Nonparametric Belief Propagation for Self-Localization of Sensor Networks

Alexander T. Ihler*, *Student Member, IEEE,* John W. Fisher III, *Member, IEEE,* Randolph L. Moses, *Senior Member, IEEE,* and Alan S. Willsky, *Fellow, IEEE*

***Abstract*— Automatic self-localization is a critical need for the effective use of ad-hoc sensor networks in military or civilian applications. In general, self-localization involves the combination of absolute location information (e.g. GPS) with relative calibration information (e.g. distance measurements between sensors) over regions of the network. Furthermore, it is generally desirable to distribute the computational burden across the network and minimize the amount of inter-sensor communication. We demonstrate that the information used for sensor localization is fundamentally local with regard to the network topology and use this observation to reformulate the problem within a graphical model framework. We then present and demonstrate the utility of *nonparametric belief propagation* (NBP), a recent generalization of particle filtering, for both estimating sensor locations and representing location uncertainties. NBP has the advantage that it is easily implemented in a distributed fashion, admits a wide variety of statistical models, and can represent multi-modal uncertainty. Using simulations of small- to moderately-sized sensor networks, we show that NBP may be made robust to outlier measurement errors by a simple model augmentation, and that judicious message construction can result in better estimates. Furthermore, we provide an analysis of NBP's communications requirements, showing that typically only a few messages per sensor are required, and that even low bit-rate approximations of these messages can have little or no performance impact.**

I. INTRODUCTION

Improvements in sensing technology and wireless communications are rapidly increasing the importance of sensor networks for a wide variety of application domains [1, 2]. Collaborative networks are created by deploying a large number of low-cost, self-powered sensor nodes of varying modalities (e.g. acoustic, seismic, magnetic, imaging, *etc.*). Sensor localization, i.e. obtaining estimates of each sensor's position as well as accurately representing the uncertainty of that estimate, is a critical step for effective application of large sensor networks. Manual calibration[1] of each sensor may be impractical or even impossible, and equipping every sensor with a GPS receiver or equivalent technology may be cost prohibitive. Consequently, methods of *self-localization* which can exploit relative information (e.g. obtained from received signal strength or time delay between sensors) and a limited amount of global reference information as might be available to a small subset of sensors are desirable. In the wireless sensor network context, localization is further complicated by the need to minimize inter-sensor communication in order to preserve energy resources.

We present a localization method in which each sensor has available noisy distance measurements to neighboring sensors. In the special case that the noise on distance observations is well modeled by a Gaussian distribution, localization may be formulated as a nonlinear least-squares optimization problem. In [3] it was shown that a relative calibration solution which approached the Cramer-Rao bound could be obtained using an iterative optimization approach.

In contrast, we reformulate localization as an inference problem on a graphical model. This allows us to apply nonparametric belief propagation (NBP, [4]), a variant of the popular belief propagation (BP) algorithm [5], to obtain an approximate solution. This approach has several advantages:

- It exploits the local nature of the problem; a given sensor's location estimate depends primarily on information about nearby sensors.
- It naturally allows for a distributed estimation procedure.
- It is not restricted to Gaussian measurement models.
- It produces both an estimate of sensor locations and a representation of the location uncertainties.

The last is notable for random sensor deployments where multi-modal uncertainty in sensor locations is a frequent occurrence. Furthermore, estimation of uncertainty (whether multi-modal or not) provides guidance for expending additional resources in order to obtain more refined solutions.

In the subsequent sections, we describe the sensor localization problem in more detail and relate it to inference in graphical models. In Sections II–III, we formalize the problem and discuss the types of uncertainty which occur in localization. Section IV re-formulates the localization problem as a graphical model, and presents a solution based on the NBP algorithm. We show several empirical examples demonstrating the ability of NBP to solve difficult distributed localization problems. We conclude with three modifications to improve NBP's performance in practical applications: Section VI shows how NBP may be augmented to include an outlier model in the measurement process, and demonstrates its improved robustness to non-Gaussian measurement errors; Section VII presents an alternative sampling procedure which may improve

A. Ihler and A. Willsky are with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology 32-D570, 77 Massachusetts Ave, Cambridge, MA 02139 USA. Tel: (617) 253-4874. Fax: (617) 258-8364. (e-mail: ihler@mit.edu; willsky@mit.edu)

J. Fisher is with the Computer Science & Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 77 Massachusetts Ave, Cambridge, MA 02139 USA. Tel: (617) 253-0788. (e-mail: fisher@csail.mit.edu)

R. Moses is with the Department of Electrical and Computer Engineering, The Ohio State University, 2015 Neil Ave, Columbus, OH 43210 USA. Tel: (614) 292-1325. (e-mail: moses.2@osu.edu)

This research was supported in part by AFOSR grant F49620-00-0362 and by ODDR&E MURI through ARO grant DAAD19-00-0466.

[1]In the context of this paper, we use the term *localization* interchangeably with the more general term *calibration* in sensor networks.

the performance of NBP in systems with limited computational resources; and Section VIII considers the communication costs inherent in a distributed implementation of NBP, and provides simulations to demonstrate the inherent tradeoff between communication and estimate quality.

II. Self-localization of Sensor Networks

This section describes a statistical framework for the sensor network self-localization problem, similar but more general than that given in [6]. We restrict our attention to cases in which individual sensors obtain noisy distance measurements of a (usually nearby) subset of the other sensors in the network. This includes, for example, scenarios in which each sensor is equipped with a wireless and/or acoustic transceiver and distance is estimated by received signal strength or time delay of arrival between sensor locations. Typically this involves a broadcast from each source as all other sensors listen [6, 7].

While the framework we describe is not the most general possible, it is sufficiently flexible to be extended to more complex scenarios. For instance, our method may be easily modified to fit cases in which sources are not co-located with a cooperating sensor, to incorporate direction-of-arrival information (which also necessitates estimation of the orientation of each sensor) [6], or simultaneous estimation of other sensor characteristics such as transmitter power [7].

