \rightarrow m}^2$  is the variance associated to this measurement (7),  $\mathbf{p}_s^{(k)}$  is the satellite position, and

$$p_{\mathcal{V}}(\mathbf{x}; \varrho, \boldsymbol{\mu}_s, \sigma_{\varrho}^2) \triangleq \frac{1}{Z} \exp \left[ -\frac{1}{2\sigma_{\varrho}^2} (\|\mathbf{p} - \boldsymbol{\mu}_s\| + b - \varrho)^2 \right] \quad (18)$$

is a *hyper-conic* distribution family parametrized by radius  $\varrho$  (pseudorange value), center  $\boldsymbol{\mu}_s$  (satellite position) and variance  $\sigma_{\varrho}^2$  (measurement variance). Note that in this case parametric BP does not introduce any approximation.

The  $\mathcal{V}$  distribution is in fact a Gaussian distribution having as support the surface of a sphere that shrinks or expands with the bias. Such support can be visualized as a spherical cone with spheres as intersections with hyperplanes perpendicular to the  $b$ -axis, and cones as intersections with  $\mathbf{p}$  planes (hence the name of hyper-conic distribution). Fig. 3b shows a section of a  $\mathcal{V}$  distribution.

3) *Messages from peer-to-peer factors*: The analytical expression of messages from P2P factors is

$$\eta_{h_{n,m} \rightarrow \mathbf{x}_m^{(k)}}(\mathbf{x}_m^{(k)}) \propto \int h_{n,m}(\mathbf{x}_m^{(k)}, \mathbf{x}_n^{(k)}) \times \eta_{\mathbf{x}_n^{(k)} \rightarrow h_{n,m}}(\mathbf{x}_n^{(k)}) d\mathbf{x}_n^{(k)}. \quad (19)$$

In (19), the term  $h_{n,m}(\mathbf{x}_m^{(k)}, \mathbf{x}_n^{(k)})$  represents the likelihood of the range measurement  $\rho_{n \rightarrow m}^{(k)}$  received by node  $m$  from its neighbor  $n$ , given the position  $\mathbf{x}_n^{(k)}$ ; the message  $\eta_{\mathbf{x}_n^{(k)} \rightarrow h_{n,m}}(\mathbf{x}_n^{(k)})$  provides a p.d.f. of  $\mathbf{x}_n^{(k)}$ , based on the available information at node  $n$  (i.e., measurements from visible satellites and all neighbors except  $m$ ). Thus, the message update rule can be interpreted as a *filtering* operation of the range likelihood function with the estimated p.d.f. of the position of node  $n$ .

In order to express P2P messages in a parametric form, let us first consider  $\mathbf{x}_n^{(k)} = \hat{\mathbf{x}}_n^{(k)}$  fixed, as though  $\eta_{\mathbf{x}_n^{(k)} \rightarrow h_{n,m}}(\mathbf{x}_n^{(k)})$  were a Dirac delta function in  $\hat{\mathbf{x}}_n^{(k)}$ . Then, the resulting message would amount to  $h_{n,m}(\mathbf{x}_m^{(k)}, \hat{\mathbf{x}}_n^{(k)})$ . This message can be visualized as a Gaussian distribution with

variance  $\sigma_{n \rightarrow m}^2$  around the surface of a *hyper-cylinder* in the  $\{\mathbf{p}, b\}$  plane, uniformly extended over the entire range of  $b$  (because P2P messages do not carry bias information), with axis  $\mathbf{p} = \hat{\mathbf{p}}_n^{(k)}$  and radius  $\rho_{n \rightarrow m}^{(k)}$ . Now, filtering  $h_{n,m}$  with  $\eta_{\mathbf{x}_n^{(k)} \rightarrow h_{n,m}}$  is a summation of an infinite number of such cylinders with different axes. The resulting distribution is still cylindrical, because Gaussian distributions are additive and the bias information in the incoming message is irrelevant for the P2P message; the variance of the resulting distribution is the sum of measurement variance and the covariance of  $\mathbf{p}_n^{(k)}$ . Thus,

$$\eta_{h_{n,m} \rightarrow \mathbf{x}_m^{(k)}}(\mathbf{x}_m^{(k)}) = p_{\mathcal{C}}(\mathbf{x}_m^{(k)}; r_{n \rightarrow m}^{(k)}, \boldsymbol{\mu}_{\mathbf{p}_{n \rightarrow m}}^{(k)}, \boldsymbol{\Sigma}_{\mathbf{p}_{n \rightarrow m}}^{(k)} + \sigma_{n \rightarrow m}^2 \mathbf{I}), \quad (20)$$

where  $\boldsymbol{\mu}_{\mathbf{p}_{n \rightarrow m}}^{(k)}$ ,  $\boldsymbol{\Sigma}_{\mathbf{p}_{n \rightarrow m}}^{(k)}$  are mean and variance of the message  $\eta_{\mathbf{x}_n^{(k)} \rightarrow h_{n,m}}(\mathbf{x}_n^{(k)})$ ,  $\mathbf{I}$  is the identity matrix, and

$$p_{\mathcal{C}}(\mathbf{x}; \varrho, \boldsymbol{\mu}_{\mathbf{p}_n}, \boldsymbol{\Sigma}_{\mathbf{p}_n + \varrho}) \triangleq \frac{1}{Z} \exp \left[ -\frac{1}{2} (\mathbf{p}' - \varrho \frac{\mathbf{p}'}{\|\mathbf{p}'\|})^T \boldsymbol{\Sigma}_{\mathbf{p}_n + \varrho}^{-1} (\mathbf{p}' - \varrho \frac{\mathbf{p}'}{\|\mathbf{p}'\|}) \right], \quad (21)$$

where  $\mathbf{p}' \triangleq \mathbf{p} - \boldsymbol{\mu}_{\mathbf{p}_n}$ . A section of the *hyper-cylindrical* distribution family  $\mathcal{C}$  is shown in Fig. 3c.

