Signal Processing 89 (2009) 1144–1154

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Sensor self-localization with beacon position uncertainty

Mahesh Vemula *, Mónica F. Bugallo, Petar M. Djurić

Department of Electrical and Computer Engineering, Stony Brook University, Stony Brook, NY 11794, USA

ARTICLE INFO

Article history:
Received 14 May 2007
Received in revised form
15 December 2008
Accepted 19 December 2008
Available online 1 January 2009

Keywords:
Localization
Least squares
Monte Carlo methods
Cramér–Rao bounds

ABSTRACT

We propose algorithms for distributed sensor self-localization using beacon nodes. These beacon nodes broadcast some information which describes their positions. The sensor nodes with unknown location information utilize these descriptions along with the characteristics of received signals to obtain estimates of their positions. Sensors with resolved positions, in the successive stages of the algorithm also broadcast their location information to other sensors so that they can resolve their own positions. Conditional upon the availability of probabilistic distributions of noise processes, we propose iterative and Monte Carlo sampling-based methods for obtaining sensor location descriptions. We also provide approximate hybrid Cramér–Rao bounds for distributed sensor self-localization and compare them with the proposed algorithms. We demonstrate the performance of the proposed algorithms through extensive computer simulations.

1. Introduction and motivation

Wireless sensor networks play a vital role towards pervasive sensing and computing with minimal human supervision. Recent interest in sensor networks has been primarily due to improvements in their performance and decrease in their cost. As these sensors are inexpensive, tiny and untethered, they can be deployed in large numbers. Some civilian applications of sensor networks include monitoring systems in vineyards, environmental habitats, community areas, traffic highways and smart bridges [1]. In military applications, sensors networks are used for surveillance of armed troops and vehicles in battlefields and detection, tracking and classification of enemy targets [1,2].

In many sensor networking applications such as object tracking, sensors measure signals that are functions of the geometry between the object under surveillance and the sensor. Therefore, to extract meaningful information of the object's dynamics, knowledge about the sensors' location is required. Incorporation of technologies like global positioning system (GPS) into these networks increases the cost and power requirements of the sensors. Alternatively, a periodic calibration of sensors' positions can be accomplished by establishing collaboration among the sensors. This procedure is known as self-localization.

In centralized sensor networks, the sensor measurements are routed to a central unit which performs the task of obtaining the sensors' locations. In [3,4], the authors suggest a centralized framework and propose a maximum likelihood (ML)-based solution for finding the sensors' locations. Self-localization is also addressed using beacon nodes, also known as anchor nodes, leader nodes or access points [5,6]. In the remainder of this paper, we use the term beacon nodes when referring to sensors which have some initial information about their positions. In distributed sensor localization algorithms, the beacons broadcast their locations. The sensors estimate their distances from the beacons and using these distances, a set of equations or geometric constraints are formed.

* Corresponding author.
E-mail addresses: vema@ece.sunysb.edu (M. Vemula), monica@ece.sunysb.edu (M.F. Bugallo), djuric@ece.sunysb.edu (P.M. Djurić).

0165-1684/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.sigpro.2008.12.019

Triangulation, trilateration, least squares (LS) or other optimization methods have been used in the literature to obtain the sensors' positions [7,8]. In these methods, the effects of uncertainty due to measurement noise and beacon position are not considered. A quantitative comparison of some of these distributed techniques is provided in [9]. Static and mobile beacon-based self-localization procedures have also been explored in the literature (see, for example [6,10–13]). A more comprehensive survey of localization in sensor networks is provided in [14,15] and the references therein.

In this paper, we propose a *distributed* Bayesian (BS) framework for solving the sensor self-localization problem. In [16], an iterative multilateration technique for localizing nodes with unknown location is proposed. In our work, we consider similar algorithms, but we formulate the sensor location estimation from a probabilistic point of view and we include as input information uncertainty of the beacon positions. Most sensor localization algorithms within the probabilistic framework can be classified as (a) ML methods or (b) BS posterior estimation methods. The advantage of the BS techniques is in that they provide a principled way of dealing with location uncertainty and multi-sensor fusion [17]. This latter framework is considered in [18–21]. In [18], the sensors with unknown positions receive the descriptions of the beacon positions which are used by the sensors to update their own positions. However, it is unclear how one computes and stores the posterior probability distributions of the locations. In [19], the authors propose factorized variational BS methods for energy-based detection and localization. Two popular methods for numerical estimation of the posterior distributions of the locations are based on importance sampling (IS) and Monte Carlo Markov chains (MCMC) methods. In [20], an MCMC scheme for obtaining sensor self-localization is employed. There, the authors do not consider the impact of beacon position uncertainty. In [21], an IS based method known as non-parametric belief propagation (NPB) for self-localization of sensors is presented. According to the proposed scheme, each sensor node updates the message received from neighboring sensors and utilizes it to compute its belief distribution which is then transmitted to the neighboring sensors. In [21], each sensor transmits to its neighbor a message whereas in our approach the sensors broadcast the same message and thereby provides probabilistic descriptions of their locations.

More specifically, in this paper we propose methods for distributed sensor self-localization that incorporate beacon position uncertainty using iterative LS and Monte Carlo sampling-based methods. Contributions of this paper include:

- proposal of IS and cost-based (CS) algorithms for sensor self-localization, and
- development of approximate iterative hybrid Cramér–Rao bounds (HCRBs) for sensor self-localization under beacon position uncertainty.

The organization of the paper is as such. In Section 2.1, we formulate the sensor localization problem and describe a generic framework for the proposed algorithms. In Section 3, we present iterative BS and LS methods while in Section 4 we provide Monte Carlo sampling-based solutions to the problem. In Section 5.3, we discuss some computational and communication requirements of the proposed methods. In Section 6, we derive approximate HCRBs of the estimates of a sensor location with multiple beacons under position uncertainty and bounds for the proposed distributed framework. In Section 7, we show some simulation results that demonstrate the effectiveness of the proposed methods. We finally conclude the paper with Section 8.

2. Sensor self-localization: problem statement and proposed framework

2.1. Proposed framework and signal modeling

Consider the sensor network shown in Fig. 1. The shaded nodes represent beacons that broadcast their location details using probabilistic or spatial descriptions. The parameters of these descriptions are transmitted using known reference or pilot signals. We represent this prior information about their locations, $\ell_b \in \mathbb{R}^2, b = 1,2,3$ as $p(\ell_b)$, and we write for the signal received by a sensor

$$y_{s,b} = f(\ell_s, \ell_b) + v_{s,b} \tag{1}$$

where $y_{s,b}$[1] is the received signal characteristic by sensor s from beacon b; $\ell_s, \ell_b \in \mathbb{R}^2$ are the positions of the nodes s and b in the two-dimensional Cartesian coordinate system ($\ell_s = [l_{s,x}, l_{s,y}]^\top$ and $\ell_b = [l_{b,x}, l_{b,y}]^\top$); and $v_{s,b}$ is a Gaussian noise process with mean zero and variance $\sigma^2_{s,b}$. In our simulations we consider measurements that represent the logarithm of the received signal strength (RSS) [4], which can be expressed mathematically as

$$f(\ell_s, \ell_b) = \Psi_0 - 10\alpha \log_{10}(|\ell_s - \ell_b|) \tag{2}$$

where Ψ_0 is the power received at a known reference distance, α is the path-loss attenuation and $|\cdot|$ denotes norm of a vector. This model captures the log-normal shadowing effects in wireless networks but does not capture fully other wireless phenomena such as multipath fading.

