*(Article begins on next page)*

# Hybrid GNSS-terrestrial Cooperative Positioning via Distributed Belief Propagation

Mauricio A. Caceres[1], Federico Penna[1], Henk Wymeersch[2], Roberto Garello[1]
[1]Politecnico di Torino - Dept. of Electronics, Turin, Italy
[2]Chalmers University of Technology - Dept. of Signals and Systems, Gothenburg, Sweden
Email: {mauricio.caceresduran, federico.penna, roberto.garello}@polito.it, henkw@chalmers.se

***Abstract*—Cooperative positioning algorithms have been recently introduced to overcome the limitations of traditional methods, relying on GNSS or other terrestrial infrastructure. In particular, SPAWN (Sum-Product Algorithm over a Wireless Network) was shown to provide accurate position estimate even in challenged indoor environments, thanks to exchange of local information among peers based on terrestrial ranging. In this paper we extend the SPAWN framework by considering a hybrid scenario, where agents combine satellite and peer-to-peer terrestrial measurements. The novel hybrid SPAWN (H-SPAWN) approach allows increased availability and robustness compared to GNSS-only positioning in light and deep indoor scenarios, while keeping the advantages of a distributed implementation of the original SPAWN. A parametric message representation is proposed to reduce the communication overhead, and to improve the estimation accuracy. Simulation results show that the proposed solution outperforms traditional algorithms such as cooperative least squares and the extended Kalman filter.**

## I. Introduction

Cooperative positioning methods, based on exchange of information among peers are receiving a great attention. Some methods using belief propagation (BP) have been proposed in [1], [2], [3]. They have been especially designed for wireless ranging systems operating in critical environments where GNSS is not available. However, such cooperative schemes can be also used in combination with GNSS, to improve positioning availability and accuracy in cases where pseudorange measurements are available intermittently, or from a limited number of satellites, or are strongly affected by noise/errors. Hybrid cooperative positioning schemes can thus be designed that fuse information from peers and from GNSS satellites.

In this paper we propose an efficient hybrid positioning method, based on distributed BP, applying the sum product algorithm (SPA) over a wireless network, similarly as it was done in [1] for the peer-to-peer case. The new algorithm, called H-SPAWN (hybrid sum product algorithm over a wireless network), can be implemented in a fully distributed fashion through local exchange of messages between pairs of neighboring nodes. Compared to original SPAWN, an extra variable – the bias with respect to satellite clocks that affects pseudoranges – must be estimated along with nodes' positions.

The rest of the paper is organized as follows: Sec. II provides a mathematical description of the problem; in Sec. III we develop the factor graph (FG) model for hybrid positioning; in Sec. IV we focus on the implementation of H-SPAWN and propose an efficient parameter-based message representation; in Sec. V we test H-SPAWN via numerical simulations and compare its performance to that of competing algorithms; Finally Sec. VI concludes.

*Comment:* Sections II-B and III assume a background knowledge in graphical models and Bayesian inference. The reader unfamiliar with these topics may consult [1], [3], [4].

## II. Problem Formulation

### A. Model

Consider a network of $M$ agents and $S$ satellites. Let $\mathcal{M}$ be the set of agents and $\mathcal{S}$ the set of satellites. Referring to a particular agent $m \in \mathcal{M}$, denote by $\mathcal{M}_m$ the subset of peers it can communicate with ("neighbors") and by $\mathcal{S}_m$ the subset of satellites it can see. Position variables are denoted by $\boldsymbol{x}_i$ (where $i$ may be either a satellite or an agent). Our focus will be 2-dimensional positions, as the extension to the 3-dimensional case is conceptually straightforward. The clock bias of node $m$ is denoted by $b_m$ and expressed in distance units. Therefore, the state of each node $m$ is identified by:

$$\tilde{\boldsymbol{x}}_m \triangleq [\boldsymbol{x}_m \; b_m]. \tag{1}$$

In the considered hybrid scenario, two types of measurements are performed by nodes: (i) *range* measurements, i.e., distance between peers:

$$r_{nm} = \|\boldsymbol{x}_n - \boldsymbol{x}_m\| + v_{nm}, \tag{2}$$

and (ii) *pseudorange* measurements, i.e., measured distance from satellites:

$$\rho_{sm} = \|\boldsymbol{x}_s - \boldsymbol{x}_m\| + b_m + v_{sm}, \tag{3}$$

where the symbol $\|\cdot\|$ denotes Euclidean distance, $m, n \in \mathcal{M}$, $s \in \mathcal{S}$, $v_{nm}$ and $v_{sm}$ are measurement noise. Notice that pseudorange measurements are affected by the additional unknown $b_m$, that is one of the variables to be estimated[1]. We introduce the following vector notation to group together different nodes' variables: $\boldsymbol{X} \triangleq \{\boldsymbol{x}_{m\in\mathcal{M}}\}$; $\boldsymbol{b} \triangleq \{b_{m\in\mathcal{M}}\}$; $\tilde{\boldsymbol{X}} \triangleq \{\tilde{\boldsymbol{x}}_{m\in\mathcal{M}}\}$; $\boldsymbol{r}_m \triangleq \{r_{nm} \; \forall n \in \mathcal{M}_m\}$; $\boldsymbol{R} \triangleq \{\boldsymbol{r}_{m\in\mathcal{M}}\}$; $\boldsymbol{\rho}_m \triangleq \{\rho_{sm} \; \forall s \in \mathcal{S}_m\}$; $\boldsymbol{P} \triangleq \{\boldsymbol{\rho}_{m\in\mathcal{M}}\}$.