4) *Messages to peer-to-peer factors*: According to the SPA, messages from variable nodes to P2P factors are given by

$$\eta_{\mathbf{x}_m^{(k)} \rightarrow h_{n,m}}(\mathbf{x}_m^{(k)}) \propto \eta_{f_m \rightarrow \mathbf{x}_m^{(k)}}(\mathbf{x}_m^{(k)}) \times \prod_{s \in \mathcal{S}_m^{(k)}} \eta_{g_{s,m} \rightarrow \mathbf{x}_m^{(k)}}(\mathbf{x}_m^{(k)}) \prod_{l \in \mathcal{M}_m^{(k)} \setminus n} \eta_{h_{l,m} \rightarrow \mathbf{x}_m^{(k)}}(\mathbf{x}_m^{(k)}), \quad (22)$$

Due to heterogeneous shape and varying number of incoming messages, it is impossible to give a general expression of the resulting distribution. In addition, even if computed analytically, such distribution would be a very complex and irregular function, making it difficult to express messages at the subsequent iteration. For these reasons, we approximate the result of message multiplication as a multivariate Gaussian

distribution

$$\eta_{\mathbf{x}_m^{(k)} \rightarrow h_{n,m}}(\mathbf{x}_m^{(k)}) \approx p_{\mathcal{N}}(\mathbf{x}_m^{(k)}; \boldsymbol{\mu}_{\mathbf{x}_m^{(k)} \rightarrow n}, \boldsymbol{\Sigma}_{\mathbf{x}_m^{(k)} \rightarrow n}). \quad (23)$$

The problem then reverts to finding the parameters of the output distribution that best approximate the product of the incoming parametric messages. To this purpose, we propose a parameter estimation algorithm (see Alg. 2) inspired from importance sampling [17]. At each iteration, samples  $\mathbf{z}_i$  are drawn according to the current estimated output distribution, and weighted proportional to the ratio between the product of the probabilities of each of the incoming messages (computed according to their respective analytical distributions evaluated at the given samples) and the probability of the sample according to the output distribution. The distribution parameters are then refined through weighted sample mean and covariance estimators, and used to draw a new set of samples in the next iteration. We do not impose restrictions on the covariance matrix, so it can take any ellipsoidal shape.

---

**Algorithm 2:** Parametric Message Multiplication

---

**input** : Initial estimate of  $\hat{\boldsymbol{\mu}}_{\mathbf{x}}, \hat{\boldsymbol{\Sigma}}_{\mathbf{x}}$ ; distributions of all incoming messages (factors)

**output**: Updated  $\hat{\boldsymbol{\mu}}_{\mathbf{x}}, \hat{\boldsymbol{\Sigma}}_{\mathbf{x}}$  after product

---

```

1 repeat
2   Draw  $N$  samples  $\mathbf{z}_i$  from  $\mathcal{N}(\hat{\boldsymbol{\mu}}_{\mathbf{x}}, \hat{\boldsymbol{\Sigma}}_{\mathbf{x}})$ 
3   Compute the probability of each sample  $q(\mathbf{z}_i)$  in the
   distribution it was drawn from, using (15)
4   Evaluate the p.d.f.s at samples  $p_j(\mathbf{z}_i)$  for each factor,
   using (15), (18), or (21)
5   Assign a weight to each sample as:  $w_i = \frac{\prod_j p_j(\mathbf{z}_i)}{q(\mathbf{z}_i)}$ , then
   normalize them such that  $\sum_{i=1}^N w_i = 1$ 
6   Estimate new mean and covariance using weighted samples:
    $\hat{\boldsymbol{\mu}}_{\mathbf{x}} = \sum_{i=1}^N w_i \mathbf{z}_i, \hat{\boldsymbol{\Sigma}}_{\mathbf{x}} = \frac{\sum_{i=1}^N w_i (\mathbf{z}_i - \hat{\boldsymbol{\mu}}_{\mathbf{x}})(\mathbf{z}_i - \hat{\boldsymbol{\mu}}_{\mathbf{x}})^T}{1 - \sum_{i=1}^N w_i^2}$ 
7 until convergence

```

---

5) *Beliefs*: Finally, beliefs  $\hat{p}(\mathbf{x}_m^{(k)})$ , i.e. approximations of the marginals of interest  $p(\mathbf{x}_m^{(k)} | \mathcal{D}_{\mathcal{M}}^{(1:k)}, \mathcal{I}_{\mathcal{M}}^{(1:k)})$  (9), are given by

$$\hat{p}(\mathbf{x}_m^{(k)}) \propto \eta_{f_m \rightarrow \mathbf{x}_m^{(k)}}(\mathbf{x}_m^{(k)}) \prod_{s \in \mathcal{S}_m^{(k)}} \eta_{g_{s,m} \rightarrow \mathbf{x}_m^{(k)}}(\mathbf{x}_m^{(k)}) \times \prod_{n \in \mathcal{M}_m^{(k)}} \eta_{h_{n,m} \rightarrow \mathbf{x}_m^{(k)}}(\mathbf{x}_m^{(k)}). \quad (24)$$

The above expression is a message multiplication similar to the one used for messages to P2P factors. Hence, the same parametric message multiplication algorithm (Alg. 2) can be used to compute beliefs:

$$\hat{p}(\mathbf{x}_m^{(k)}) \approx p_{\mathcal{N}}(\mathbf{x}_m^{(k)}; \boldsymbol{\mu}_{\mathbf{x}_m^{(k)}}, \boldsymbol{\Sigma}_{\mathbf{x}_m^{(k)}}). \quad (25)$$

### C. Parametric BP: Separate Position-Bias Model

We now describe the message update rules for the FG of Fig. 1b.