2.2. Proposed framework

We explain our sensor self-localization framework by using Fig. 1. There are three sensors and three beacons in this network. The beacons 1–3 are within the sensing range of sensor 1, whereas only beacons 2 and 3 are within the sensing range of sensor 2 and only beacons 1 and 3 are within the sensing range of sensor 3. In each time epoch, nodes with known locations broadcast their location descriptions, which are utilized by the neighboring nodes with unknown locations to obtain their estimates. In the first time epoch, beacons 1–3, transmit their location descriptions. Sensor node 1 receives signals from all the beacons and combines the received prior descriptions and the measurements to obtain an estimate of its location. However, sensors 2 and 3 which do not have beacons 1

[1] Note that the signals are not time-varying but random that account shadow fading in wireless environments.

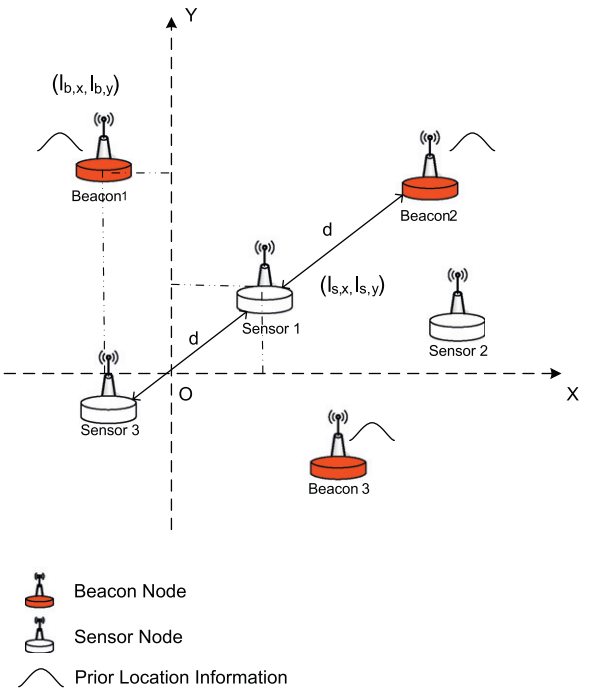

Fig. 1. Sensor network with beacon and sensor nodes.

and 2 within their sensing range, respectively, do not receive any signals from these beacons and, therefore, cannot resolve the ambiguity in their positions. At the end of the first time epoch only the beacons 1–3 and sensor 1 have position information. Sensors which have knowledge about their position may broadcast their locations that may be used by other sensors to estimate their positions. Therefore, in the next time epoch, sensors 2 and 3 utilize the information transmitted by the beacons in their sensing ranges and the sensor to obtain estimates of their respective locations. The algorithm, thus, proceeds in successive time epochs with sensors estimating, broadcasting or updating their location estimates. A suitable criterion based upon convergence of position estimates or upon energy resources can be used for determining when sensors can stop broadcasting their location information.

Clearly, two important issues of the proposed framework are

- the combining of the information in the noisy measurements and the distributions of the beacon locations to obtain accurate sensor estimates, and
- the representation of the sensor location estimates for effective transmission.

In the sequel, we propose methods which address these two problems. In Section 3, we provide two iterative solutions based on linearization of (2) and in Section 4, we present Monte Carlo-based solutions that avoid any linearization of the model and yield sample-based distributions. In these sections we analyze situations with known and unknown measurement noise distributions.

3. Iterative sensor localization methods

In this section we propose two iterative methods for sensor localization which incorporate beacon location uncertainty and are based on linearization of the measurement function in (1). We consider a generic scenario where a sensor s receives signal measurements from B beacons as well as some information about their locations. We denote the set of measurements as $\{y_{s,1},\ldots,y_{s,B}\}$ and describe the uncertainty in beacon location with standard multivariate Gaussian distributions.

Stacking these set of measurements and writing in vector notation, we have

$$\mathbf{y}_s = \mathbf{f}(\ell) + \mathbf{v}_s \tag{3}$$

where $\mathbf{y}_s = [y_{s,1}\ldots y_{s,B}]^\top$, $\mathbf{f}(\ell) = [f(\ell_s,\ell_1),\ldots,f(\ell_s,\ell_B)]^\top$, and $\mathbf{v} = [v_{s,1}\ldots v_{s,B}]^\top$ are all vectors of dimension $B\times 1$. Under assumptions of independent zero mean Gaussian noise, we denote the distribution of $\mathbf{v}_s$ as $\mathcal{N}(\mathbf{0},\boldsymbol{\Sigma}_v)$ where $\boldsymbol{\Sigma}_v = \mathrm{diag}([\sigma^2_{v_{s,1}},\ldots,\sigma^2_{v_{s,B}}])$.[2] The prior description of the location of the beacon nodes is given by $p(\ell_b) = \mathcal{N}(\hat{\ell}_b,\boldsymbol{\Sigma}_{\ell_b}), b = 1,2,\ldots,B$. Assuming independence among the beacon prior distributions, we have $p(\ell_{1:B}) = \mathcal{N}(\hat{\boldsymbol{\mu}}_\ell,\mathbf{P}_{\ell_B})$, where $\hat{\boldsymbol{\mu}}_\ell = [\hat{\ell}_1^\top,\ldots,\hat{\ell}_B^\top]^\top$, and $\mathbf{P}_{\ell_B}$ is a block diagonal matrix, i.e., $\mathbf{P}_\ell = \mathrm{diag}(\boldsymbol{\Sigma}_{\ell_1},\ldots,\boldsymbol{\Sigma}_{\ell_B})$.[3]

3.1. BS method with linearization

A direct approach for solving the sensors' positions is within the ML framework. The optimization criterion can be written as

$$\arg\min_{\ell_s}[(\mathbf{y}_s - \mathbf{f}(\ell))^\top \boldsymbol{\Sigma}_v^{-1}(\mathbf{y}_s - \mathbf{f}(\ell))] \tag{4}$$

The Levenberg–Marquardt method is one popular optimization method for finding solutions to problems like (4) [22]. However, in its straightforward application, the prior information $p(\ell_{1:B})$ of the beacons positions uncertainty is not incorporated. We change the procedure to incorporate this prior knowledge.

We denote the sensor position and the set of beacon locations as $\ell = [\ell_s^\top,\ell_{1:B}^\top]^\top$, a $2(B+1)\times 1$ vector. Using a Taylor series expansion and neglecting higher order terms, we linearize (3), with respect to ℓ around ℓ^0 as follows:

$$\mathbf{y}_s - f(\ell^0) \approx \mathbf{H}\times(\ell-\ell^0) + \mathbf{v}_s \tag{5}$$

[2] The symbol diag($\mathbf{x}$) represents a diagonal matrix formed with the vector $\mathbf{x}$ as its diagonal.

[3] The symbol $\mathrm{diag}(\mathbf{A},\mathbf{B}) = \left(\begin{smallmatrix}\mathbf{A} & \mathbf{0}\\ \mathbf{0} & \mathbf{B}\end{smallmatrix}\right)$, where $\mathbf{A},\mathbf{B}$ are diagonal matrices.