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

$$p(\tilde{\boldsymbol{x}}_m^{(t)} | \boldsymbol{R}^{(1:t)}, \boldsymbol{P}^{(1:t)}) \quad \forall m \in \mathcal{M}. \tag{4}$$

[1]On the contrary, terrestrial ranges are typically estimated with methods avoiding bias problem, like round-trip time of arrival or RSS measurements.

### B. Assumptions

We will make the following assumptions, which hold approximately in many practical scenarios.

- **A1:** Peer-to-peer measurement noise samples are independent Gaussian, with symmetric link variance (assumed as known by both nodes):
$$v_{mn}, v_{nm} \sim \mathcal{N}\left(0,\, \sigma^2_{mn}\right). \tag{5}$$
- **A2:** Satellite measurement noise samples are independent Gaussian with:
$$v_{sm} \sim \mathcal{N}\left(0,\, \sigma^2_{sm}\right). \tag{6}$$
- **A3:** Nodes' mobility is assumed as Markovian and mutually independent:
$$p\Big(\tilde{\boldsymbol{X}}^{(t)}\Big|\tilde{\boldsymbol{X}}^{(0:t-1)}\Big)=p\Big(\tilde{\boldsymbol{X}}^{(t)}\Big|\tilde{\boldsymbol{X}}^{(t-1)}\Big)=\prod_{m\in\mathcal{M}} p\Big(\tilde{\boldsymbol{x}}^{(t)}_m\Big|\tilde{\boldsymbol{x}}^{(t-1)}_m\Big). \tag{7}$$
- **A4:** Measurement likelihood depends only on the current state and can be split into two factors, since range and pseudorange measurements are independent:
$$\begin{aligned} p\left(\tilde{\boldsymbol{R}}^{(t)}, \tilde{\boldsymbol{P}}^{(t)}\Big|\tilde{\boldsymbol{X}}^{(0:t)}\right) &= p\left(\boldsymbol{R}^{(t)},\, \boldsymbol{P}^{(t)}\Big|\tilde{\boldsymbol{X}}^{(t)}_m\right) \\ &= p\left(\boldsymbol{R}^{(t)}\Big|\tilde{\boldsymbol{X}}^{(t)}\right) p\left(\boldsymbol{P}^{(t)}\Big|\tilde{\boldsymbol{X}}^{(t)}\right). \end{aligned} \tag{8}$$

## III. Bayesian Inference on Factor Graph

Due to our assumptions, we can compute the *a posteriori* distribution (4) at each time slot recursively (similarly to [1]):
$$\begin{aligned} p\left(\tilde{\boldsymbol{x}}^{(t)}_m\Big|\boldsymbol{R}^{(1:t)}, \boldsymbol{P}^{(1:t)}\right) &= \int p\left(\boldsymbol{R}^{(t)}, \boldsymbol{P}^{(t)}\Big|\tilde{\boldsymbol{x}}^{(t)}_m, \tilde{\boldsymbol{X}}^{(t)}_{\sim m}\right)\times \\ &\prod_{n\in\mathcal{M}} p\left(\tilde{\boldsymbol{x}}^{(t)}_n\Big|\tilde{\boldsymbol{x}}^{(t-1)}_n\right) p\left(\tilde{\boldsymbol{x}}^{(t-1)}_n\Big|\boldsymbol{R}^{(1:t-1)}, \boldsymbol{P}^{(1:t-1)}\right) d\tilde{\boldsymbol{X}}^{(t)}_{\sim m}. \end{aligned} \tag{9}$$
where $\tilde{\boldsymbol{X}}^{(t)}_{\sim m}$ denotes all state vectors at time slot $t$ except $\tilde{\boldsymbol{x}}_m$.

Hence, given $p(\tilde{\boldsymbol{x}}^{(t-1)}_n|\boldsymbol{R}^{(1:t-1)}, \boldsymbol{P}^{(1:t-1)})$ $\forall n$, we create a factor graph of $p(\boldsymbol{R}^{(t)}, \boldsymbol{P}^{(t)}|\tilde{\boldsymbol{x}}^{(t)}_m, \tilde{\boldsymbol{X}}^{(t)}_{\sim m})\times \prod_{n\in\mathcal{M}} p(\tilde{\boldsymbol{x}}^{(t)}_n|\tilde{\boldsymbol{x}}^{(t-1)}_n)$ – shown in Fig. 1 – taking into account both the evidence (given by measurements likelihood) and state temporal evolution (according to a mobility model). Vertices on top have as downward messages $p(\tilde{\boldsymbol{x}}^{(t-1)}_n|\boldsymbol{R}^{(1:t-1)}, \boldsymbol{P}^{(1:t-1)})$, so that performing the SPA on this FG leads to approximations of (4).