Let us assume that we have N sensors scattered in a planar region, and denote the two-dimensional location of sensor t by x_t. The sensor t obtains a noisy measurement d_{tu} of its distance from sensor u with some probability $P_o(x_t, x_u)$:

$$d_{tu} = \|x_t - x_u\| + \nu_{tu} \qquad \nu_{tu} \sim p_\nu(x_t, x_u) \tag{1}$$

We use the binary random variable o_{tu} to indicate whether this observation is available, i.e. $o_{tu} = 1$ if d_{tu} is observed, and $o_{tu} = 0$ otherwise. Finally, each sensor t has a (potentially uninformative) prior distribution, denoted $p_t(x_t)$. Thus, the joint distribution is given by

$$p(x_1, \ldots, x_N, \{o_{tu}\}, \{d_{tu}\}) = \prod_{(t,u)} p(o_{tu}|x_t, x_u) \prod_{(t,u):o_{tu}=1} p(d_{tu}|x_t, x_u) \prod_t p_t(x_t) \tag{2}$$

The typical goal of sensor localization is to estimate the maximum likelihood (ML) sensor locations x_t given a set of observations $\{d_{tu}\}$. Of course, there is a distinction between the individual ML estimates of each x_t versus the ML estimate of all $\{x_t\}$ jointly. For this work, it is convenient to select the former; in a discrete system, this would correspond to minimizing the bit-error rate (as opposed to an "all-or-nothing" sequence-error probability).

The estimated distances d_{ut} and d_{tu} may be different, and it is even possible to have $o_{ut} \neq o_{tu}$ (indicating that only one of sensors u and t can observe the other). It will later be convenient to symmetrize these relationships, a process which involves exchanging information between any pair of sensors u, t which observe either d_{tu} or d_{ut}; this may involve multi-hop message routing or other communication protocols which are beyond the scope of this paper. For Gaussian p_ν, the two estimates are simply averaged. However, for arbitrary distributions the process of using both measurements, while not difficult, becomes notationally cumbersome; we thus assume in the development that $o_{ut} = o_{tu}$ and $d_{ut} = d_{tu}$, and include remarks on the differences when this is not the case.

Also, the amount of prior location information may be almost nonexistent. In this case, we may wish to solve for a *relative* sensor geometry (versus estimating the sensor locations with respect to some absolute frame of reference) [3]. Given only the relative measurements $\{o_{tu}, d_{tu}\}$, the sensor locations x_t may only be solved up to an unknown rotation, translation, and negation (mirror image) of the entire network. We avoid ambiguities in the relative calibration case by assuming priors which enforce known conditions for three sensors (denoted s_1, s_2, s_3):

1) *Translation*: s_1 has known location (taken to be the origin: $x_1 = [0; 0]$)
2) *Rotation*: s_2 is in a known direction from s_1 ($x_2 = [0; a]$ for some $a > 0$)
3) *Negation*: s_3 has known sign ($x_3 = [b; c]$ for some b, c with $b > 0$).

Typically s_1, s_2, s_3 are taken to be spatially close to each other in the network. When our goal is absolute calibration (calibration with respect to a known coordinate reference), we simply assume that the prior distributions $p_t(x_t)$ contain sufficient information to resolve this ambiguity. The sensors with significant prior information (or $s_{1\ldots3}$ for relative calibration) are referred to as *anchor* nodes.

In general finding the ML sensor locations is a complex nonlinear optimization problem. If the uncertainties p_ν, p_t above are Gaussian and P_o is assumed constant, ML joint estimation of the $\{x_t\}$ reduces to a nonlinear least-squares optimization [6]. In the case that we observe distance measurements between *all* pairs of sensors (i.e. $P_o(\cdot) \equiv 1$), this also corresponds to a well studied distortion criterion ("STRESS") in multidimensional scaling problems [8]. However, for large-scale sensor networks, it is reasonable to assume that only a subset of pairwise distances will be available, primarily between sensors which are in the same region. One model (proposed by [3]) assumes that the probability of detecting nearby sensors falls off exponentially with squared distance:

$$P_o(x_t, x_u) = \exp\left(-.5 \, \|x_t - x_u\|^2 / R^2\right) \tag{3}$$

We use (3) in our example simulations, though alternative forms are equally simple to incorporate into our framework, leaving open the possibility of estimating P_o from training data, if available; such experiments have already been performed for certain sensor types and measurement methods [7].

A large number of methods have been previously proposed to estimate sensor locations [7, 9–13]. An exhaustive categorization is beyond the scope of this paper; here we are able to list only a few. For better or worse, many of these methods eschew a statistical interpretation in favor of computational simplicity. Some examples include approximating the unobserved distances and applying classical multidimensional scaling [8], multi-lateration [12], or other techniques [9]. Other approaches search for locations which satisfy convex distance constraints [11]. Yet another method heuristically minimizes the rank of the distance matrix [14].

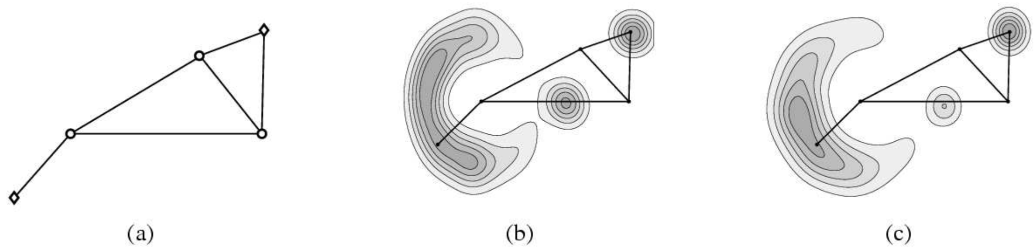


Fig. 1. Example sensor network. (a) Sensor locations are indicated by symbols and distance measurements by connecting lines. Calibration is performed relative to the three sensors drawn as circles. (b) Marginal uncertainties for the two remaining sensors (one bimodal, the other crescent-shaped), indicating that their estimated positions may not be reliable. (c) Estimates of the same marginal distributions using NBP.

However, these algorithms often lack a direct statistical interpretation, and as one consequence rarely provide an estimate of the remaining uncertainty in each sensor location. Iterative least-squares methods such as [6, 10, 12, 13] *do* have a statistical interpretation, but assume a Gaussian model for all uncertainty, which may be questionable in practice. As we discuss in Section III, non-Gaussian uncertainty is a common occurrence in sensor localization problems. In consequence, the Cramer-Rao bound may be an overly optimistic characterization of the actual sensor location uncertainty, particularly for multi-modal distributions. Estimating which, if any, sensor positions are unreliable is an important task when parts of the network are under-determined. Furthermore, Gaussian noise models are often inadequate for real-world noise, which may have some fraction of highly erroneous (outlier) measurements.