where

$$\mathbf{H} = \begin{bmatrix} \frac{\partial f(\ell_s,\ell_1)}{\partial \ell_s} & \frac{\partial f(\ell_s,\ell_1)}{\partial \ell_1} & \cdots & \frac{\partial f(\ell_s,\ell_1)}{\partial \ell_B} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial f(\ell_s,\ell_B)}{\partial \ell_s} & \frac{\partial f(\ell_s,\ell_B)}{\partial \ell_1} & \cdots & \frac{\partial f(\ell_s,\ell_B)}{\partial \ell_B} \end{bmatrix} = \begin{bmatrix} \frac{\partial f(\ell_s,\ell_1)}{\partial \ell_s} & \frac{\partial f(\ell_s,\ell_1)}{\partial \ell_1} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial f(\ell_s,\ell_B)}{\partial \ell_s} & \mathbf{0} & \cdots & \frac{\partial f(\ell_s,\ell_B)}{\partial \ell_B} \end{bmatrix} \quad (6)$$

with $\partial f(\ell_s,\ell_i)/\partial \ell_j = 0, \forall i \neq j,\ i,j \in 1,\ldots,B$. Here $\mathbf{H}$ is a $2B \times 2(B+1)$ matrix and ℓ^0 is a vector of dimension $2(B+1)\times 1$. A BS solution to this linear problem which incorporates the prior information is as follows [23]:

$$\hat{\ell} - \ell^0 = (\boldsymbol{\Sigma}_l^{-1} + \mathbf{H}^\top \boldsymbol{\Sigma}_v^{-1}\mathbf{H})^{-1}\mathbf{H}^\top \boldsymbol{\Sigma}_v^{-1}(\mathbf{y}_s - \mathbf{f}(\ell^0)) \quad (7)$$

where $\boldsymbol{\Sigma}_l^{-1} = \mathrm{diag}(\mathbf{0}_2, \mathbf{P}_{\ell_B}^{-1})$. For the sensor localization problem, we solve for ℓ by iteratively computing (7) with a suitable stopping criterion. Therefore, at iteration i we have

$$\hat{\ell}^{(i)} = \hat{\ell}^{(i-1)} + (\boldsymbol{\Sigma}_l^{-1} + \mathbf{H}^{(i)\top}\boldsymbol{\Sigma}_v^{-1}\mathbf{H}^{(i)})^{-1}\mathbf{H}^{(i)\top}\boldsymbol{\Sigma}_v^{-1}(\mathbf{y}_s - \mathbf{f}(\hat{\ell}^{(i-1)})) \quad (8)$$

where $\mathbf{H}^{(i)}$ is defined in (6) and computed at $\hat{\ell}^{(i-1)}$. The initialization of the beacon parameters by $\hat{\ell}^{(0)}$ is obtained using $\hat{\boldsymbol{\mu}}_\ell$, and the sensor location is initialized using the mean of the beacon positions. The covariance matrix of the estimate of ℓ is $(\boldsymbol{\Sigma}_l^{-1} + \mathbf{H}^{(i)\top}\boldsymbol{\Sigma}_v^{-1}\mathbf{H}^{(i)})$.

3.2. LS method with linearization

A LS criterion that incorporates the prior knowledge of the beacons' locations can be written as [24]

$$\arg\min_{\ell}[(\ell-\ell^0)^\top \boldsymbol{\Sigma}_l^{-1}(\ell-\ell^0) + |\mathbf{y}_s - \mathbf{H}\times(\ell-\ell^0)|^2]$$

and its solution is

$$\hat{\ell} = \ell^0 + (\boldsymbol{\Sigma}_l^{-1} + \mathbf{H}^\top\mathbf{H})^{-1}\mathbf{H}^\top(\mathbf{y}_s - f(\ell^0))$$

Similarly we can iteratively solve for ℓ. Thus, at iteration i, we have

$$\hat{\ell}^{(i)} = \hat{\ell}^{(i-1)} + (\boldsymbol{\Sigma}_l^{-1} + \mathbf{H}^{(i)\top}\mathbf{H}^{(i)})^{-1}\mathbf{H}^{(i)\top}(\mathbf{y}_s - \mathbf{f}(\hat{\ell}^{(i-1)})) \quad (9)$$

and the corresponding covariance matrix of $\hat{\ell}^{(i)}$ is $(\boldsymbol{\Sigma}_l^{-1} + \mathbf{H}^{(i)\top}\mathbf{H}^{(i)})$.

Clearly, we can see that (8) differs from (9) in that the LS solution does not take into account the distribution of the noise process $\mathbf{v}_s$. In the proposed iterative methods, the iterations are performed in one time epoch and once a sensor resolves its position, then in the subsequent time epochs it, too, broadcasts its position description. Using the final estimates and the covariance matrix, the sensor position can be modeled as a Gaussian distribution and these parameters are transmitted in the following time epochs for both the BS and LS methods.

4. Monte Carlo-based methods for sensor localization

In Monte Carlo-based methods the posterior distribution of the sensor location is approximated by a set of samples. In these methods, the measurement functions are not linearized but the marginal distributions are represented by a set of weighted samples. We consider the earlier scenario where each sensor receives measurements from B beacons and their location descriptions.

4.1. IS-based method

In a BS framework, all knowledge about the sensors' and beacons' positions is contained in the posterior distribution $p(\ell|\mathbf{y}_s)$. We, therefore, have

$$p(\ell|\mathbf{y}_s) = p(\ell_s,\ell_1,\ldots,\ell_B|\mathbf{y}_s) \propto p(\ell_s)\prod_{b=1}^{B} p(y_{s,b}|\ell_s,\ell_b)p(\ell_b) \quad (10)$$

which is obtained using Bayes' theorem and assuming independence among the prior distributions. Closed form analytical solutions to this posterior density cannot be obtained when the function $f(\cdot)$ in (1) is nonlinear in the state parameters and/or the noise process $v_{s,b}$ is non-Gaussian. Therefore, we use Monte Carlo (MC) methods for capturing the posterior distribution or some of its statistics. In most MC methods these posterior distributions are represented by sample-based discrete random measures. IS is one such method where one can obtain a discrete representation of the posterior distribution [25].

Very briefly, with IS, the density $p(\ell)$ is approximated by a weighted set of samples $\boldsymbol{\Xi}_l \equiv \{\ell^{(m)}, w^{(m)}\}_{m=1}^{M}$ with M being the total number of drawn samples and m the sample index. This approach is particularly useful when it is infeasible to draw samples directly from the density $p(\ell)$ but can be evaluated up to a constant. The samples are obtained from another function known as importance function or proposal density, $\pi(\ell)$. The importance weights are proportional to $p(\ell^{(m)})/\pi(\ell^{(m)})$ and measure the quality of the generated particles. A rigorous treatment of the subject can be found in [25].

The posterior density (10) is approximated as a weighted set of particles $\{\omega^{(m)}, \ell_s^{(m)}, \ell_1^{(m)},\ldots,\ell_B^{(m)}\}$,

$$p(\ell_s,\ell_1,\ldots,\ell_B|y_{s,1},\ldots,y_{s,B}) = \sum_{m=1}^{M}\omega^{(m)}\delta(\ell_s-\ell_s^{(m)})\times\prod_{b=1}^{B}\delta(\ell_b-\ell_b^{(m)}) \quad (11)$$

which is then easily marginalized, i.e.,

$$p(\ell_s|y_{s,1},\ldots,y_{s,K}) \approx \sum_{m=1}^{M}\omega^{(m)}\delta(\ell_s-\ell_s^{(m)}) \quad (12)$$

As before, if the sensor obtains information about its location, it is described by a probability distribution which is broadcast in the next time epoch to other sensors so that they in turn estimate their own positions.