1) *Temporal messages*: Similar to the joint case, temporal messages are approximated by Gaussian distribution, but separately for bias and position variables:

$$\eta_{f_{\mathbf{p}_m} \rightarrow \mathbf{p}_m^{(k)}}(\mathbf{p}_m^{(k)}) \approx p_{\mathcal{N}}(\mathbf{p}_m^{(k)}; \boldsymbol{\mu}_{\mathbf{p}_m^{(k)}}, \boldsymbol{\Sigma}_{\mathbf{p}_m^{(k)}}), \quad (26)$$

$$\eta_{f_{b_m} \rightarrow b_m^{(k)}}(b_m^{(k)}) \approx p_{\mathcal{N}}(b_m^{(k)}; \mu_{b_m^{(k)}}, \sigma_{b_m^{(k)}}^2), \quad (27)$$

where  $\boldsymbol{\mu}_{\mathbf{p}_m^{(k)}}$  and  $\mu_{b_m^{(k)}}$  are computed, respectively, from mobility and clock drift models, and  $\boldsymbol{\Sigma}_{\mathbf{p}_m^{(k)}}$  and  $\sigma_{b_m^{(k)}}^2$  are updated such that  $\boldsymbol{\Sigma}_{\mathbf{p}_m^{(k)}} \succeq \boldsymbol{\Sigma}_{\mathbf{p}_m^{(k-1)}}$ ,  $\sigma_{b_m^{(k)}}^2 \geq \sigma_{b_m^{(k-1)}}^2$  to take into account prediction uncertainty.

2) *Messages from satellite factors*: Messages from factors  $g_{s,m}$  can now be divided into two types: messages to position variables, that take as input an estimated bias value, and messages to bias variables, which, conversely, take as input a position value. Based on similar considerations as in Sec. IV-B3, *satellite-to-position* messages can be computed as

$$\eta_{g_{s,m} \rightarrow \mathbf{p}_m^{(k)}}(\mathbf{p}_m^{(k)}) = p_{\mathcal{D}}(\mathbf{p}_m^{(k)}; \rho_{s \rightarrow m}^{(k)} - \mu_{b_m^{(k)}}, \mathbf{p}_s^{(k)}, \sigma_{s \rightarrow m}^2 + \sigma_{b_m^{(k)}}^2), \quad (28)$$

where

$$p_{\mathcal{D}}(\mathbf{p}; \varrho, \boldsymbol{\mu}_s, \sigma_{\varrho}^2) \triangleq \frac{1}{Z} \exp \left[ -\frac{1}{2\sigma_{\varrho}^2} (\|\mathbf{p} - \boldsymbol{\mu}_s\| - \varrho)^2 \right] \quad (29)$$

denotes a Gaussian distribution with *spherical* support (cf. [18]), characterized by radius  $\varrho$  (pseudorange measurement), center  $\boldsymbol{\mu}_s$  (satellite position), and variance  $\sigma_{\varrho}^2$ . An example of  $\mathcal{D}$  distribution is depicted in Fig. 4b.

*Satellite-to-bias* messages are represented by a univariate Gaussian p.d.f.:

$$\eta_{g_{s,m} \rightarrow b_m^{(k)}}(b_m^{(k)}) = p_{\mathcal{N}}(b_m^{(k)}; \rho_{s \rightarrow m}^{(k)} - \|\mathbf{p}_s^{(k)} - \boldsymbol{\mu}_{\mathbf{p}_m^{(k)}}\|, \sigma_{s \rightarrow m}^2 + \text{tr}(\boldsymbol{\Sigma}_{\mathbf{p}_m^{(k)}})), \quad (30)$$

where  $\text{tr}(\cdot)$  denotes the trace operator.

3) *Messages from peer-to-peer factors*: Like it the joint case, P2P messages carry only position information. The cylindrical distribution therefore reduces to an *ellipsoidal* distribution in the  $\mathbf{p}$  space, i.e., a section of the  $\mathcal{C}$  distribution perpendicular to the  $b$  axis:

$$\eta_{h_{n,m} \rightarrow \mathbf{p}_m^{(k)}}(\mathbf{p}_m^{(k)}) = p_{\mathcal{E}}(\mathbf{p}_m^{(k)}; r_{n \rightarrow m}^{(k)}, \boldsymbol{\mu}_{\mathbf{p}_n^{(k)}}, \boldsymbol{\Sigma}_{\mathbf{p}_n^{(k)}} + \sigma_{n \rightarrow m}^2 \mathbf{I}), \quad (31)$$

where

$$p_{\mathcal{E}}(\mathbf{p}; \varrho, \boldsymbol{\mu}_{\mathbf{p}_n}, \boldsymbol{\Sigma}_{\mathbf{p}_n + \varrho}) \triangleq \frac{1}{Z} \exp \left[ -\frac{1}{2} \left( \mathbf{p}' - \varrho \frac{\mathbf{p}'}{\|\mathbf{p}'\|} \right)^T \boldsymbol{\Sigma}_{\mathbf{p}_n + \varrho}^{-1} \left( \mathbf{p}' - \varrho \frac{\mathbf{p}'}{\|\mathbf{p}'\|} \right) \right] \quad (32)$$

with  $\mathbf{p}' = \mathbf{p} - \boldsymbol{\mu}_{\mathbf{p}_n}$ . The  $\mathcal{E}$  distribution is depicted in Fig. 4c. Observe that, for  $\varrho = 0$ , it becomes a Gaussian distribution  $\mathcal{N}(\boldsymbol{\mu}_{\mathbf{p}_n}, \boldsymbol{\Sigma}_{\mathbf{p}_n + \varrho})$ .