Thanks to A1-A4, (9) can be factorized as follows:
$$\prod_{m\in\mathcal{M}}\left[f_m\left(\tilde{\boldsymbol{x}}^{(t)}_m, \tilde{\boldsymbol{x}}^{(t-1)}_m\right)\prod_{\substack{n\in\mathcal{M}_m\\ n<m}} h_{nm}\left(\tilde{\boldsymbol{x}}^{(t)}_m, \tilde{\boldsymbol{x}}^{(t)}_n\right)\prod_{s\in\mathcal{S}_m} g_{sm}\left(\tilde{\boldsymbol{x}}^{(t)}_m\right)\right], \tag{10}$$
where $f_m\left(\tilde{\boldsymbol{x}}^{(t)}_m, \tilde{\boldsymbol{x}}^{(t-1)}_m\right) \equiv p\left(\tilde{\boldsymbol{x}}^{(t)}_m\Big|\tilde{\boldsymbol{x}}^{(t-1)}_m\right)$ represents temporal evolution, $h_{nm}\left(\tilde{\boldsymbol{x}}^{(t)}_m, \tilde{\boldsymbol{x}}^{(t)}_n\right) \equiv p\left(r_{nm}\Big|, \tilde{\boldsymbol{x}}^{(t)}_m\, \tilde{\boldsymbol{x}}^{(t)}_n\right)$ represents the range measurement likelihood given the positions of nodes $m$ and $n$, and $g_{sm}\left(\tilde{\boldsymbol{x}}^{(t)}_m\right) \equiv p\left(\rho_{sm}\Big|\tilde{\boldsymbol{x}}^{(t)}_m\right)$ represents the pseudorange measurement likelihood given the state (position-bias) of node $m$. The resulting FG representation is depicted in Fig. 1. The marginal posteriors at each time slot

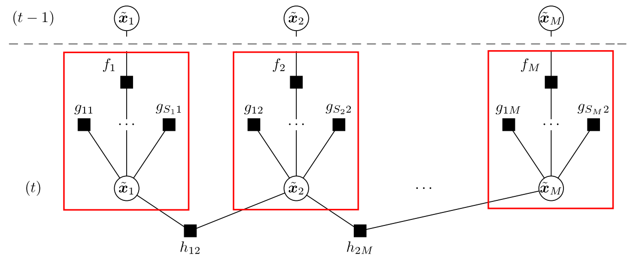



Figure 1. FG model for hybrid cooperative positioning. Red boxes represent physical nodes (i.e., factors inside a box are computed internally by a node). Instead, factors connected to pairs of nodes imply message exchanges. This representation allows a direct mapping of the FG onto the physical network, hence a distributed implementation.

(4) can be estimated in a distributed manner by executing SPA, similarly as in [1], leading to an algorithm we name HSPAWN.

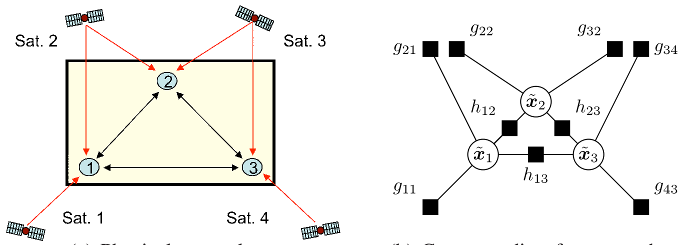
## IV. Parametric Belief Propagation in H-SPAWN

### A. SPA and H-SPAWN

In the proposed H-SPAWN algorithm, nodes exchange messages according to the well-known belief propagation update rules [4], [5]. Letting $\mathcal{F}_m$ be the set of factors connected to variable node $m$ and $\mathcal{V}_f$ the set of variables connected to factor $\varphi$, messages from $m$ to $\varphi$ are of the form
$$\mathcal{M}_{m\to\varphi}(\tilde{\boldsymbol{x}}_m) = \prod_{h\in\mathcal{F}_m\setminus\varphi} \mathcal{M}_{h\to m}(\tilde{\boldsymbol{x}}_m), \tag{11}$$
and messages from $\varphi$ to $m$ are
$$\mathcal{M}_{\varphi\to m}(\tilde{\boldsymbol{x}}_m) = \int \varphi\left(\tilde{\boldsymbol{x}}_m, \{\tilde{\boldsymbol{x}}_j\}_{j\in\mathcal{V}_f\setminus m}\right)\prod_{j\in\mathcal{V}_f\setminus m}\mathcal{M}_{j\to\varphi}(\tilde{\boldsymbol{x}}_j)\, \mathrm{d}\{\sim\tilde{\boldsymbol{x}}_m\}. \tag{12}$$
where the notation $\int \ldots \mathrm{d}\{\sim\tilde{\boldsymbol{x}}_m\}$ denotes integration over all the variables involved in $\varphi$ except $\tilde{\boldsymbol{x}}_m$.