In this paper we pose the sensor localization problem as inference on a graphical model, and propose an approximate solution making use of a recent sample-based message-passing estimation technique called *nonparametric belief propagation* (NBP). NBP allows us to apply the general, flexible statistical formulation described above, and can capture the complex uncertainties which occur in localization of sensor networks.

III. Uncertainty in sensor location

The sensor localization problem as described in the previous section involves the optimization of a complex nonlinear likelihood function. However, it is often desirable to also have some measure of confidence in the estimated locations. Even for Gaussian measurement noise ν, the nonlinear relationship of inter-sensor distances to sensor positions results in highly non-Gaussian uncertainty of the sensor location estimates.

For sufficiently small networks it is possible to use Gibbs sampling [15] to obtain samples from the joint distribution of the sensor locations. In Fig. 1(a), we show an example network with five sensors. Calibration is performed relative to measurements from the three sensors marked by circles. A line is shown connecting each pair of sensors which obtain a distance measurement. Contour plots of the marginal distributions for the two remaining sensors are given in Fig. 1(b); these sensors do not have sufficient information to be well-localized, and in particular have highly non-Gaussian, multi-modal uncertainty (suggesting the utility of a nonparametric representation). Although we defer the details of NBP to Section IV-C, for comparison an estimate of the same marginal uncertainties performed using NBP is displayed in Fig. 1(c).

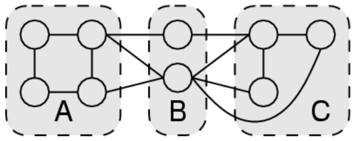


Fig. 2. Graph separation and conditional independence of variables: all paths between the sets A and C pass through B, implying $p(x_A, x_C|x_B) = p(x_A|x_B)p(x_C|x_B)$.

IV. Graphical Models for Localization

Graphical models are a popular means of encapsulating the factorization of a probability distribution, enabling the application of a number of simple, general algorithms for exact or approximate inference [5, 16, 17]. Interpreting the distribution (2) as a graphical model allows one in principle to apply any of a number of inference algorithms [16, 17], of which belief propagation (BP) is perhaps the best-known. In practice, however, we shall see that the typical, discrete implementation of BP has an unacceptably high computational cost. However, a particle-based approximation to BP, called nonparametric belief propagation (NBP), results in a more tractable algorithm.

A. Graphical Models

An undirected graphical model consists of a set of vertices $V = \{v_t\}$ and a collection of edges $e_{tu} \in E$. Two vertices v_t, v_u are *connected* if there exists an edge $e_{tu} \in E$ between them, and a subset $A \subset V$ is *fully connected* if all pairs of vertices $v_t, v_u \in A$ are connected. Each vertex v_t is also associated with a random variable x_t, and the edges of the graph are used to indicate conditional independence relationships through *graph separation*. Specifically, if every path between two sets $A, C \subset V$ passes through a set $B \subset V$ (see Fig. 2), then the sets of random variables $x_A = \{x_a : v_a \in A\}$ and $x_C = \{x_c : v_c \in C\}$ are independent given $x_B = \{x_b : v_b \in B\}$. This relationship may also be written in terms of the joint distribution: $p(x_A, x_B, x_C) = p(x_B)p(x_A|x_B)p(x_C|x_B)$.

The relationship between the graph and joint distribution may be quantified in terms of *potential functions* ψ which are defined over the graph's cliques (the fully connected subsets of V), which we denote by Q [16]:

$$p(x_1, \dots, x_N) \propto \prod_{\text{cliques } Q} \psi_Q(\{x_i : i \in Q\}) \qquad (4)$$

Again taking x_t to be the location of sensor t, we may immediately define potential functions which equate (4) to the joint distribution (2). Notably, this only requires functions defined over single nodes and pairs of nodes. Take

$$\psi_t(x_t) = p_t(x_t) \tag{5}$$

to be the single-node potential at each node v_t, and define the pairwise potential between nodes v_t and v_u as

$$\psi_{tu}(x_t, x_u) = \begin{cases} P_o(x_t, x_u)\ p_\nu(d_{tu} - \|x_t - x_u\|) & \text{if } o_{tu} = 1 \\ 1 - P_o(x_t, x_u) & \text{otherwise} \end{cases} \tag{6}$$

We make no distinction between ψ_{tu} and ψ_{ut}, only one of which[2] appears in the product (4). The joint posterior likelihood of the x_t is then

$$p(x_1, \ldots, x_N | \{o_{tu}, d_{tu}\}) \propto \prod_t \psi_t(x_t) \prod_{t,u} \psi_{tu}(x_t, x_u) \tag{7}$$

Notice also that for non-constant P_o *every* sensor t has some information about the location of each sensor u (i.e. there is *some* information contained in the fact that two sensors do not observe a distance between them, namely that they should prefer to be far from each other). This is a *probabilistic* relationship, and thus can account for the fact that sometimes (such as in the case of physical barriers) sensors which are near may still not observe each other.[3]

Unfortunately, fully connected graphs are very difficult for most inference algorithms, and thus it behooves us to approximate the exact model. Experimentally (see [18]) it appears that there is little loss in information by discarding the interactions between nodes which are far apart, in the following sense. Let the "1-step" graph be the graph in which we join two nodes t, u if and only if we observe a distance d_{tu} (so that $o_{tu} = 1$). We create the "2-step" graph by also adding an edge between t and u if we observe d_{tv} and d_{vu} for some sensor v, but not d_{tu}, and may extend this definition to "3-step" and so forth. Edges for which $o_{tu} = 1$ we refer to as *observed*; those with $o_{tu} = 0$ we call *unobserved* edges. Note that the "1-step" graph is exact if P_o is a constant, since in this case the unobserved edges offer no additional information.

There is also a convenient relationship between the *statistical* and *communications* graph in localization. Specifically, distance measurements are only obtained for sensor pairs which have communications links[4]. Thus, messages along observed edges may be communicated directly, while messages along unobserved edges may require a multi-hop forwarding protocol (with 2-step edges requiring at most 2 hops, *etc.*).