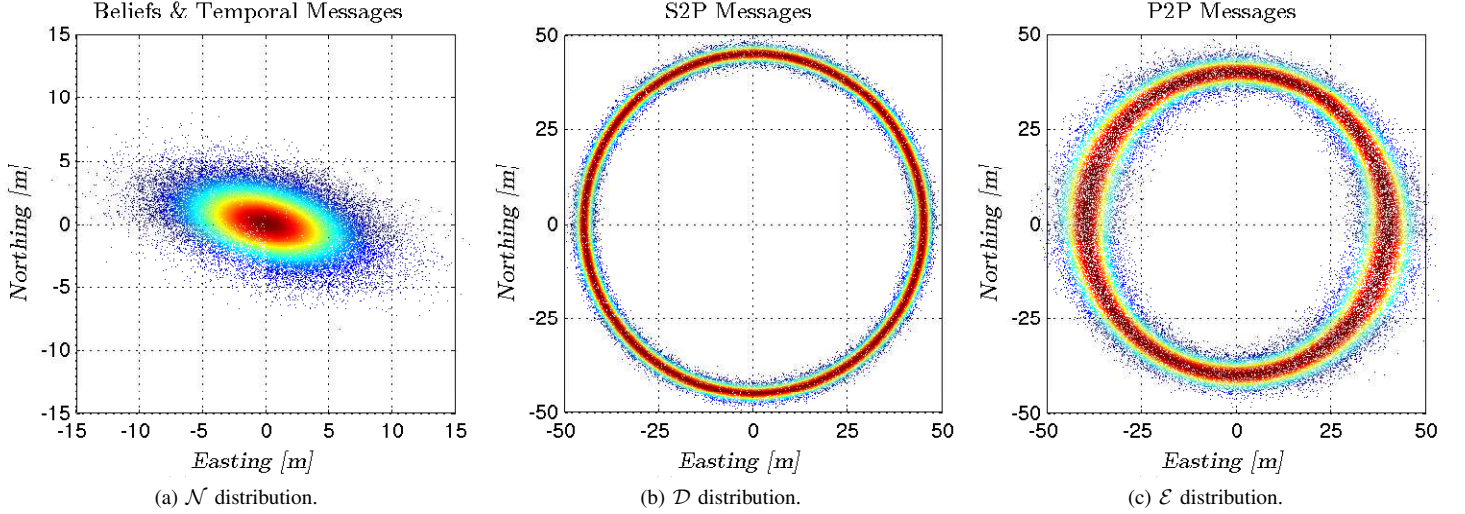


Figure 4. Distribution families for the separate position bias model. Horizontal sections.

4) *Messages to peer-to-peer factors:* These messages, having a product form as

$$\eta_{\mathbf{p}_m^{(k)} \rightarrow h_{m,n}}(\mathbf{p}_m^{(k)}) \propto \eta_{f_{\mathbf{p}_m} \rightarrow \mathbf{p}_m^{(k)}}(\mathbf{p}_m^{(k)}) \times \prod_{s \in \mathcal{S}_m^{(k)}} \eta_{g_{s,m} \rightarrow \mathbf{p}_m^{(k)}}(\mathbf{p}_m^{(k)}) \prod_{l \in \mathcal{M}_m^{(k)} \setminus n} \eta_{h_{l,m} \rightarrow \mathbf{p}_m^{(k)}}(\mathbf{p}_m^{(k)}), \quad (33)$$

are approximated by  $D$ -variate Gaussian distributions  $\mathcal{N}$ , whose parameters are computed via the parametric message multiplication algorithm (Alg. 2).

5) *Messages to satellite factors:* These messages are unique to the separate representation. They include *bias-to-satellite messages*,

$$\eta_{b_m^{(k)} \rightarrow g_{s,m}}(b_m^{(k)}) \propto \eta_{f_{b_m} \rightarrow b_m^{(k)}}(b_m^{(k)}) \prod_{v \in \mathcal{S}_m^{(k)} \setminus s} \eta_{g_{v,m} \rightarrow b_m^{(k)}}(b_m^{(k)}), \quad (34)$$

which can be computed as univariate Gaussian distributions with

$$\sigma_{b_m^{(k)} \rightarrow g_{s,m}}^2 = \left[ \frac{1}{\sigma_{f_{b_m} \rightarrow b_m^{(k)}}^2} + \sum_{y \in \mathcal{S}_m^{(k)} \setminus s} \frac{1}{\sigma_{g_{y,m} \rightarrow b_m^{(k)}}^2} \right]^{-1}, \quad (35)$$

$$\mu_{b_m^{(k)} \rightarrow g_{s,m}} = \sigma_{b_m^{(k)} \rightarrow g_{s,m}}^2 \left[ \frac{\mu_{f_{b_m} \rightarrow b_m^{(k)}}}{\sigma_{f_{b_m} \rightarrow b_m^{(k)}}^2} + \sum_{y \in \mathcal{S}_m^{(k)} \setminus s} \frac{\mu_{g_{y,m} \rightarrow b_m^{(k)}}}{\sigma_{g_{y,m} \rightarrow b_m^{(k)}}^2} \right], \quad (36)$$

and *position-to-satellite messages*  $\eta_{\mathbf{p}_m^{(k)} \rightarrow g_{s,m}}$ , which can be approximated as Gaussian distributions using Alg. 2, similar to (33).

6) *Beliefs:* Position beliefs  $\hat{\mathbf{p}}(\mathbf{p}_m^{(k)})$  are then computed like (33), including all incoming P2P messages  $n \in \mathcal{M}_m$ . Bias beliefs  $\hat{\mathbf{p}}(b_m^{(k)})$  are computed like (34), i.e., univariate Gaussian distributions with mean and variance given by (35,36), including all satellites  $s \in \mathcal{S}_m$  in the summation.

#### D. Complexity

Complexity of the H-SPAWN algorithm is dominated by message multiplication needed in messages to P2P factors and messages from position variables to satellite factors. Considering a certain node agent with one temporal message,  $P$  peer-to-peer messages and  $S$  satellite-to-peer messages, using  $N$  samples  $\mathbf{z}_i$  and requiring  $I$  iterations in the product estimate, the complexity scales as  $\mathcal{O}(IPN(S + P + 1)(D + 1)^3)$ . In contrast, all other messages are computed analytically in  $\mathcal{O}((S + P + 1)(D + 1))$ . Table I lists the number of floating point operations (FLOP) needed in one execution of Alg. 2.