Since in the considered model factors are connected to one or, at most, two variables, the above expression simplifies to the factor itself for satellite factors ($\mathcal{M}_{g_{sm}\to m}(\tilde{\boldsymbol{x}}_m) = g_{sm}(\tilde{\boldsymbol{x}}_m)$) and for temporal or peer-to-peer factors the product in (12) contains one term only. This message passing scheme can be directly mapped on the network nodes, making possible a *distributed implementation*. Messages from satellite factors ($\mathcal{M}_{g_{sm}\to m}$) are computed by node $m$ based on the data received from the satellite; temporal messages ($\mathcal{M}_{f_m\to m}$) are computed internally by node $m$; peer-to-peer messages ($\mathcal{M}_{h_{mn}\to n}$) involve actual messages (i.e., packets over the network) passed from node $m$ to $n$, and are computed by $n$ based on the information received from $m$ ($r_{mn}$ and $\hat{\boldsymbol{x}}_m$).

As the messages are functions of continuous variables, care must be taken in the message representation. We have chosen a parametric message representation in H-SPAWN, since it has lower computational and communication requirements compared to a non-parametric (sample-based) approach. All beliefs and messages are thus approximated by known probability distribution families, therefore they can be represented by the parameters of each family. In the remainder of this section, we describe these families and how (11)-(12) can be computed for those families.



(a) Physical network.

(b) Corresponding factor graph.

Figure 2. Simulation scenario.

### B. Distribution Families

1) *Beliefs* of position-bias variables $\tilde{\boldsymbol{x}}$ are approximated by multivariate Gaussian distributions $\mathcal{N}_{\tilde{\boldsymbol{x}}}\left(\boldsymbol{\mu}_{\tilde{\boldsymbol{x}}},\, \Sigma_{\tilde{\boldsymbol{x}}}\right)$, whose parameters are mean $\boldsymbol{\mu}_{\tilde{\boldsymbol{x}}} = \begin{bmatrix}\boldsymbol{\mu}_{\boldsymbol{x}} & \mu_b\end{bmatrix}$ and covariance matrix $\Sigma_{\tilde{\boldsymbol{x}}}$. The p.d.f. is of the form

$$\frac{1}{\sqrt{8\pi^3 |\Sigma_{\tilde{\boldsymbol{x}}}|}} \exp\left[-\frac{1}{2}\left(\tilde{\boldsymbol{x}} - \boldsymbol{\mu}_{\tilde{\boldsymbol{x}}}\right)^T \Sigma_{\tilde{\boldsymbol{x}}}^{-1}\left(\tilde{\boldsymbol{x}} - \boldsymbol{\mu}_{\tilde{\boldsymbol{x}}}\right)\right]. \quad (13)$$

2) *Peer-to-peer messages* are represented by "cylindrical distributions" $\mathcal{C}_{\tilde{\boldsymbol{x}}}\left(\varrho,\, \boldsymbol{\mu}_{\boldsymbol{x}},\, \sigma_\varrho^2\right)$, characterized by radius $\varrho$, center $\boldsymbol{\mu}_{\boldsymbol{x}}$ (position of peer), variance $\sigma_\varrho^2$. This family is similar to the $\mathcal{D}$ distribution introduced in [6], but with uniform bias probability (inside a certain interval) since peer-to-peer messages do not carry any information about bias. The p.d.f. is of the form

$$\frac{1}{Z_{\mathcal{C}}} \exp\left[-\frac{1}{2\sigma_\varrho^2}\left(\|\boldsymbol{x} - \boldsymbol{\mu}_{\boldsymbol{x}}\| - \varrho\right)^2\right], \quad (14)$$

where $Z_{\mathcal{C}}$ is a normalizing constant.

3) *Satellite messages* involve the $b$ component as well and are therefore represented by a "conic distribution" family $\mathcal{V}_{\tilde{\boldsymbol{x}}}\left(\varrho,\, \boldsymbol{\mu}_{\boldsymbol{s}},\, \sigma_\varrho^2\right)$, with radius $\varrho$, center $\boldsymbol{\mu}_{\boldsymbol{s}}$ (satellite position) and variance $\sigma_\varrho^2$. The p.d.f. is of the form

$$\frac{1}{Z_{\mathcal{V}}} \exp\left[-\frac{1}{2\sigma_\varrho^2}\left(\|\boldsymbol{x} - \boldsymbol{\mu}_{\boldsymbol{s}}\| + b - \varrho\right)^2\right], \quad (15)$$

where again $Z_{\mathcal{V}}$ is a normalization constant.

### C. Message Filtering

The integration in (12) involves "filtering" the incoming message with the factor itself. Three cases can be distinguished.