[2]The definition of ψ is slightly more complicated for asymmetric measurements, since to obtain a self-consistent undirected graphical model we require both t and u to know and agree on $\psi_{tu} = \psi_{ut}$, which will thus involve all four quantities $o_{tu}, o_{ut}, d_{tu}, d_{ut}$.

[3]The effect of these constraints is similar to, but less strict than, that achieved by approximating unobserved distances by shortest paths [12], and to the non-convex constraints mentioned in [11]. This has the additional benefit of being less vulnerable to distortion (as observed by [12]) when the sensor configuration is not entirely convex.

[4]While technically the time-varying nature of these links means that communications may not be entirely reliable, we ignore this subtlety and assume that, over the short period of time in which localization is performed, the communications graph is stable.

B. Belief Propagation

Having defined a graphical model for sensor localization, we now turn to the task of estimating the sensor locations. Inference among variables in a graphical model is a problem which has received considerable attention. Although exact inference in general graphs can be NP-hard, approximate inference algorithms such as loopy belief propagation (BP) [5, 19] produce excellent empirical results in many cases. BP can be formulated as an iterative, local message passing algorithm, in which each node v_t computes its "belief" about its associated variable x_t, communicates this belief to and receives messages from its neighbors, then updates its belief and repeats. In the wireless localization context, such algorithms are apropos.

The computations performed at each iteration of BP are relatively simple. In integral form, each node v_t computes its belief about x_t (a normalized estimate of the posterior likelihood of x_t) at iteration n by taking a product of its local potential ψ_t with the messages from its neighbors, denoted Γ_t:

$$\hat{p}^n(x_t) \propto \psi_t(x_t) \prod_{u \in \Gamma_t} m^n_{ut}(x_t) \tag{8}$$

Typically the (arbitrary) proportionality constants are chosen to normalize $\hat{p}^n$, i.e. $\int \hat{p}^n(x_t) dx_t = 1$. The messages m_{tu} from the node v_t to v_u are computed in a similar fashion:

$$\begin{aligned} m^n_{tu}(x_u) &\propto \int \psi_{tu}(x_t, x_u) \psi_t(x_t) \prod_{v \in \Gamma_t \setminus u} m^{n-1}_{vt}(x_t)\, dx_t \\ &\propto \int \psi_{tu}(x_t, x_u) \frac{\hat{p}^{n-1}(x_t)}{m^{n-1}_{ut}(x_t)}\, dx_t \end{aligned} \tag{9}$$

One appealing consequence of using a message-passing inference method and assigning each vertex of the graph to a sensor in the network is that computation is naturally distributed. Each node (sensor) performs computations using information sent by its neighbors, and disseminates the results, as described in Alg. 1. This process is repeated until some convergence criterion is met, after which each sensor is left with an estimate of its location and uncertainty.

Alg. 1 also uses a suggestion of [20], in which a reweighted marginal distribution $\hat{p}^n(x_t)$ is used as an estimate of the product of messages (9). In addition to the advantages discussed in [20], this has a hidden communication benefit—all messages from t to its neighbors Γ_t may be communicated *simultaneously* via a broadcast step. This is because the message from t to each neighbor $u \in \Gamma_t$ is a function of the marginal $\hat{p}^{n-1}(x_t)$, the previous iteration's message from u to t, and the compatibility ψ_{tu} (which depends only on the observed distance between t and u). Since the latter two quantities are also known at node u, t may simply communicate its estimated marginal $\hat{p}^n(x_t)$ to all its neighbors, and allow each u to deduce the rest.

C. Nonparametric Belief Propagation

The BP update and belief equations (8)-(9) are easily computed in closed form for discrete or Gaussian likelihood functions; unfortunately neither discrete nor Gaussian BP is well-suited to localization, since even the two-dimensional space in which the x_t reside is too large to accommodate an

Sensor Self-Localization with BP
Initialization:
- Each sensor obtains local information $p_t(x_t)$, if available.
- Obtain distance estimates:
  - Broadcast sensor id & listen for other sensor broadcasts
  - Estimate distance d_{tu} for any received sensor IDs
  - Communicate with observed neighbors to symmetrize distance estimates
- Initialize $m_{ut} \equiv 1$ and $p_t^0 = p_t$ for all u, t.

Belief Propagation: for each sensor t
- Broadcast $\hat{p}^n(x_t)$ to neighbors; listen for neighbors' broadcasts
- Compute m_{ut}^{n+1} from m_{tu}^n and $\hat{p}^n(x_u)$ using (9)
- Compute new marginal estimate $\hat{p}^{n+1}(x_t)$ via (8)
- Repeat until sufficiently converged (see Sec. VIII)

Alg. 1: Belief propagation for sensor self-localization.

efficient discretized estimate[5], and the presence of nonlinear relationships and potentially highly non-Gaussian uncertainties makes Gaussian BP undesirable as well. The development of a version of BP making use of particle-based representations, called nonparametric belief propagation (NBP) [4], enables the application of BP to inference in sensor networks.

In NBP, each message is represented using either a sample-based density estimate (as a mixture of Gaussians) or as an analytic function. Both types are necessary for the sensor localization problem. Messages along observed edges are represented by samples, while messages along unobserved edges must be represented as analytic functions since often $1 - P_o(x_t, x_u)$ is not normalizable (typically tending to 1 as $\|x_t - x_u\|$ becomes large) and thus is poorly approximated by any finite set of samples. The belief and message update equations (8)-(9) are performed using stochastic approximations, in two stages: first, drawing samples from the estimated marginal $\hat{p}(x_t)$, then using these samples to approximate each outgoing message m_{tu}. We discuss each of these steps in turn, and summarize the procedure in Alg. 2.