Table I  
COMPLEXITY OF H-SPAWN FOR A SINGLE AGENT.

Computation	No. of FLOP
Compute time message	$2(D + 1)^2 + 2(D + 1)^3$
<i>The following operations are repeated <math>P + 1</math> times and iterated <math>I</math> times</i>	
Draw multiplication samples	$N(D + 1)^2$
Evaluate message distributions	$N(S + P + 2)(D + 1)^3$
Compute message multiplication	$N(S + P + 4)$
Estimate mean	$2N(D + 1)$
Estimate covariance	$N(D + 1)^2$

## V. SIMULATION RESULTS

In this section the performance of H-SPAWN is analyzed in two scenarios: a small static network, as a proof of concept, and a large mobile network, as a realistic case. The performance of H-SPAWN is compared in both scenarios with two classical approaches: a hybrid-cooperative weighted least squares (HC-WLS) algorithm, inspired on the iterative descent method proposed in [7] and extended to hybrid GNSS-terrestrial ranging like in [14]; and the hybrid-cooperative extended Kalman filter (HC-EKF) algorithm presented in [19].

#### A. Small-scale Static Scenario

The first scenario, shown in Fig. 5a, is defined as follows:

- Real GPS satellite orbits, seen from the equator.



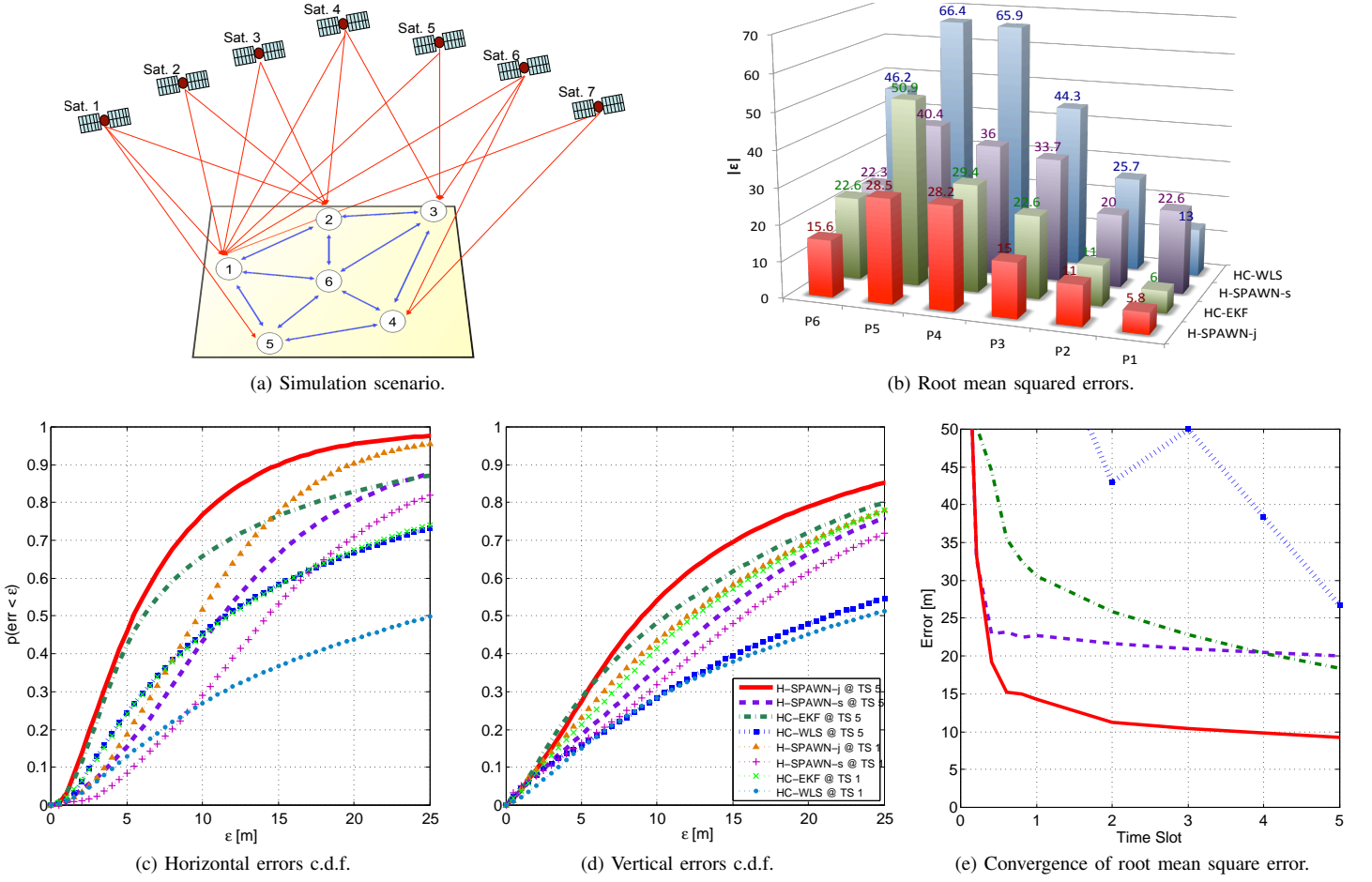


Figure 5. Simulation scenario and RMSE by agent (top), global error c.d.f. and RMSE evolution averaged over all agents (bottom). Results after 1 and 5 time slots, 100 Monte Carlo runs.