The main steps of this IS-based procedure are:

- Generation of beacon and sensor location samples $\{\ell_s^{(m)}, \ell_1^{(m)}, \ldots, \ell_B^{(m)}\}$: The beacon location samples are drawn from their transmitted prior location distributions, i.e., $\ell_b^{(m)} \sim p(\ell_b)$, and samples representing the sensor location drawn from an importance distribution function $\pi(\ell_s)$. In Section 4.3, we propose methods for constructing these importance distributions.
- Computation of weights: The weights, $\omega^{(m)}$, are obtained according to

$$\omega^{(m)} \propto \frac{p(\ell_s^{(m)})}{\pi(\ell_s^{(m)})} \prod_{b=1}^{B} \frac{p(y_{s,b}|\ell_s^{(m)}, \ell_b^{(m)}) p(\ell_b^{(m)})}{\pi(\ell_b^{(m)})} \tag{13}$$

- Estimation of the transmitting distribution parameters: Transmission of these sample-based distributions would require large amounts of communication resources, and therefore we approximate them with standard probability distributions and broadcast their parameters only. As an example, we approximate the sample-based distribution by a Gaussian $\mathcal{N}(\boldsymbol{\mu}_{\ell_s}, \boldsymbol{\Sigma}_{\ell_s})$ where the means $\boldsymbol{\mu}_{\ell_s}$ and covariance matrices $\boldsymbol{\Sigma}_{\ell_s}$ are obtained as

$$\begin{aligned} \boldsymbol{\mu}_{\ell_s} &= \sum_{m=1}^{M} \omega^{(m)} \ell_s^{(m)} \\ \boldsymbol{\Sigma}_{\ell_s} &= \sum_{m=1}^{M} \omega^{(m)} (\ell_s^{(m)} - \boldsymbol{\mu}_{\ell_s})(\ell_s^{(m)} - \boldsymbol{\mu}_{\ell_s})^{\top} \end{aligned} \tag{14}$$

4.2. CS Monte Carlo sampling method

In the above proposed IS method, the computation of the weights in (13) requires the evaluation of the likelihood term for which the probability distribution of the measurement noise should be completely known. In many scenarios, such information is not available. We propose an alternative sampling method which does not make probabilistic assumptions of the model noise. In the following method, we only require that the noise process is zero mean.

We motivate the approach as follows: under assumption of zero mean noise, a LS criterion for obtaining the sensor location given the measurements $y_{s,1}, \cdots, y_{s,B}$ can be formulated as

$$\hat{\ell}_s = \arg\min_{\ell_s} \left\{ \mathscr{C}(\ell_s) = \sum_{b=1}^{B} |y_{s,b} - f(\ell_s, \ell_b)|^2 \right\} \tag{15}$$

Note that in this formulation we have not explicitly included the beacon location information. However, the optimization is carried out by incorporating the beacon location uncertainty. Using this formulation as a starting point, we propose a procedure where we draw samples representing the beacons' and sensors' locations and associate a cost with these samples using the measurements and beacon location descriptions. The main steps of the algorithm are as follows:

- *Generation of beacon and sensor location samples*: The beacon location samples $\{\ell_1^{(m)}, \ldots, \ell_B^{(m)}\}$ are drawn using the beacon location descriptions. The sensor samples $\ell_s^{(m)}$ are obtained using an importance function as discussed in Section 4.3. These beacon and sensor samples can be considered as representative solutions of the beacon and sensor locations.
- *Obtaining costs*: For each of the samples generated in the previous step, we associate a cost that defines their quality under sensor measurements. In general, the cost function depends on the observations and the sampled locations. We assign the costs according to

$$\mathscr{C}^{(m)} = \sum_{b=1}^{B} \varrho(y_{s,b}, \ell_s^{(m)}, \ell_b^{(m)}) = \sum_{b=1}^{B} \varrho(\varepsilon_{s,b}) \tag{16}$$

where $\varepsilon_{s,b} = y_{s,b} - f(\ell_s^{(m)}, \ell_b^{(m)})$ and $\rho(\cdot,\cdot,\cdot)$ is a nonnegative function of its arguments. The cost function is chosen to reflect that more representative samples of the locations have smaller costs and samples which are far away from the true locations have larger costs. We use the following cost functions:
(1) L2 cost function: $\varrho(\varepsilon) = |\varepsilon|^2$
(2) L1 cost function: $\varrho(\varepsilon) = |\varepsilon|$
(3) Fair function: $\varrho(\varepsilon) = 2k^2[\frac{|\varepsilon|}{k} \log(1 + \frac{|\varepsilon|}{k})]$ with $k = 1.3998$ [26].

Using these costs we form a pseudo probability measure such that,

$$\tilde{\pi}(\ell^{(m)}) \propto \frac{1}{\mathscr{C}^{(m)}}$$

such that

$$\sum_{m} \tilde{\pi}(\ell^{(m)}) = 1. \tag{17}$$

- *Estimating sensor location regions*: We can approximate the sensor location distribution by using standard probability distributions such as the Gaussian distribution (as in the previous subsection). These distributions can be obtained using the sample mean sensor location and the covariance matrix which are calculated using the pseudo probability measure (17). These statistics are obtained as earlier with

$$\begin{aligned} \tilde{\boldsymbol{\mu}}_{\ell_s} &\approx \sum_{m=1}^{M} \tilde{\pi}(\ell^{(m)}) \ell_s^{(m)} \\ \tilde{\boldsymbol{\Sigma}}_{\ell_s} &\approx \sum_{m=1}^{M} \tilde{\pi}(\ell^{(m)}) (\ell_s^{(m)} - \tilde{\boldsymbol{\mu}}_{\ell_s})(\ell_s^{(m)} - \tilde{\boldsymbol{\mu}}_{\ell_s})^{\top}. \end{aligned} \tag{18}$$

4.3. Construction of the importance function for sensor location

Samples representing the beacons' positions are drawn from their prior or marginal posterior distributions of their locations. However, in absence of any prior of the sensor location, we construct importance functions for the sensor location. Naive approaches of drawing samples from the entire sensor field or regions within the beacon sensing region require a large number of samples to be

drawn for sufficient accuracy in location estimates. Intuitively, we would like to draw samples from regions around the sensors true location density.

For RSS-based measurements, we obtain initial estimates of the sensor location using fast trilateration methods for constructing the prior density. We first convert the RSS measurements to distance measurements. Consider the measurement $y_{s,b}$ received by the sensor s from the beacon b and define

$$z_{s,b} = \frac{\Psi_0 - y_{s,b}}{10\alpha} \quad b = 1, 2, \ldots, B$$
$$d_{s,b} = 10^{z_{s,b}} \approx \sqrt{(l_{s,x} - \hat{l}_{b,x})^2 + (l_{s,y} - \hat{l}_{b,y})^2} \tag{19}$$

where $d_{s,b}$ represents an initial estimate of the Euclidean distance between the sensor and the beacon, $z_{s,b}$ is an intermediate variable and $[\hat{l}_{b,y}, \hat{l}_{b,y}]$ are estimates of the beacon position. When beacons approximate their location information using Gaussian distributions, this estimate is readily available through the mean of the distribution. Using (19) and a fast trilateration scheme, we obtain the initial estimates $\tilde{\boldsymbol{\mu}}_s = [\tilde{l}_{s,x}, \tilde{l}_{s,y}]^\top$ of the sensor position. Thus, we can construct a Gaussian importance function, $\pi(\ell_s) = \mathcal{N}(\tilde{\boldsymbol{\mu}}_s, \tilde{\boldsymbol{\Sigma}}_s)$ for drawing samples to compute the weights in (13). The choice of the covariance matrix $\tilde{\boldsymbol{\Sigma}}_s$ is arbitrary. In our simulations we have used $\tilde{\boldsymbol{\Sigma}}_s = \tilde{\sigma}_s^2 I_2$ with a judicious choice of $\tilde{\sigma}_s$. A similar approach is torun the iterative algorithms discussed in Section 3 for smaller number of iterations and use the obtained estimates to construct Gaussian distributions as discussed above.

5. Discussion of the algorithms

5.1. Assumption of independence of prior distributions

The proposed algorithms for sensor localization are suboptimal because of our construction of the *joint* distributions of the locations of the beacons and sensors that broadcast. We obtain the joint distribution as a product of marginals (of the individual beacons and sensors), which is only an approximation. The optimal fusion algorithm can be derived for small networks along the lines shown in (10) by taking into consideration the dependence of these distributions on data. Due to lack of space, we do not present the optimal fusion algorithm which has significant communication, topological, and processing requirements. Some of these requirements arise due to loops in the network.