Given M weighted samples $\{w_t^{(i)}, x_t^{(i)}\}$ from the marginal estimate $\hat{p}_t^n(x_t)$ obtained at iteration n, computing a Gaussian mixture estimate of the outgoing message from t to u is relatively simple. We first consider the case of observed edges. Given a measurement of the distance d_{tu}, each sample $x_t^{(i)}$ is moved in a random direction by d_{tu} plus noise[6]:

$$m_{tu}^{(i)} = x_t^{(i)} + (d_{tu} + \nu^{(i)})[\sin(\theta^{(i)}); \cos(\theta^{(i)})]$$
$$\theta^{(i)} \sim U[0, 2\pi) \qquad \nu^{(i)} \sim p_\nu \tag{10}$$

The samples are then weighted by the remainder of (9), $w_{tu}^{(i)} = w_t^{(i)} \cdot P_o(m_{tu}^{(i)})/m_{ut}(x_t^{(i)})$, and (as is typical in kernel density estimation) a single covariance Σ_{tu} is assigned to all samples. There are a number of possible techniques for choosing the covariance Σ_{tu}; one simple method is the *rule of thumb* estimate [23], given by computing the (weighted) covariance of the samples

$$\text{Covar}[m_{tu}^{(i)}] = \sum_{i,j} w_{tu}^{(i)} w_{tu}^{(j)} (m_{tu}^{(i)} - \bar{m})(m_{tu}^{(j)} - \bar{m})^T \tag{11}$$

(where $\bar{m} = \sum_i w_{tu}^{(i)} m_{tu}^{(i)}$) and dividing by $M^{\frac{1}{3}}$. A simple and computationally efficient alternative has been proposed by [24]; if the uncertainty added by ψ_{tu} is Gaussian, we may simply use the mean ($\nu^{(i)} = 0$) and apply the covariance of the Gaussian uncertainty to each sample ($\Sigma_{tu} = \sigma_\nu^2 I$). This method may also be extended to small Gaussian mixtures, and works well when the number of particles is sufficiently large.

As stated previously, messages along unobserved edges (pairs t, u for which d_{tu} is not observed) are represented using an analytic function. Using the probability of detection P_o and samples from the marginal at x_t, an estimate of the outgoing message to u is given by

$$m_{tu}(x_u) = 1 - \sum_i w_t^{(i)} P_o(x_u - x_t^{(i)}) \tag{12}$$

which is easily evaluated for any analytic model of P_o.

Estimation of the marginal $\hat{p}^n = \psi_t \prod m_{ut}$ is potentially more difficult. Since it is the product of several Gaussian mixtures, computing $\hat{p}^n$ exactly is exponential in the number of incoming messages. However, efficient methods of drawing samples from the product of several Gaussian mixture densities is investigated in [25]; in this work we primarily use a technique called *mixture importance sampling*. Denote the set of neighbors of t having *observed* edges to t by Γ_t^o. In order to draw M samples, we create a collection of $k \cdot M$ weighted samples (where $k \geq 1$ is a parameter of the sampling algorithm) by drawing $\frac{kM}{|\Gamma_t^o|}$ samples from each message $m_{ut}, u \in \Gamma_t^o$ and assigning each sample a weight equal to the ratio $\prod_{v \in \Gamma_t} m_{vt} / \sum_{v \in \Gamma_t^o} m_{vt}$. We then draw M values from this collection with probability proportional to their weight (with replacement), yielding equal-weight samples drawn from the product of all incoming messages. Computationally, this requires $\mathcal{O}(k|\Gamma_t|M)$ operations per marginal estimate.

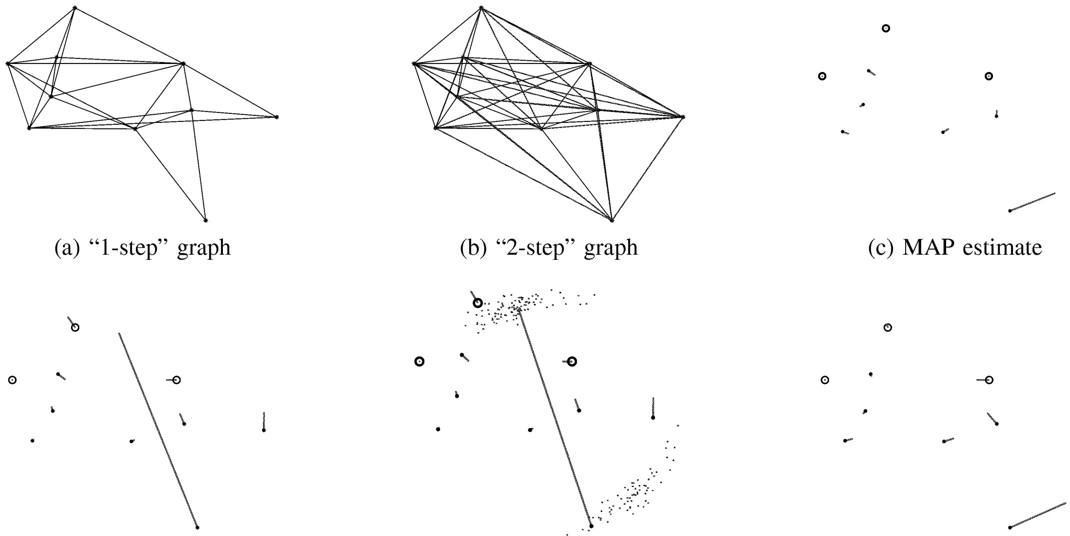
V. Empirical Calibration Examples

We show two example sensor networks to demonstrate NBP's utility. All the networks in this section have been generated by placing N sensors at random with spatially uniform probability in an $L \times L$ area, and letting each sensor observe its distance from another sensor (corrupted by Gaussian noise with variance σ_ν^2) with probability given by (3). We investigate the *relative* calibration problem, in which the sensors are given no absolute location information; the anchor nodes are indicated by open circles. These simulations used $M = 200$ particles and underwent three iterations of the sequential message schedule described in Section VIII; each iteration took less than 1 second per node on a P4 workstation.

The first example (Fig. 3(a)) shows a small graph ($N = 10$), generated using $R/L = .2$ and noise $\sigma_\nu/L = .02$; this made the average measured distance about $.33L$, and each sensor observed an average of 5 neighbors. One sensor (the lowest) has significant multi-modal location uncertainty, since it observes only two measurements. The joint MAP configuration

[5] For M bins per dimension, calculating each message requires $\mathcal{O}(M^4)$ operations, though there has been some work to improve this [21, 22].

[6] If p_ν is non-Gaussian and $d_{tu} \neq d_{ut}$, we may draw some samples according to each of $p(x_u|x_t, d_{tu})$ and $p(x_u|x_t, d_{ut})$ and weight by the influence of the other observation.