1) *Messages from temporal factors* are computed within agents and propagate the beliefs $\tilde{\boldsymbol{x}}$ at time $t-1$ to time $t$. Position update can be determined according to a predefined mobility model, while bias update can take into account a clock drift model. Based on these two models, a new position $\boldsymbol{\mu}_{\tilde{\boldsymbol{x}}^{(t)}}$ is estimated based on $\boldsymbol{\mu}_{\tilde{\boldsymbol{x}}^{(t-1)}}$. In addition, since every prediction carries some uncertainty, we assume that the variance is increased by some function (e.g., linear with the elapsed time), such that $\Sigma_{\tilde{\boldsymbol{x}}_m^{(t)}} \succeq \Sigma_{\tilde{\boldsymbol{x}}_m^{(t-1)}}$. The temporal message is then defined as

$$\mathcal{M}_{f_m \to m}\left(\tilde{\boldsymbol{x}}_m^{(t)}\right) := \mathcal{N}_{\tilde{\boldsymbol{x}}_m^{(t)}}\left(\boldsymbol{\mu}_{\tilde{\boldsymbol{x}}_m^{(t)}},\, \Sigma_{\tilde{\boldsymbol{x}}_m^{(t)}}\right). \quad (16)$$

(With a slight abuse of notation, symbol $:=$ means that the message is the p.d.f. of the considered distribution).

2) *Messages from satellite factors* are computed by agents, based on a satellite position and pseudorange. These messages belong to the conic distribution $\mathcal{V}$:

$$\mathcal{M}_{g_{sm} \to m}^{(t)}\left(\tilde{\boldsymbol{x}}_m^{(t)}\right) := \mathcal{V}_{\tilde{\boldsymbol{x}}_m^{(t)}}\left(\rho_{sm}^{(t)},\, \boldsymbol{x}_s^{(t)},\, \sigma_{\rho_{sm}^{(t)}}^2\right). \quad (17)$$

3) *Messages from peer-to-peer factors* are computed by agents, based on peer-to-peer ranges and peer information. These messages belong to the cylindrical distribution $\mathcal{C}$:

$$\mathcal{M}_{h_{nm} \to m}^{(t)}\left(\tilde{\boldsymbol{x}}_m^{(t)}\right) := \mathcal{C}_{\tilde{\boldsymbol{x}}_m^{(t)}}\left(r_{nm}^{(t)}, \boldsymbol{\mu}_{\boldsymbol{x}_n^{(t)}},\, \sigma_{r_{nm}^{(t)}}^2 + \operatorname{tr}\Sigma_{\boldsymbol{x}_n^{(t)}}\right), \quad (18)$$

where for simplicity the uncertainty of the peer's position is assumed circular with a radial variance equal to the sum of the variances of axes.

### D. Message Multiplication

Message multiplication is used both for belief marginalization and computation of messages from variable nodes to factor nodes (11). Due to the different shapes of the incoming messages, the multiplication is approximated as a multivariate Gaussian distribution without any restriction on the covariance matrix, so it can take any ellipsoidal shape:

$$\mathcal{M}_{m \to \varphi}(\tilde{\boldsymbol{x}}_m^{(t)}) := \mathcal{N}_{\tilde{\boldsymbol{x}}_m^{(t)}}\left(\boldsymbol{\mu}_{\tilde{\boldsymbol{x}}_m^{(t)}},\, \Sigma_{\tilde{\boldsymbol{x}}_m^{(t)}}\right), \quad (19)$$

$$\hat{p}\left(\tilde{\boldsymbol{x}}_m^{(t)}\right) := \mathcal{N}_{\tilde{\boldsymbol{x}}_m^{(t)}}\left(\boldsymbol{\mu}_{\tilde{\boldsymbol{x}}_m^{(t)}},\, \Sigma_{\tilde{\boldsymbol{x}}_m^{(t)}}\right). \quad (20)$$

The problem is then reverts to finding the parameters of the chosen output distribution that best approximate the product of the incoming parametric messages. This is achieved by importance sampling, sample mean and variance estimators, as described in Alg. 1.

**Algorithm 1** Parametric Message Multiplication

**Require:** Initial estimate of $\hat{\boldsymbol{\mu}}_{\tilde{\boldsymbol{x}}},\, \hat{\Sigma}_{\tilde{\boldsymbol{x}}}$.

**repeat**
  Draw $N$ samples $\tilde{\boldsymbol{x}}_k$ from $\mathcal{N}\left(\hat{\boldsymbol{\mu}}_{\tilde{\boldsymbol{x}}},\, \hat{\Sigma}_{\tilde{\boldsymbol{x}}}\right)$.
  Compute the probability of each sample $q\left(\tilde{\boldsymbol{x}}_k\right)$ in the distribution it was drawn from, using (13).
  Evaluate the p.d.f. of each message in the multiplication at the given samples $p_i\left(\tilde{\boldsymbol{x}}_k\right)$ using Eqs. (13), (14), (15).
  Assign a weight to each sample as: $w_k = \frac{\prod_i p_i(\tilde{\boldsymbol{x}}_k)}{q(\tilde{\boldsymbol{x}}_k)}$, then normalize them such that $\sum_{k=1}^N w_k = 1$.
  Estimate new parameters with the weighted sample mean and covariance estimators:
  $\hat{\boldsymbol{\mu}}_{\tilde{\boldsymbol{x}}} = \sum_{k=1}^N w_k \boldsymbol{x}_k$,
  $\hat{\Sigma}_{\tilde{\boldsymbol{x}}} = \frac{\sum_{k=1}^N w_k (\boldsymbol{x}_k - \hat{\boldsymbol{\mu}}_{\tilde{\boldsymbol{x}}})(\boldsymbol{x}_k - \hat{\boldsymbol{\mu}}_{\tilde{\boldsymbol{x}}})^T}{1 - \sum_{k=1}^N w_k^2}$.
**until** convergence
**return** $\hat{\boldsymbol{\mu}}_{\tilde{\boldsymbol{x}}},\, \hat{\Sigma}_{\tilde{\boldsymbol{x}}}$.