5.2. Parameterizations of the location distributions

In the Monte Carlo methods, the resulting location distributions are represented by weighted sets of samples. To avoid large communication requirements, the sensors approximate their location distributions with parametric distributions whose parameters are then transmitted. Since in most scenarios the distributions are unimodal, one can approximate them with Gaussian distributions. In that case, each broadcast of a location requires the transmission of five numbers, two for the mean and three that define the covariance matrix of the Gaussian. This communication can further be reduced by approximation the distribution with a Gaussian with a covariance matrix of the form $\sigma^2\mathbf{I}$. Then the total of transmitted number per reported location is only three.

5.3. Computation requirements

The BS and LS methods require the calculation of the sensor location estimates through (8) and (9), which includes multiplicative and inverse operations on matrices of dimensions $2(B+1) \times 2(B+1)$ with B being the number of nodes communicating with a particular sensor. The LS method requires two matrix inversions, two matrix–matrix multiplications and one matrix–vector multiplication whereas the BS method requires three matrix inversions, four matrix–matrix multiplications and one matrix–vector multiplication. For the same number of iterations, the LS method has smaller computational requirements. In the section on simulations, we see that when compared with the BS algorithm, the LS method is less accurate. Thus, we have a tradeoff between computation and accuracy of the two algorithms.

In the Monte Carlo IS and CS algorithms, the computational requirements of the weights of the IS algorithm depend on the form of the likelihood function and those of the CS algorithm, on the cost function. A Gaussian-type likelihood function requires computations of exponential (or logarithmic) functions, which is demanding. On the other hand, the L2 and L1 cost functions have smaller computational requirements. If the computations are a big issue, one can choose an appropriate cost function that does not require much computation (for example, a zero-one cost function).

6. HCRBs for sensor self-localization

We now derive Cramér–Rao Bounds (CRBs) for sensor self-localization for the two scenarios (a) a single sensor which receives measurements and location information from multiple beacons, and (b) multiple sensors with multiple beacons which localize themselves using the proposed distributive framework.

The CRB is a bound of the variance of unbiased estimators for non-random parameters and the BS or the Van-Trees version of the CRB is a bound of the mean square error of estimates of random parameters [23,27]. We consider here a hybrid Cramér–Rao bound (HCRB), which is a bound of the mean square error of estimates of random and non-random parameters [28,29].

6.1. Single sensor, multiple beacons

Recall that $\ell = [\ell_s, \ell_1, \ldots, \ell_B] = [\ell_s, \bar{\ell}]$, is the unknown set of parameters to be estimated, i.e., the vector containing the locations of the sensor ℓ_s and the beacons $\bar{\ell}$ which are modeled as non-random and random parameters, respectively, and $\mathbf{y}_s = [y_{s,1}, \ldots, y_{s,B}]$ is the set of beacon signal measurements made by sensor s. The HCRB has the

following form:

$$\boldsymbol{\Sigma}_\ell = \mathbb{E}\{[\hat{\ell} - \ell][\hat{\ell} - \ell]^\top\} \geqslant \mathbf{J}^{-1} \tag{20}$$

where $\boldsymbol{\Sigma}_\ell$ is the estimation error covariance matrix, and $\mathbf{J}$ is the hybrid Fisher information matrix (HFIM) defined as

$$\begin{aligned}\mathbf{J} &= \mathbb{E}_{\ell,\mathbf{y}_s}[\{\nabla_\ell \log p(\mathbf{y}_s, \ell)\ \ \nabla_\ell^\top \log p(\mathbf{y}_s, \ell)\}] \\ \mathbf{J} &= \mathbb{E}_\ell[\mathbb{E}_{\mathbf{y}_s|\ell}[\{\nabla_\ell \log p(\mathbf{y}_s|\ell)\ \ \nabla_\ell^\top \log p(\mathbf{y}_s|\ell)\}]] + \mathbf{J}_b \\ \mathbf{J} &= \mathbf{J}_s + \mathbf{J}_b\end{aligned} \tag{21}$$

with

$$\mathbf{J}_b = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbb{E}_{\bar{\ell}}[\{\nabla_{\bar{\ell}} \log p(\bar{\ell})\ \ \nabla_{\bar{\ell}}^\top \log p(\bar{\ell})\}] \end{bmatrix} = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_{bb} \end{pmatrix}$$

and $\nabla_\ell = [\partial/\partial\ell_s, \partial/\partial\ell_1, \ldots, \partial/\partial\ell_B]^\top$. We model the prior distribution of the beacon location as a Gaussian, and therefore the matrix $\mathbf{J}_{bb}$ is a block diagonal matrix of inverses of all the covariance matrices of the beacons' location distributions. Details of the computation of these CRBs are provided in Appendix A.

6.2. Multiple sensors, multiple beacons

We calculate bounds for the proposed distributed framework where at every time epoch sensors with known position information broadcast their location information and sensors with unknown information attempt to localize. Also, sensors with known information utilize measurements from new beacons to update their information. These aspects are considered in the calculation of the HCRB for the proposed distributive framework.

With a slight change in notation, we write the set of all received signals $\mathbf{y}_s$, at sensor s as

$$\mathbf{y}_s = \mathbf{f}_s(\ell) + \mathbf{v}_s \tag{22}$$

with $\mathbf{f}_s(\ell) = [f_{s,1}(\cdot), \ldots, f_{s,B}(\cdot)]^\top$ and $\mathbf{v}_s = [v_{s,1}, \ldots, v_{s,B}]^\top$. We form a vector $\hat{\mathbf{y}}$ by stacking all the measurements $\hat{\mathbf{y}} = [\mathbf{y}_1^\top, \ldots, \mathbf{y}_B^\top]^\top$ of all the sensors in the network. Note that only sensors with known location information broadcast their position information, therefore, the entries corresponding to sensors which do not broadcast are zero in this vector. We obtain the expression

$$\hat{\mathbf{y}} = \hat{\mathbf{f}}(\ell) + \hat{\mathbf{v}} \tag{23}$$

When we consider the set of sensor locations with prior information as random parameters and the set of sensor locations with no prior information as non-random parameters, the HFIM at time epoch n can be obtained as in (21)

$$\mathbf{J}^{(n)} = \mathbb{E}_{\ell,\hat{\mathbf{y}}^{1:n}}[-\{\Delta_\ell^\ell \log p(\hat{\mathbf{y}}^{1:n}, \ell)\}] \tag{24}$$

with expectation performed over sensors with resolved positions, $\Delta_\ell^\ell = \nabla_\ell \nabla_\ell^\top$, and $\hat{\mathbf{y}}^{1:n}$, the set of all measurements from time epoch 1 to n. We have that $p(\hat{\mathbf{y}}^{1:n}, \ell) = p(\hat{\mathbf{y}}^n|\ell)p(\hat{\mathbf{y}}^{1:n-1}, \ell)$, and therefore we can write (24) as

$$\begin{aligned}\mathbf{J}^{(n)} &= \mathbb{E}_{\ell,\hat{\mathbf{y}}^n}[-\{\Delta_\ell^\ell \log p(\hat{\mathbf{y}}^n|\ell)\}] + \mathbb{E}_{\ell,\hat{\mathbf{y}}^{1:n-1}}[-\{\Delta_\ell^\ell \log p(\hat{\mathbf{y}}^{1:n-1}, \ell)\}] \\ &= \mathbb{E}_{\ell,\hat{\mathbf{y}}^n}[-\{\Delta_\ell^\ell \log p(\hat{\mathbf{y}}^n|\ell)\}] + \mathbf{J}^{(n-1)} \\ &= \mathbb{E}_\ell[\mathbb{E}_{\hat{\mathbf{y}}^n|\ell}[\{\nabla_\ell \log p(\hat{\mathbf{y}}^n|\ell)\ \ \nabla_\ell^\top \log p(\hat{\mathbf{y}}^n|\ell)\}]] + \mathbf{J}^{(n-1)} \\ &= \mathbb{E}_\ell[\{\nabla_\ell \hat{\mathbf{f}}^\top(\ell) \boldsymbol{\Sigma}_v^{-1} \nabla_\ell^\top \hat{\mathbf{f}}^\top(\ell)\}] + \mathbf{J}^{(n-1)}\end{aligned} \tag{25}$$