(a) "1-step" graph (b) "2-step" graph (c) MAP estimate

(d) "1-step" NBP estimates (e) "1-step" NBP marginal (f) "2-step" NBP estimates

Fig. 3. (a) A small (10-sensor) graph with edges denoting observed pairwise distances; (b) the same network with "2-step" unobserved relationships also shown. Calibration is performed relative to the sensors drawn as open circles. (c) A centralized estimate of the MAP solution shows generally similar errors (lines) to (d), NBP's approximate (marginal maximum) solution. However, NBP's estimate of uncertainty (e) for the poorly-resolved sensor displays a clear bi-modality. Adding "2-step" potentials (f) results in a reduction of the spurious mode and an improved estimate of location.

Compute NBP messages: Given M weighted samples $\{w_t^{(i)}, x_t^{(i)}\}$ from $\hat{p}^n(x_t)$, construct an approximation to $m_{tu}^n(x_u)$ for each neighbor $u \in \Gamma_t$:

- If $o_{tu} = 1$ (we observe inter-sensor distance d_{tu}), approximate with a Gaussian mixture:
  - Draw random values for $\theta^{(i)} \sim U[0, 2\pi)$ and $\nu^{(i)} \sim p_\nu$
  - Means: $m_{tu}^{(i)} = x_t^{(i)} + (d_{tu} + \nu^{(i)})[\sin(\theta^{(i)}); \cos(\theta^{(i)})]$
  - Weights: $w_{tu}^{(i)} = P_o(m_{tu}^{(i)})\, w_t^{(i)} / m_{ut}^{n-1}(x_t^{(i)})$
  - Variance: $\Sigma_{tu} = M^{-\frac{1}{3}} \cdot \text{Covar}[m_{tu}^{(i)}]$
- Otherwise, use the analytic function:
  - $m_{tu}(x_u) = 1 - \sum_i w_t^{(i)} P_o(x_u - x_t^{(i)})$

Compute NBP marginals: Given several Gaussian mixture messages $m_{ut}^n = \{m_{ut}^{(i)}, w_{ut}^{(i)}, \Sigma_{ut}\}, u \in \Gamma_t^o$, sample from $\hat{p}^{n+1}(x_t)$:

- For each observed neighbor $u \in \Gamma_t^o$,
  - Draw $\frac{kM}{|\Gamma_t^o|}$ samples $\{x_t^{(i)}\}$ from each message m_{ut}^n
  - Weight by $w_t^{(i)} = \prod_{v \in \Gamma_t} m_{vt}^n(x_t^{(i)}) / \sum_{v \in \Gamma_t^o} m_{vt}^n(x_t^{(i)})$
- From these kM locations, re-sample by weight (with replacement) M times to produce M equal-weight samples.

Alg. 2: Using NBP to compute messages and marginals for sensor localization.

is shown in Fig. 3(c) while the "1-step" NBP estimate is shown in Fig. 3(d). Comparison of the error residuals would indicate that NBP has significantly larger error on the sensor in question. However, this is mitigated by the fact that NBP has a representation of the marginal uncertainty (shown in Fig. 3(e)) which accurately captures the bi-modality of the sensor location, and which could be used to determine that the location estimate is questionable. Additionally, exact MAP uses more information than "1-step" NBP. We approximate this information by including some of the unobserved edges ("2-step" NBP). The result is shown in Fig. 3(f); the error residuals are now comparable to the exact MAP estimate.

While the previous example illustrates some important details of the NBP approach, our primary interest is in automatic calibration of moderate- to large-scale sensor networks with sparse connectivity. We examine a graph of a network with 100 sensors generated with $R/L = .08$ (giving an average of about 9 observed neighbors) and $\sigma_\nu/L = .005$, shown in Fig. 4. For problems of this size, computing the true MAP locations is considerably more difficult. The iterative nonlinear minimization of [3] converges slowly and is highly dependent on initialization. As a benchmark to illustrate the best possible performance, an idealized estimate in which we initialize using the *true* locations is shown in Fig. 4(c). In practice, we cannot expect to perform this well; starting from a more realistic value (initialization given by classical MDS [8]) finds the alternate local minimum shown in Fig. 4(d). The "1-step" and "2-step" NBP solutions are shown in Fig. 4(e)-(f). Errors due to multi-modal uncertainty similar to those discussed previously arise for a few sensors in the "1-step" case. Examination of the "2-step" solution shows that the errors are comparable to the estimate with an idealized initialization.

In the "2-step" examples above, we have included *all* "2-step" edges, but this is often not required. The sensors which require this additional information are typically those with too few observed neighbors, and we could achieve similar results by including only "2-step" edges which are incident on a node with fewer than, for example, four observed edges.

VI. Modeling Non-Gaussian Measurement Noise

It is straightforward to change the form of the noise distribution p_ν so long as sampling remains tractable. This may be used to accommodate alternative distance noise models such as the log-normal model of [10], as might arise when distance