## V. Simulation Results

### A. Simulation Setup

To test H-SPAWN performance we use the scenario depicted in Fig. 2 (a). Each of the three agents sees two satellites and can communicate with all other peers. The corresponding FG is depicted in Fig. 2 (b). Clearly, none of the agents would be able to localize itself without peer-to-peer information.

Agents are placed randomly in an area of $100\,\mathrm{m} \times 100\,\mathrm{m}$; their clock bias values are random as well, drawn from a uniform distribution between $-10$ m and 10 m. Satellites are placed at distances on the order of $20.000$ km from the agents. Pseudorange measurement noise standard deviation is $\sigma_{sm} = 4$ m $\forall s, m$, while range measurement noise has a $\sigma_{nm} = 20$ cm standard deviation $\forall n, m$. All initial beliefs are set to Gaussian distributions centered in the origin with very large variances. The scenario is static (i.e., agents do not move), hence temporal factor updates (16) are as follows: $\boldsymbol{\mu}_{\tilde{\boldsymbol{x}}_m^{(t)}} = \boldsymbol{\mu}_{\tilde{\boldsymbol{x}}_m^{(t-1)}}$ (positions are kept constant), $\Sigma_{\tilde{\boldsymbol{x}}_m^{(t)}} = \Sigma_{\tilde{\boldsymbol{x}}_m^{(t-1)}} + \boldsymbol{I}$ (position and bias uncertainty is increased by 1 m through the identity matrix $\boldsymbol{I}$). At each time slot, new measurements are generated and H-SPAWN is run until convergence is reached in the given slot. The number of iterations needed depends on the network size and topology. In the example of Fig. 2, convergence is reached after the second iteration.

H-SPAWN performance is compared to two other cooperative positioning approaches. As a non-Bayesian approach, we consider cooperative least squares (CLS), implemented according to the iterative descent algorithm proposed in [1] and extended to the hybrid GNSS and terrestrial ranging like in [7]. As a Bayesian approach, we consider the hybrid extended Kalman filter (EKF) algorithm presented in [8]. To make EKF consistent with CLS and H-SPAWN, peer position variances are summed to range measurement variances – as it is done in H-SPAWN, eq. (18) – and the mobility model is the same as in H-SPAWN (Sec. IV-C).

### B. Performance Comparison

Fig. 3 depicts the convergence of position and bias estimates for the three algorithms, with the network configuration of Fig. 2. Faster convergence of H-SPAWN can be appreciated compared to CLS and EKF. Also, the estimated covariances of H-SPAWN always contain the true value inside the $\pm 3\sigma$ interval, whereas the EKF tends to be too optimistic. CLS, on the contrary, does not provide any information to evaluate the estimation uncertainty.

For a quantitative performance comparison, Fig. 4 shows the CDFs of positions and bias error – computed as the difference between true value and (mean) estimated value – for the 3 algorithms, averaged over $100$ Monte Carlo simulations of $10$ time slots each. At every Monte Carlo run, a new network topology is created based on the same scenario of Fig. 2, with random agents' positions and biases in the given range.

Each panel shows error CDFs after 1 and 10 slots. In both cases, H-SPAWN turns out to outperform CLS and EKF, and a remarkable gap can be seen especially after $1$ slot. This result means that H-SPAWN requires significantly less measurements to provide an accurate estimation.

Other advantages of H-SPAWN are: (i) it is not sensitive to the initial guess (which, on the contrary, is very critical for CLS), (ii) is less likely to get stuck in local minima than both CLS and EKF, and (iii) can be extended to non-Gaussian distributions and non-linear likelihood functions (whereas EKF is intrinsically dependent on the Gaussian assumption and linearization of the state and/or measurement equations).

### C. Complexity

The complexity of H-SPAWN is dominated by message multiplication (11). For an agent with its time message, $M$ peer-to-peer messages and $S$ satellite-to-peer messages, using $N$ samples to represent its distribution and requiring $I$ iterations in Algorithm 1, the complexity scales as $\mathcal{O}\left(N(M+S+1)I\right)$. In contrast, the filtering step (12) can be performed analytically in $\mathcal{O}(M+S+1)$.

## VI. Conclusion

The problem of hybrid positioning for wireless networks has been addressed in this paper by proposing a novel, distributed approach based on iterative message passing on a factor graph model. The resulting H-SPAWN algorithm, which extends the previous SPAWN proposed in [1] for peer-to-peer positioning, combines terrestrial ranging from neighboring peers and pseudoranging from visible satellites, and provides an estimation of the *a posteriori* probability of the state (position and clock bias) of each node. Simulation results show the improved performance of H-SPAWN compared to competing algorithms, such as cooperative least squares and extended Kalman filter.