with $\mathbf{J}^{(0)} = \mathbb{E}_\ell[\{\nabla_\ell \log p(\ell)\ \ \nabla_\ell^\top \log p(\ell)\}]$. If the sensor in a particular time epoch is able to resolve its position, then its HCRB is smaller than when it is not able to resolve its position. Using this criterion, we identify sensors which can become beacons in the next time epoch. We rearrange the matrix $\mathbf{J}^{(n-1)}$ such that the elements corresponding to sensors with unknown locations are all 0's and denote this new prior matrix as $\tilde{\mathbf{J}}_b^{(n-1)}$. We write (25) as $\mathbf{J}^{(n)} = \mathbf{J}_s^{(n)} + \tilde{\mathbf{J}}_b^{(n-1)}$. In computing the Cramér–Rao bound we assume that the prior is a Gaussian with mean equal to the true locations of the nodes and a covariance matrix given by $(\tilde{\mathbf{J}}_b^{(n-1)})^{-1}$. Thus, in the computation of $\mathbf{J}_s^{(n)}$, the averaging was done using these prior locations.

7. Simulation results

To assess the performance of the proposed algorithms, we performed several simulation experiments with the purpose of (a) motivating the need of prior information as in the LS, BS, and IS methods for sensor localization, and (b) studying the performance of these algorithms for a large network.

7.1. Motivation

We considered the sensor network as shown in Fig. 2(a). The beacon nodes, $B1$, $B2$, and $B3$ with some prior information are represented with shaded circles and the sensor $S1$ with unknown location is denoted by a diamond mark. The beacons $B1$, $B2$, and $B3$ and the sensor $S1$ were located at (0,15), (0,0), (15,0) and (21.213, 21.213), respectively. We assumed Gaussian prior location distributions $\mathcal{N}(\boldsymbol{\mu}_{\mathbf{l}_b}, \rho\mathbf{I}_2)$ for the beacon positions, where the prior mean had an offset b from the true position, i.e, $\boldsymbol{\mu}_{\mathbf{l}_b} = [\mathbf{l}_{b,x} + \mathrm{b}, \mathbf{l}_{b,y} + \mathrm{b}]^\top$. In the LS and BS methods, the position of sensor $S1$ was initialized with the average of the three beacon mean locations. Recall that the LS and BS methods are iterative procedures. These procedures were run for 50 iterations.[4] Alternately, a simple stopping procedure for these methods is to compare the change in the residual error between successive iterations. For the IS method, in the construction of the sensor proposal distribution we ran the LS and BS methods with 10 iterations and utilized this sensor estimate in the construction of the prior proposal distribution with $\tilde{\sigma_s} = 3$ as outlined in Section 4.3. The number of drawn samples for each node location was $M = 1000$.

[4] Although these computations may seem demanding, this is usually not an issue if power dissipation is of concern. Sensors spend much more power on communication.

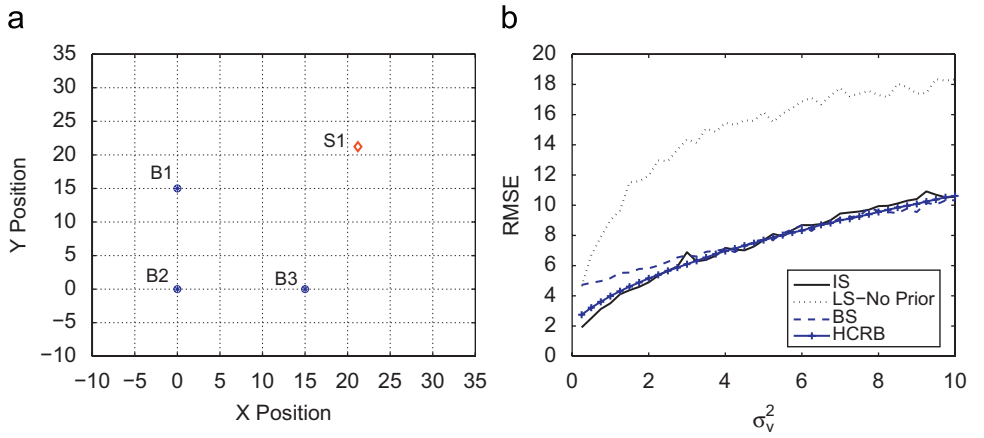

Fig. 2. Topology of sensor network (left) and RMSEs of the LS, BS, and IS methods with prior information and of LS without prior information (right). (a) Sensor network, (b) RMSEs.

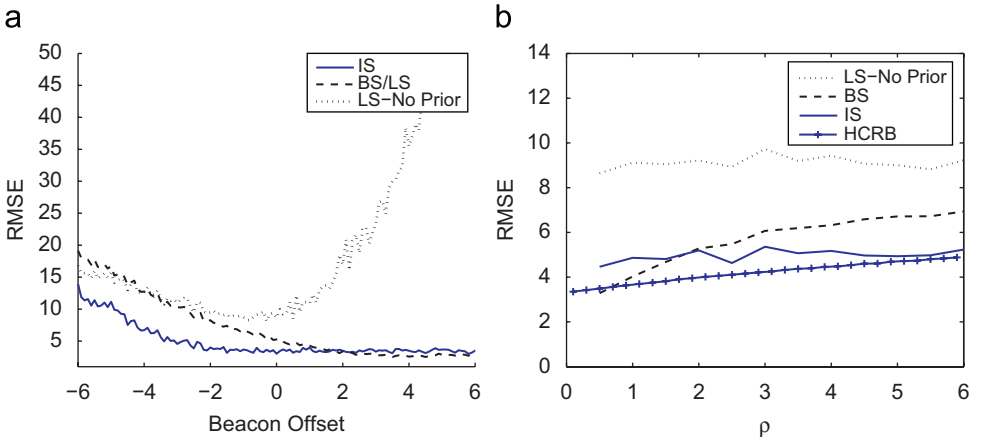

Fig. 3. RMSEs of the LS, BS and IS methods. (a) Varying b_2, (b) varying ρ.

In Fig. 2(b), we plotted the root mean square errors (RMSEs) of the location estimate of sensor $S1$ with varying σ_v^2. In our simulations the reference power was fixed at $\Psi_0 = -50$ dB, the measurement noise variance σ_v^2 across all sensors was the same, the beacon offset was set to $b = 0$ and the variance parameter to $\rho = 2$. We also simulated the LS algorithm wherein we assumed that the mean location of the beacons was their true positions and did not take into account any uncertainty associated with it. We termed this method as LS-No Prior. Clearly, the LS-No Prior has inferior results in terms of RMSE in comparison with the other methods.

In another set of simulations, we studied the effect of the sensor position estimate by varying the offset of beacon $B2$. The beacons $B1$ and $B3$ had zero offsets. In Fig. 3(a) we plotted the corresponding RMSEs with varying b_2, the offset of the location of beacon $B2$. For this scenario, the performance of the IS procedure is similar to that of the LS and BS methods when b_2 is positive. Also, when the offset is positive, the LS-No Prior method had a large error in estimating the sensor's position. When the offset is negative, for small values of b_2, the IS outperforms the LS/BS methods. Thus, the IS and LS/BS which take into account the beacon prior position information produce reliable sensor estimates. Clearly, we can see that the LS-No prior method is considerably inferior than the other procedures which incorporate prior information about the beacon's location.