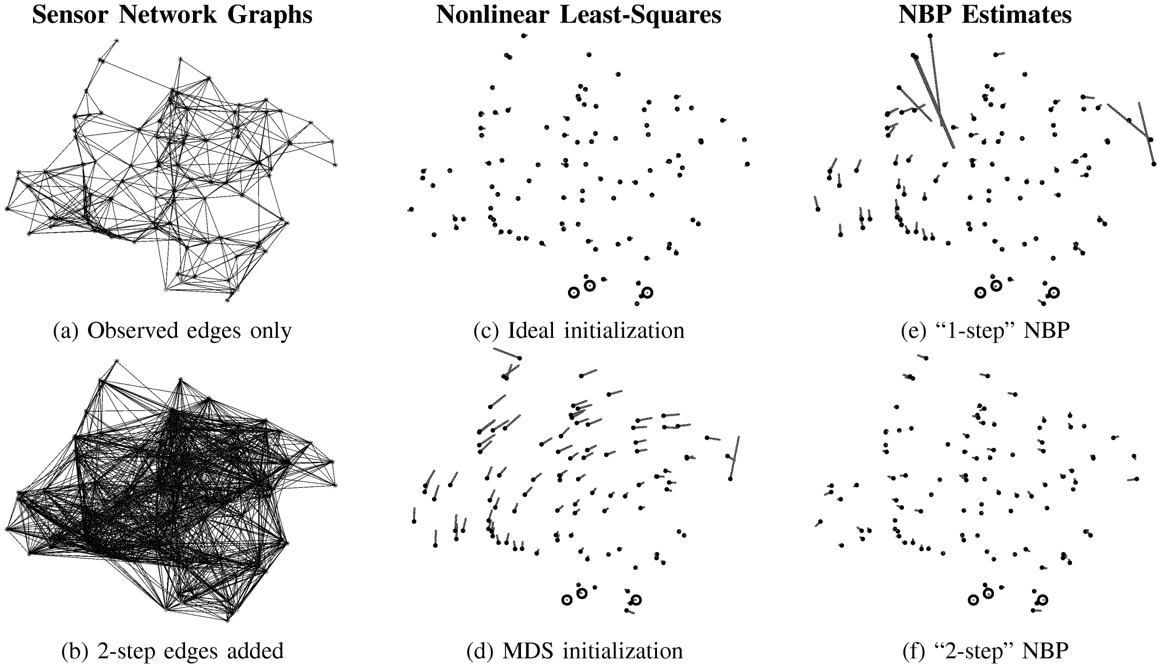


Fig. 4. Large (100-node) example sensor network. (a-b) 1- and 2-step edges. Even in a centralized solution we can at best hope for (c) the local minimum closest to the true locations; a more realistic initialization (d) yields higher errors. NBP (e-f) provides similar or better estimates, along with uncertainty, and is easily distributed. Calibration is performed relative to the three sensors shown as open circles.

between sensor pairs is estimated using the received signal strength, or models which have been learned from data [7].

Although this fact can also be used to model the presence of a broad outlier process, the form of NBP's messages as Gaussian mixtures provides a more elegant solution. We augment the Gaussian mixtures in each message by a single, high-variance Gaussian to approximate an outlier process in the uncertainty about d_{tu}, in a manner similar to [24]. To be precise, we add an extra particle to each outgoing message, centered at the mean of the other particles and with weight and variance chosen to model the expected outliers, e.g. weight equal to the probability of an outlier, and standard deviation sufficiently large to cover the expected support of P_o. Direct approximation of the outlier process requires fewer particles than naive sampling to adequately represent the message, and thus is also more computationally efficient.

Fig. 5(a) shows the same small ($N = 10$) "1-step" network examined in Fig. 3 but with several additional distance measurements (indicated as lines), on which we have introduced a single outlier measurement (the dashed line). We again perform calibration relative to the three sensors shown as circles. If we possessed an oracle which allowed us to detect and discard the erroneous measurement, the optimal sensor locations can be found using an iterative nonlinear least-squares optimization [3]; the residual errors after this procedure (for a single noise realization) are shown in Fig. 5(b). However, with the outlier measurement present, the same procedure results in a large distortion in the estimates of some sensor locations (Fig. 5(c)). NBP, by virtue of the measurement outlier process discussed in Section IV-C, remains robust to this error and produces the near-optimal estimate shown in Fig. 5(d).

In order to provide a measure of the robustness of NBP in the presence of non-Gaussian (outlier) distance measurements, we perform Monte Carlo trials, keeping the same sensor locations and connectivity used in Fig. 5(a) but introducing different sets of observation noise and outlier measurements. At every trial, each distance measurement is replaced with probability $.05$ by a value drawn uniformly in $[0, L]$. As there are 37 measurements in the network, on average approximately two outlier measurements are observed in each trial. We then measure the number of times each sensor's estimated location is within distance r of its true location, as a function of r/L. We repeat the same experiments for two noise levels, $\sigma_\nu/L = .02$ and $\sigma_\nu/L = .002$. The curves are shown in Fig. 6 for both NBP and nonlinear least-squares estimation. As can be seen, NBP provides an estimate which is more often "nearby" to the true sensor location, indicating its increased robustness to the outlier noise; this becomes even more prominent as the σ_ν becomes small and the outlier process begins to dominate the total noise variance. Both methods asymptote around 90%, indicating the probability that the outlier process completely overwhelms the information at one or more nodes.

However, Fig. 6 understates the advantages of NBP for this scenario. NBP also provides an estimate of the *uncertainty* in sensor position; trials resulting in large errors also display highly uncertain (often bimodal) estimates for the sensor locations in question, as in Fig. 1. Thus, in addition to providing a more robust estimate of sensor location, NBP also provides a measure of the reliability of each estimate.

VII. Parsimonious Sampling

We may also apply techniques from importance sampling [26, 27] in order to improve the small-sample performance of NBP, which may play an important part of reducing

(a) (b) (c) (d)

Fig. 5. (a) A small (10-sensor) graph and the observable pairwise distances; calibration is performed relative to the location of the sensors shown in green. One distance (shown as dashed) is highly erroneous, due to a measurement outlier. (b) The MAP estimate of location, discarding the erroneous measurement. (c) A nonlinear least-squares estimate of location is highly distorted by the outlier; (d) NBP is robust to the error by inclusion of a measurement outlier process in the model.

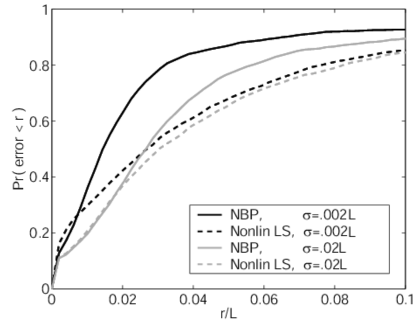


Fig. 6. Monte Carlo localization trials on the sensor network in Fig. 5(a). We measure the probability of a sensor's estimated location being within a radius r of its true location (normalized by the region size L), with noise $\sigma_\nu = .02L$ and $.002L$ for both NBP and nonlinear least-squares, indicating NBP's superior performance in the presence of outlier measurements.

Using previous iterations' angular information: Perform NBP as in Alg. 2, but at iteration $n > 1$, replace $\theta^{(i)} \sim U[0, 2\pi)$ by:

- Draw samples $\tilde{x}_t^{(i)} \sim \hat{p}^{n-1}(x_t), \quad \tilde{x}_u^{(i)} \sim \hat{p}^{n-1}(x_u)$
- Construct a kernel density estimate p_θ using $\tilde{\theta}^{(i)} = \arctan(\tilde{x}_u^{(i)} - \tilde{x}_t^{(i)}) \quad (\tilde{\theta} \in [-\pi, \pi])$
  - To approximate 2π-periodicity, construct p_θ using samples at $\tilde{\theta}^{(i)} + \{2\pi, 0, -2\pi\}$
- Draw $\theta^{(i)} \sim p_\theta, \theta \in [-\pi, \pi]$
- Calculate $w_{tu}^{(i)}$ in a manner similar to Alg. 2, but with additional weighting to cancel p_θ:
  - $w_{tu}^{(i)} = \frac{P_o(m_{tu}^{(i)})\ w_t^{(i)}}{m_{ut}^{n-1}(x_t^{(i)})} \frac{1}{p_\theta(\theta^{(i)})}$

Alg. 3: Using an alternative angular proposal distribution for NBP. The previous iteration's marginals may be used to estimate their relative angle, and better focus samples on the region of importance. The estimate is made asymptotically equivalent to that of Alg. 2 by importance weighting.