## Acknowledgment

This work was funded by European Space Agency (ESA) project "Peer to Peer Positioning" (contract number 286/09/NL/AT). The authors are grateful to Jaron Samson for his support in this research.

## References

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

[2] J. Chen, Y. Wang, C. Maa, and J. Chen, "Network-side Mobile Position Location Using Factor Graphs," *IEEE Transactions on Wireless Communications*, vol. 5, 2006, pp. 2696-2704.

[3] A. Ihler, J. FisherIII, R. Moses, and A. Willsky, "Nonparametric belief propagation for self-calibration in sensor networks," *Proc. 3rd International Symposium on Information Processing in Sensor Networks*, 2004.

[4] F. Kschischang, B. Frey, H. Loeliger, "Factor Graphs and the Sum-Product Algorithm," *IEEE Trans. on Information Theory*, vol. 47, no. 2, 2001.

[5] H. Loeliger, "An introduction to factor graphs," *IEEE Signal Processing Magazine*, vol. 21, 2004, pp. 28-41.

[6] J. Lien, Ulric. J. Ferner, W. Srichavengsup, H. Wymeersch, Moe Z. Win, "Practical Cooperative Localization and Tracking," submitted to *IEEE Trans. on Wireless Communications*, 2009.

[7] G. Heinrichs, P. Mulassano, and F. Dovis, "A hybrid positioning algorithm for cellular radio networks by using a common rake receiver," *Symposium on Personal, Indoor and Mobile Radio*, 2004, pp. 2347-2351.

[8] C. Mensing, S. Sand, and A. Dammann, "GNSS Positioning in Critical Scenarios: Hybrid Data Fusion with Communications Signals," *Proc. of the International Workshop on Synergies in Communications and Localization (SyCoLo)*, Dresden, Germany, 2009.