Table II  
SATELLITE POSITIONS IN ECEF AND PEER POSITIONS IN EASTING-NORTHING-UP (ENU)  
(ORIGIN AT 45.065275° LAT., 7.658954° LONG., 311.96 M HEIGHT)

Sat.	1	2	3	4	5	6	7	Peer	1	2	3	4	5	6
$x$ [m]	19263524	26124976	24768710	8048029	8543818	2082386	-7307117	E[m]	-50	0	50	30	-30	0
$y$ [m]	-13725770	-5749420	1601307	-13014437	15561017	23437415	-14002994	N[m]	10	30	40	-20	-40	0
$z$ [m]	11583188	-846377	9925575	21563572	19676844	12048145	21528812	U[m]	40	55	50	42	51	49

- 6 nodes deployed on the Earth surface over an area of  $100 \times 100$  m, as in Table II.
- Uniform initial distributions in a circle of radius 1 km. (a *warm-start* condition, i.e., nodes have prior knowledge of the region where they are located when turned on).
- Static nodes. As a consequence, the following model is used when updating temporal messages:  $\mathbf{x}_m^{(k)} = \mathbf{x}_m^{(k-1)}$ ;  $\Sigma_{\mathbf{x}_m}^{(k)} = \Sigma_{\mathbf{x}_m}^{(k-1)} + \sigma^2 \mathbf{I}$ , with  $\sigma^2 = 4$ .
- Pseudorange and P2P range measurements are corrupted by AWGN with standard deviation  $\sigma_{s \rightarrow m} = 5$  m and  $\sigma_{n \rightarrow m} = 0.2$  m respectively, typical values for GNSS in open sky conditions and UWB-based ranging in line-of-sight (LoS).
- New measurements are generated at each time slot, and algorithms run 5 iterations per slot.

The performance of H-SPAWN, HC-EKF, and HC-WLS in the above scenario is evaluated in terms of RMSE, computed

from the error vectors  $\epsilon_m^{(k)} \triangleq \mu_{p_m}^{(k)} - p_m^{(k)}$  for all agents  $m \in \mathcal{M}$  over 100 Monte Carlo simulations. The global localization RMSE is reported for each node in Fig. 5b, comparing the three considered algorithms after five time slots. H-SPAWN in its joint position-bias version outperforms all the other algorithms, while HC-WLS provides the worst accuracy overall. In order to evaluate the benefit of cooperation, we analyze the performance achieved by nodes in relation to their connectivity conditions. Nodes 1 and 2 (which have enough satellites in view) achieve the least errors. More interesting are the cases of peers 3-5, which have 3, 2 and 1 satellite in view respectively and are therefore unable to localize themselves using GNSS only. For these nodes, localization is made possible thanks to cooperation, although with larger errors compared to nodes with complete satellite visibility. Finally, node 6, even without visible satellites, achieves good performance, thanks to collaboration with the other nodes in

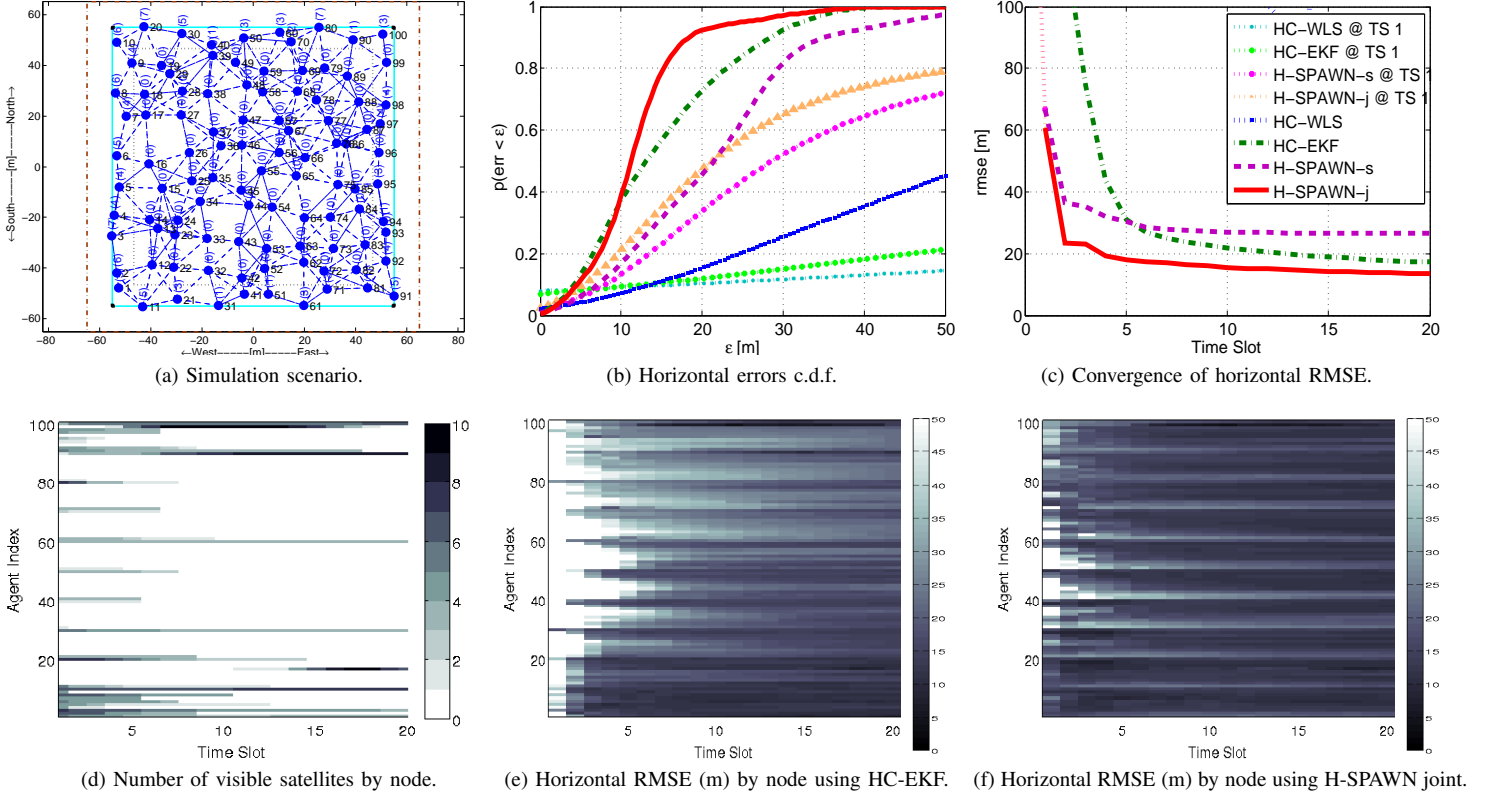


Figure 6. Large-scale mobile simulation scenario; numbers in brackets = number of visible satellites. Performance in terms of horizontal error c.d.f. (top-center) and RMSE evolution (top-right) averaged over all agents, after 1 and 20 time slots. Bottom: node-by-node satellite visibility (left) and RMSE evolution using HC-EKF (center) or H-SPAWN (right).

the network. RMSE values are close to the Cramér-Rao lower bound, derived for the same scenario in [20], thus confirming the effectiveness of the proposed algorithm implementation.