7.2. Localization in a large network

In this experiment, we studied the performance of the IS and CS methods for self-localization in large networks. To this end we considered a network which consisted of 48 randomly distributed sensors with unknown locations and of 16 beacons with some location information. As earlier, the distributions of the prior location of the beacon nodes were modeled using Gaussian distributions $\mathcal{N}(\boldsymbol{\mu}_{l_b}, \rho\mathbf{I})$ with $\boldsymbol{\mu}_{l_b} = [l_{b,x} + b, l_{b,y} + b]^\top$ where b represents the unknown offset in the beacon location. In this experiment, we quantified the performance of the proposed algorithms by using the cumulative distributive function (CDF) of the RMSE.

The RMSEs of the estimated sensor positions were computed from $K = 100$ different realizations of measurements. In the first set of simulations, we studied the effect of b on determining the sensor locations. There, we assumed that we know the values of the following parameters: $\rho = 0.5$, $\Psi = -50\,\mathrm{dB}$, and $\alpha = 2.5$. The measurement noise across all the sensors was formulated as Gaussian with $\mu_v = 0$ and $\sigma_v = 1$. In Figs. 4(a) and (b), we plotted the CDFs of the RMSE of the IS, LS, BS and CS methods for $b = 0$ and 2. Further, to rank the performances of our methods we obtained the value of $d_{0.95}$, the 95th percentile of the RMSE error. This percentile states that 95% of the time the RMSE was below this quantity. From Fig. 2(a), $d_{0.95,IS} = 1.25$, $d_{0.95,BS} = 3.25$, and $d_{0.95,CS} = 4.2$. Obviously, in these scenarios, the IS procedure had the best performance. In Fig. 4(b), it can be seen that the tails of the CDFs of the BS and CS methods are more towards the left than for the IS method, suggesting that the best performance of the IS is lower than the best performance of the other methods. However, if one considers the $d_{0.95}$ performance metric, the IS method produced better RMSEs for all the sensors.

In Fig. 5(a), we show more results of our study of the IS procedure. The plotted curves were obtained from an experiment where we used various positive and negative values of b. As can be seen, the method is robust to the sign of the offsets. In Fig. 5(b), we plotted the results from an experiment where we varied the standard deviation of the measurement noise. We see that with increasing the noise, the CDF curve shifts to the right indicating increase in estimation error. We also studied the performance of the CS method in comparison to the IS method. In Fig. 6(a), the performances of the IS and CS methods (with $L2$, $L1$ and Fair cost functions) are displayed. The performance of the CS method which makes no assumptions of the noise distributions is comparable with the performance of the IS method.

Finally, we conducted a set of simulations to study the robustness of the methods to knowledge of the noise distributions. The probabilistic IS method assumed wrong measurement noise distribution $\hat{p}(v) = \mathcal{N}(0, 0.1^2)$ while the true distribution was $p(v) = 0.8\mathcal{N}(0,1) + 0.2\mathcal{N}(3, 0.2^2)$. When the IS method made wrong assumptions, it had much poorer performance than the CS method. Fig. 6(b) shows the robustness of the CS methods

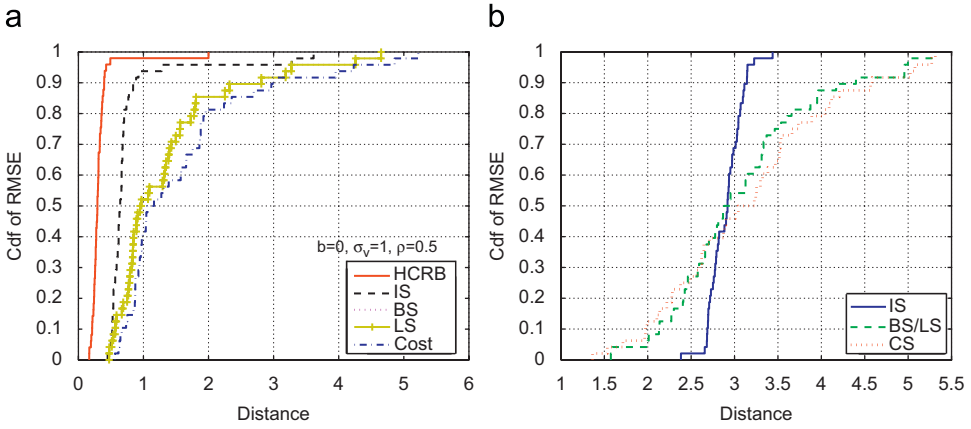

Fig. 4. CDFs of RMSEs using the IS, LS, BS and CS methods. (a) $b = 0$; (b) $b = 2$.

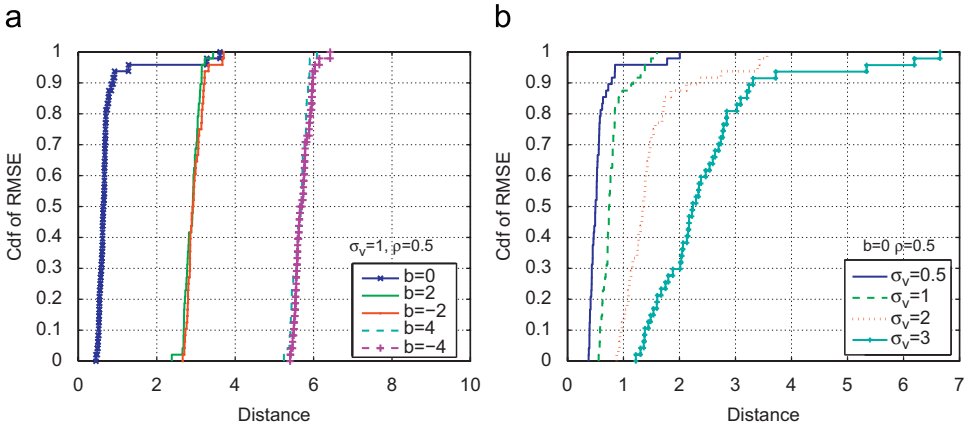

Fig. 5. CDFs of RMSEs of the IS method. (a) RMSEs with various b, (b) RMSEs with σ_v.

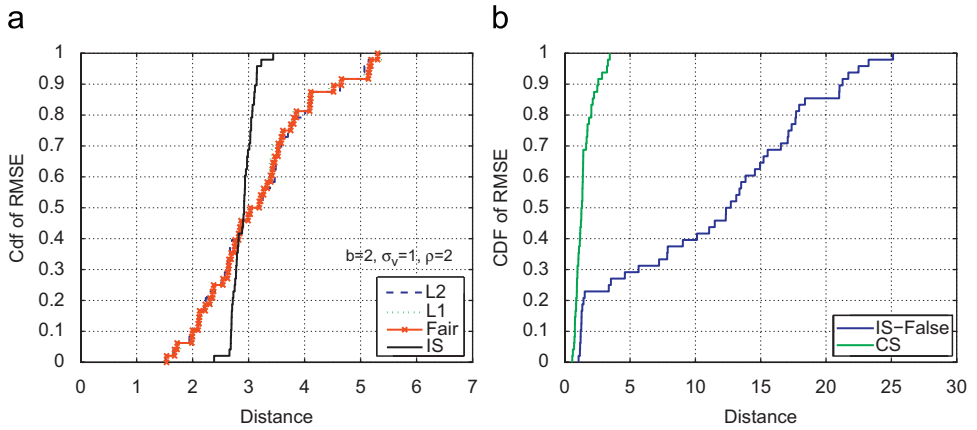

Fig. 6. CDFs of RMSEs of the CS and IS methods. (a) CS and IS with correct assumptions. (b) CS and IS with false assumptions.

and the sensitivity of the probabilistic method to the knowledge of the noise distribution.