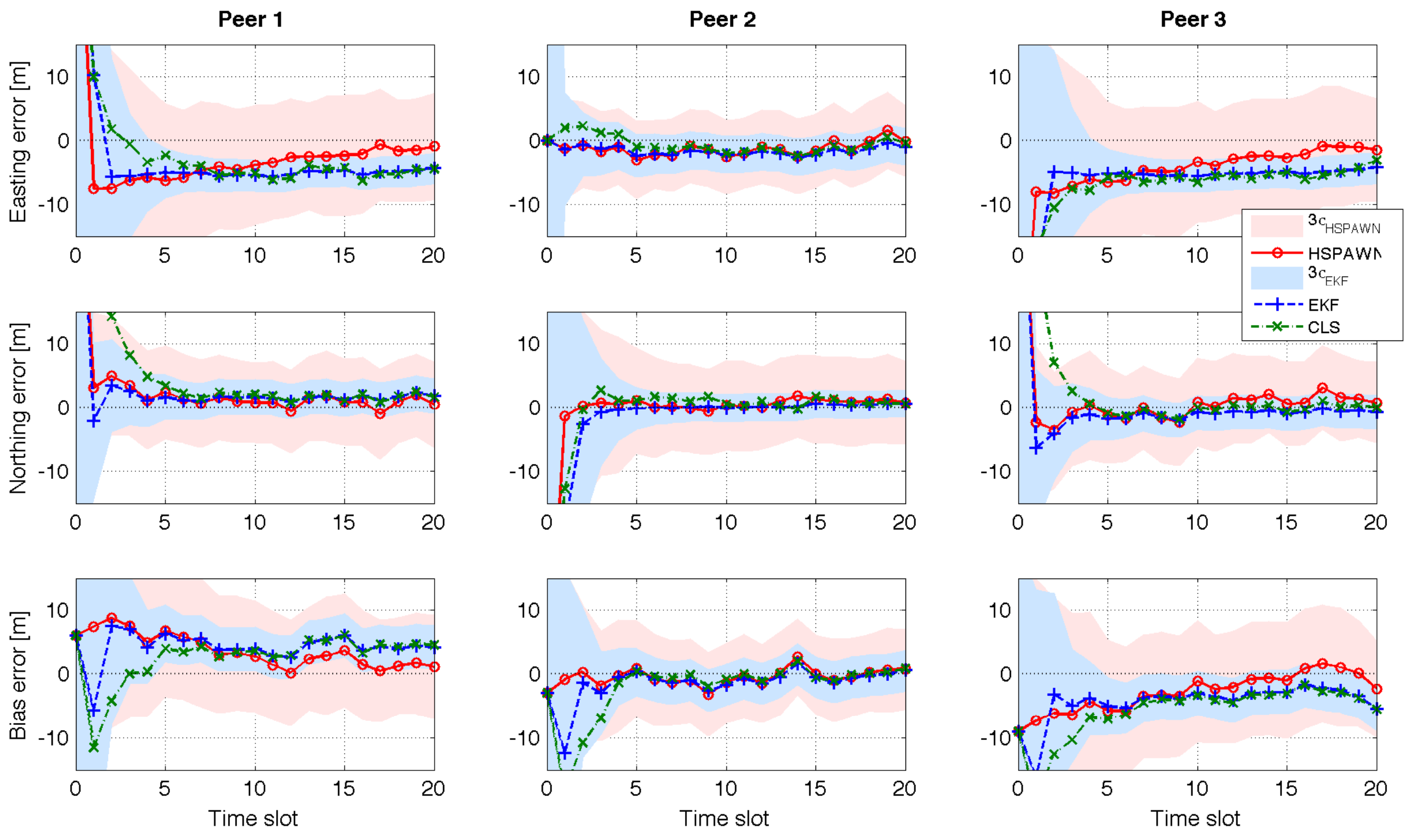



Figure 3. Convergence of H-SPAWN, Cooperative LS and EKF. The plots show, for each time slot, the value obtained at convergence (2 iterations in the considered network).

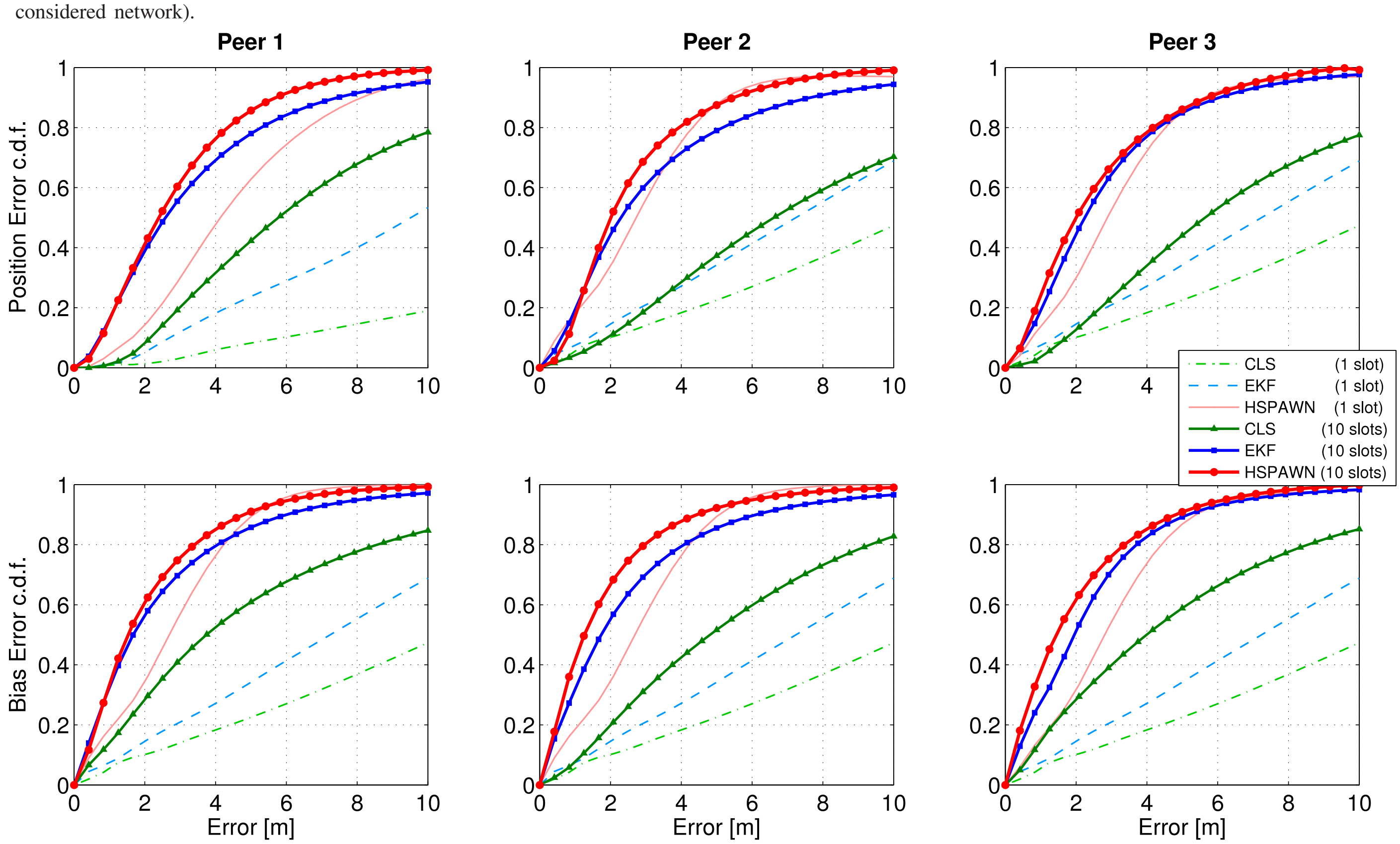



Figure 4. Cumulative density functions of H-SPAWN, CLS, and EKF position and bias errors averaged over 100 Monte Carlo runs with different positions and biases of agents. Light lines = error after 1 time slot; bold lines = error after 10 time slots.