A more detailed comparison is shown in Figs. 5c-5d, where the c.d.f.s of the global errors over all nodes in the network are plotted at the first and the fifth time slot. A coordinate transformation from ECEF to ENU is performed in order to analyze errors in horizontal and vertical components, as it is usual in GNSS performance tests. It can be noticed how H-SPAWN-joint, after a single time slot, achieves superior accuracy than H-SPAWN-separate after 5 slots, thus providing faster convergence. With more time slots, estimates are enhanced thanks to the availability of new measurements, and H-SPAWN-joint still obtains superior performance compared to other algorithms with the same number of measurements. Differences are more evident in the horizontal component than in the vertical one, where the HC-EKF has a performance close to that of H-SPAWN-joint.

The convergence of the considered algorithms is shown in Figure 5e, which illustrates the evolution of the global RMSE iteration by iteration over 5 time slots. H-SPAWN-joint rapidly achieves a good performance in a few iterations, and slightly improves it with more measurements.

### B. Large-scale Mobile Scenario

In order to assess the performance of the proposed algorithm in a realistic, challenging scenario, we now consider a network

of 100 mobile agents placed inside a building. Only nodes that are close to the walls have partial visibility of the GNSS constellation; in addition, it is assumed that nodes are able to communicate with peers if their distance is lower than a certain threshold, obtaining in this way a mesh network, with several hops between internal nodes (without any satellite visibility) and the outer ones (with partial visibility). The scenario is depicted in Fig. 6a and is defined as follows:

- Real GPS satellite orbits from 45° latitude.
- 100 nodes deployed on the Earth surface inside a building of  $100 \times 100$  m, with *warm start* in a circle of radius 10 km.
- *Random-waypoint* mobility model [21]: each node moves towards a random point (selected with uniform probability) at constant velocity, uniformly chosen between 1 and 2m/s, remains in such point for a random time (uniform between 0 and 1s), and then moves to a new point. To make our simulation more realistic, it is assumed that the mobility model is unknown to nodes. Therefore, temporal messages follow the usual model with  $\mathbf{x}_m^{(k)} = \mathbf{x}_m^{(k-1)}$ ,  $\Sigma_{\mathbf{x}_m^{(k)}} = \Sigma_{\mathbf{x}_m^{(k-1)}} + \sigma^2 \mathbf{I}$ , with  $\sigma^2 = 4$ .
- Pseudorange are corrupted by AWGN with standard deviations varying from 10 to 20m depending on the satellite elevation angle seen from the peer (cf. [22]), modeling in this way *multipath fading* and *jitter* due to low carrier-to-noise ratio ( $C/N_0$ ).
- P2P communication between nodes, say  $m$  and  $n$ , is

possible only if  $d_{m,n} \triangleq \|\mathbf{p}_m - \mathbf{p}_n\| < 20\text{m}$ . Range measurements are corrupted by AWGN with  $\sigma_{m \rightarrow n} = 0.2\text{m}$  if  $d_{m,n} \leq 10\text{m}$ , or  $\sigma_{m \rightarrow n} = 0.4\text{m}$  if  $10 < d_{m,n} \leq 20\text{m}$ .

- New measurements are generated at each time slot. Algorithms are stopped after 3 iterations per slot. In order to better appreciate the effect of mobility, every Monte Carlo run consists now of 20 slots with duration of 1s.

Fig. 6b shows the horizontal error c.d.f.s achieved by the considered algorithms in the above scenario, at the first and last time slot. Fig. 6c shows the evolution of the RMSE as a function of time slots. H-SPAWN in the joint version confirms its superiority compared to HC-WLS and HC-EKF. Remarkably, all algorithms exhibit a significant improvement of the estimated position accuracy with time, in spite of mobility (and consequent dynamic topology). Observe that H-SPAWN and HC-EKF achieve similar performance asymptotically in time (i.e., given a sufficient number of available measurements), but H-SPAWN provides faster convergence.

We now analyze the performance achieved by individual nodes. Fig. 6d shows the number of visible satellites for the nodes in the network. Satellite visibility slightly changes with time due to mobility, but in general it is denied for most nodes. Fig. 6e and Fig. 6f, then, illustrate the evolution of RMSE with time for each node. A strong correlation can be observed between these results and Fig. 6d: RMSE convergence is faster for nodes with good satellite visibility, whereas inner nodes need more time slots because they may be several hops away from GPS-enabled peers. However, for all nodes H-SPAWN provides shorter convergence times than HC-EKF. Therefore H-SPAWN proves more effective in exploiting peer-to-peer information exchange than competing algorithms.

## VI. CONCLUSIONS

The problem of hybrid cooperative positioning for wireless networks has been addressed by proposing a novel, fully distributed approach based on iterative message passing on a factor graph model. The resulting H-SPAWN algorithm combines terrestrial ranging from neighboring peers and pseudorange from visible satellites, and provides an estimate of the *a posteriori* distributions of nodes' position. Simulation results show the superior performance of H-SPAWN with a joint position-bias representation compared to competing algorithms, such as least squares and Kalman filter.