8. Conclusions

We proposed distributed algorithms for sensor self-localization with beacon position uncertainty. The proposed algorithms can be classified as iterative least squares (LS) and Bayesian (BS) methods, Monte Carlo importance sampling (IS) and cost-based (CS) methods. The iterative LS and Monte Carlo CS methods do not require knowledge of the model noise distributions while the iterative BS and Monte Carlo IS methods do require it. Through simulations we have observed that the performance of the IS method is good over a wide range of scenarios. However, when the IS method is based on incorrect assumptions about the noise statistics, its performance can be degraded considerably. In those situations, the cost-based approach has relatively good performance.

Appendix A. HCRBs for sensor self-localization

The set of measurements obtained by sensor s can be expressed as (22). When $\mathbf{v}_s$ is a Gaussian random vector and the prior density is also Gaussian with covariance matrix $\boldsymbol{\Sigma}_\ell$, the HFIM can be written as [27]

$$\mathbf{J} = \mathbb{E}_\ell[\{\nabla_\ell \mathbf{h}^\top(\ell)\boldsymbol{\Sigma}_v^{-1}\nabla_\ell^\top \mathbf{h}^\top(\ell)\}] + \mathbf{J}_b \tag{A.1}$$

For the RSS model, the elements of $\nabla_\ell \mathbf{h}^\top(\ell)$ are calculated by

$$\begin{aligned}
\frac{\partial h^b(\ell)}{\partial l_{s,x}} &= -\frac{10\alpha}{\log 10}\left[\frac{(l_{s,x}-l_{b,x})}{(l_{s,x}-l_{b,x})^2+(l_{s,y}-l_{b,y})^2}\right] \\
\frac{\partial h^b(\ell)}{\partial l_{s,y}} &= -\frac{10\alpha}{\log 10}\left[\frac{(l_{s,y}-l_{b,y})}{(l_{s,x}-l_{b,x})^2+(l_{s,y}-l_{b,y})^2}\right]
\end{aligned} \tag{A.2}$$

$$\frac{\partial h^b(\ell)}{\partial l_{b,x}} = -\frac{\partial h^b(\ell)}{\partial l_{s,x}}; \quad \frac{\partial h^b(\ell)}{\partial l_{b,y}} = -\frac{\partial h^b(\ell)}{\partial l_{s,y}}$$

$$\frac{\partial h^b(\ell)}{\partial l_{j,x}} = \frac{\partial h^b(\ell)}{\partial l_{j,y}} = 0 \quad (\forall b \neq j). \tag{A.3}$$

References

[1] F. Zhao, Sensor Networks: An Information Processing Approach, Morgan Kaufmann, Los Altos, CA 94022, USA, 2003.

[2] C.Y. Chong, S.P. Kumar, Sensor networks: evolution, opportunities, and challenges, Proceedings of the IEEE 91 (8) (2003) 1247–1256.

[3] R.L. Moses, D. Krishnamurthy, R.M. Patterson, A self-localization method for wireless sensor networks, EURASIP Journal on Applied Signal Processing (2003) 348–358.

[4] N. Patwari, A.O. Hero III, M. Perkins, N.S. Correal, R.J. O'Dea, Relative location estimation in wireless sensor networks, IEEE Transactions on Signal Processing 51 (8) (2003) 2137–2148.

[5] N. Bulusu, J. Heidemann, D. Estrin, GPS-less low cost outdoor localization for very small devices, IEEE Personal Communications Magazine 7 (2000) 28–34.

[6] B.M. Sadler, R.J. Kozick, L. Tong, Multimodal sensor localization using a mobile access point, in: IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP-05), Philadelphia, 2005, pp. 753–756.

[7] D. Niculescu, B. Nath, Ad Hoc positioning system (APS), in: Proceedings of GLOBECOM San Antonio.

[8] A. Savvides, H. Park, M.B. Srivastava, The bits and flops of the n-hop multilateration primitive for node localization problems, in: Proceedings of the First ACM International Workshop on Wireless Sensor Networks and Applications (WSNA-02), ACM Press, New York, 2002, pp. 112–121.

[9] K. Langendoen, N. Reijers, Distributed localization in wireless sensor networks: a quantitative comparison, Computer Networks 43 (4) (2003) 499–518.

[10] M.L. Sichitiu, V. Ramadurai, Localization of wireless sensor networks with a mobile beacon, in: Proceedings of the First IEEE Conference on Mobile Ad-Hoc and Sensor Systems (MASS 2004), 2004.

[11] A. Galstyan, B. Krishnamachari, K. Lerman, S. Pattem, Distributed online localization in sensor networks using a moving target, in: Proceedings of the Third International Symposium on Information Processing in Sensor Networks (IPSN-04), ACM Press, New York, 2004, pp. 61–70.

[12] P.M. Djurić, M. Vemula, M.F. Bugallo, J. Miguez, Non-cooperative localization of binary sensors, in: IEEE Workshop on Statistical Signal Processing, 2005.

[13] P.N. Pathirana, N. Bulusu, A.V. Savkin, S.K. Jha, Node localization using mobile robots in delay-tolerant sensor networks, IEEE Transactions on Mobile Computing 4 (3) (2005) 285–296.

[14] N. Patwari, J.N. Ash, S. Kyperountas, A.O. Hero, R.L. Moses, N.S. Correal, Locating the nodes: cooperative localization in wireless sensor networks, IEEE Signal Processing Magazine 22 (4) (July 2005) 54–69.

[15] A. Savvides, L. Girod, M.B. Srivastava, D. Estrin, Localization in Sensor Networks, in: Wireless Sensor Networks, Kluwer Academic Publisher, Norwell, MA, 2004.

[16] A. Savvides, C.-C. Han, M.B. Srivastava, Dynamic fine-grained localization in Ad-Hoc networks of sensors, in: MOBICOM, 2001, pp. 166–179.

[17] D. Fox, J. Hightower, H. Kauz, L. Liao, D. Patterson, Bayesian techniques for location estimation, in: Proceedings Workshop on Location Aware Computing, October 2003 UBICOMP Conference, Seattle, WA.
[18] V. Ramadurai, M.L. Sichitiu, Localization in wireless sensor networks: a probabilistic approach, in: International Conference on Wireless Networks, 2003, pp. 275–281.
[19] V. Ceveher, R. Chellapa, J. McClellan, Gaussian approximations for energy-based detection and localization in sensor networks, in: IEEE Statistical Signal Processing Workshop, IEEE, 2007.
[20] D. Marinakis, G. Dudek, Probabilistic self-localization for sensor networks, in: AAAI National Conference on Artificial Intelligence, Boston, MA, 2006.
[21] A.T. Ihler, J.W. Fisher III, R.L. Moses, A.S. Willsky, Nonparametric belief propagation for self-localization of sensor networks, IEEE Journal on Selected Areas in Communications 23 (4) (2005) 809–819.
[22] J. Nocedal, S.J. Wright, Numerical Optimization, Springer, NY, 1999.
[23] S.M. Kay, Fundamentals of Statistical Signal Processing: Estimation Theory, Prentice-Hall PTR, Upper Sadle River, NJ 07458, 1993.
[24] A. Sayed, T. Kailath, A state-space approach to adaptive RLS filtering, Signal Processing Magazine, IEEE 11 (3) (1994) 18–60.
[25] J.S. Liu, Monte Carlo Strategies in Scientific Computing, Springer, New York, USA, 2001.
[26] W.J.J. Rey, Introduction to Robust and Quasi-Robust Statistical Methods, Springer, Berlin, 1983.
[27] H.L. Van-Trees, Detection, Estimation, and Modulation Theory, Morgan Kaufmann, Los Altos, CA, 1968.
[28] Y. Rockah, P. Schultheiss, Array shape calibration using sources in unknown locations—part i: Far-field sources, IEEE Transactions on Acoustics, Speech, and Signal Processing 35 (3) (1987) 286–299.
[29] H. Messer, The hybrid Cramer–Rao lower bound—from practice to theory, in: Sensor Array and Multichannel Processing, 2006, pp. 304–307.