As shown in Sec. V-B, H-SPAWN provides excellent performance even in challenging conditions, including large-scale networks, mobile nodes, low degrees of satellite visibility, and multi-hop peer-to-peer communication. Other advantages of H-SPAWN from a technical point of view are: (i) it is less sensitive to the initial guess, which, on the contrary, is very critical for HC-WLS, (ii) it is less likely to get stuck in local minima than both HC-WLS and HC-EKF, due to linearizations involved in these two algorithms, and (iii) it can be extended to non-Gaussian distributions and non-linear likelihood functions, whereas HC-EKF is intrinsically dependent on the Gaussian assumption.<sup>4</sup>

<sup>4</sup>H-SPAWN can be applied, in principle, to arbitrary noise statistics and likelihood functions; however, for an efficient parametric implementation, specific distribution families need to be developed on a case-by-case basis.

All the above-mentioned properties make the proposed algorithm a candidate for several military and civilian applications like rescue missions, operations in light/deep indoors, tracking of moving objects in GNSS-spotted areas.

## REFERENCES

- [1] E. Kaplan and C. Hegarty. *Understanding GPS: Principles and Applications*. Artech House Mobile Comm. Series, 2006.
- [2] J. M. Hasik and Michael R. Rip. "GPS at war: a ten-year retrospective", in *Proceedings of the 14th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS)*, Salt Lake City, UT, September 11–14, 2001, pp. 2406–2417.
- [3] G. Retscher and Q. Fu, "Integration of RFID, GNSS and DR for Ubiquitous Positioning in Pedestrian Navigation," in *Journal of Global Positioning Systems*, vol. 6, 2007, pp. 56–64.
- [4] C. Mensing, S. Sand, and A. Dammann, "GNSS Positioning in Critical Scenarios: Hybrid Data Fusion with Communications Signals," in *Proceedings of the International Workshop on Synergies in Communications and Localization*, Dresden, Germany, 2009.
- [5] A. Ihler, J. Fisher III, R. Moses, and A. Willsky, "Nonparametric belief propagation for self-calibration in sensor networks," in *Proc. of the 3rd Intl. Symposium on Information Processing in Sensor Networks*, 2004.
- [6] J. Chen, Y. Wang, C. Maa, and J. Chen, "Network-side Mobile Position Location Using Factor Graphs," in *IEEE Transactions on Wireless Communications*, vol. 5, 2006, pp. 2696–2704.
- [7] H. Wymeersch, J. Lien, and M.Z. Win, "Cooperative localization in wireless networks," in *Proceedings of the IEEE*, vol. 97, no. 2, Feb. 2009, pp. 427–450.
- [8] M.A. Caceres, F. Penna, H. Wymeersch and R. Garello, "Hybrid GNSS-terrestrial Cooperative Positioning via Distributed Belief Propagation" in *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, Miami, Florida, December 6–10, 2010, pp. 1–5.
- [9] F.R. Kschischang, B.J. Frey, H. A. Loeliger, "Factor Graphs and the Sum-Product Algorithm," in *IEEE Transactions on Information Theory*, vol. 47, no. 2, Feb. 2001.
- [10] J.-C. Chen, C.-S. Maa, Y.-C. Wang, and J.-T. Chen, "Mobile Position Location Using Factor Graphs," *IEEE Communications Letters*, vol. 7, pp. 431–433, September 2003.
- [11] C. Mensing and S. Plass, "Positioning Based on Factor Graphs," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, pp. 1–12, September 2007.
- [12] H. Q. Liu, F. K. W. Chan, and H. C. So, "Non-Line-of-Sight Mobile Positioning Using Factor Graphs," *IEEE Transactions on Vehicular Technology*, pp. 1–6, 2009.
- [13] A. T. Ihler, "Inference in Sensor Networks: Graphical Models and Particle Methods," PhD Thesis, MIT, 2005.
- [14] G. Heinrichs, P. Mulassano, and F. Dovis, "A hybrid positioning algorithm for cellular radio networks by using a common rake receiver," in *Symp. on Personal, Indoor and Mobile Radio*, 2004, pp. 2347–2351.
- [15] C. Falsi, D. Dardari, L. Mucchi, and M. Win, "Time of arrival estimation for UWB localizers in realistic environments," in *EURASIP Journal of Applied Signal Processing*, 2006:Article ID 32 082, 1–13.
- [16] H. Hashemi, "The indoor radio propagation channel," in *Proceedings of the IEEE*, vol. 81 pp. 943–968, July 1993.
- [17] B. Ristic and S. Arulampalam, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House, 2004.
- [18] J. Lien, J. Ferner, W. Srichavengsup, H. Wymeersch, M. Win, "Practical Cooperative Localization and Tracking," submitted to *IEEE Transactions on Wireless Communications*, 2009. Available at [http://dl.dropbox.com/u/14551660/LieFerSriWymWinv11\\_jl.pdf](http://dl.dropbox.com/u/14551660/LieFerSriWymWinv11_jl.pdf)
- [19] M.A. Caceres, F. Sottile, R. Garello and M.A. Spirito "Hybrid GNSS-ToA Localization and Tracking via Cooperative Unscented Kalman Filter" in *Proceedings of the IEEE 21st International Conference on Personal, Indoor and Mobile Radio Communications (PIMRC) Workshops*, Istanbul, Turkey, September 26–29 2010, pp. 271–275.
- [20] F. Penna, M.A. Caceres, and H. Wymeersch, "Cramér-Rao Bound for Hybrid Peer-to-Peer Positioning," *IEEE Communication Letters*, vol. 14, no. 11, pp. 1005–1007, 2010.
- [21] J. Broch, D.A. Maltz, D.B. Johnson, Y.C. Hu, J. Jetcheva J, "A performance comparison of multi-hop wireless ad hoc network routing protocols", *Proc. 4th ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom98)*, October 1998.
- [22] K. Kuusniemi, G. Lachapelle, and J. Takala, "Position and Velocity Reliability Testing in Degraded GPS Signal Environments," *GPS Solutions*, pp. 226–237, October 2004.