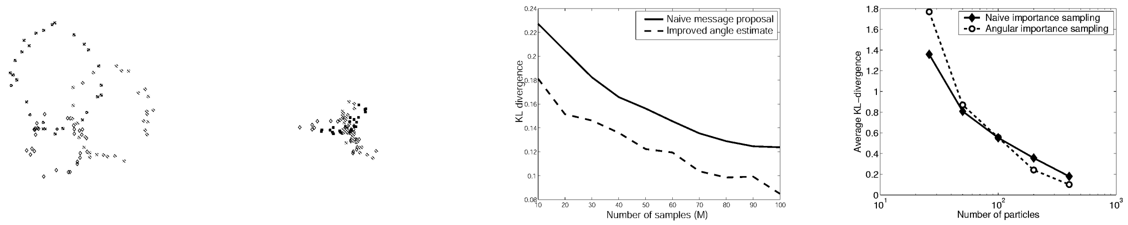
its computational burden. In Alg. 2, the outgoing messages are computed via an importance sampling procedure to estimate (9). In particular, samples are drawn from an approximation to (9) (called the *proposal distribution* in importance sampling literature), then re-weighted so as to asymptotically represent the target distribution (9).

So long as the proposal distribution f is absolutely continuous with respect to the target distribution g (meaning $g(x) > 0 \Rightarrow f(x) > 0$), we are guaranteed that, for a sufficiently large sample size M we can obtain samples which are representative of g by drawing samples from f and weighting by g/f. However, the sample size M is limited by computational power, and as is well-known in particle filtering the low-sample performance of any such approximation is strongly influenced by the quality of the proposal distribution [26, 27]. In general, one takes f to be as close as possible to g while remaining tractable for sampling. We accomplished this for (9) by drawing samples from the marginal (8), weighting by the remainder, and moving the particles in a random direction θ by the observed distance d_{tu} plus noise.

However, in the context of belief propagation, a good proposal distribution is one which allows us to accurately estimate the portions of m_{tu} which contribute to the product $p_u = \prod_s m_{su}$. We would like to use our limited representative power on parts of the message which overlap with other incoming messages, and any additional knowledge of $p(x_u)$ may be used to focus samples in the correct region [20].

One alternative proposal distribution involves utilizing previous iterations' information to determine the angular direction to neighboring sensors. Rather than estimating a ring-like distribution at each iteration (most of which is ignored as it does not overlap with any other rings), successive estimates are improved by estimating smaller and smaller arcs located in the region of interest. A simple procedure implementing this idea is given in Alg. 3. In particular, we use samples from the marginal distributions computed at the previous iteration to form a density estimate p_θ of the relative direction θ, draw samples from p_θ, and weight them by $\frac{1}{p_\theta}$ so as to cancel the asymptotic effect of drawing samples from p_θ rather than uniformly. The process requires estimating a density which is 2π-periodic; this is accomplished by sample replication [23].

We first demonstrate the potential improvement on a small example of only four sensors. Fig. 7(a)-(b) shows example messages from three sensors to a fourth, with $M = 30$ particles. Using the additional angular information results in the samples being clustered in the region of the product, effectively similar to a larger value of M. To compare both methods' performance, we first construct the marginal estimate using a large-M approximation ($M = 1000$), and compare (in terms of KL-divergence) to the results of running NBP with fewer samples ($10 \leq M \leq 100$) using both naive sampling ($\theta \sim U[0, 2\pi)$) and Alg. 3. The results are shown in Fig. 7(c);



(a) (b) (c) (d)

Fig. 7. By using an alternate proposal distribution during NBP's message construction step, we may greatly improve the fidelity of the messages. (a) Naive (uniform) sampling in angle produces ring-shaped messages; however, (b) using previous iterations' information we may preferentially draw samples from the useful regions. Monte Carlo trials (c) show the improvement in terms of average K-L divergence of the sensor's estimated marginal (from an estimate performed with $M = 1000$ samples) as a function of the number of samples M used. (d) In a larger (10-node) network, we begin to observe the effects of bias: for sufficiently large M performance improves, but for small M we may become overconfident in a poor estimate.

as expected, we find that Alg. 3 concentrates more samples in the region of interest, reducing the estimate's KL divergence.

As noted in [20], however, by re-using previous iterations' information we run the risk of biasing our results. The results of a more realistic situation are shown in Fig. 7(d)—performing the same comparison on for a relative calibration of the 10-node sensor network (shown in Fig. 3(b) reveals the possibility of biased results. When the number of particles is sufficiently large ($M \geq 100$), we observe the same improvement as seen in the 4-node case. However, for very few particles ($M = 25$), we see that it is possible for our biased sampling method to reinforce incorrect estimates, ultimately worsening performance.

VIII. Incorporating communications constraints

Communications constraints are extremely important for battery-powered, wireless sensor networks; it is one of the primary factors determining sensor lifetime. There are a number of factors which influence the communications cost of a distributed implementation of NBP. These include

1) *Resolution*, β, of all fixed- (or floating-) point values.
2) *Number of iterations* performed
3) *Schedule*—the order in which sensors transmit
4) *Approximation*—the fidelity to which the marginal estimates are communicated between sensors
5) *Censoring*—sensors may save energy by electing not to send a marginal which is "sufficiently similar" to the previous iteration's marginal.

All these aspects are, of course, interrelated, and also influence the quality of any solution obtained; often their effects are difficult to separate. Note that the number of particles M used for estimating each message and marginal influences only computational complexity. The following experiments used $M = 200$ samples per message and marginal estimate, with $k = 5$ times oversampling in the product computation.

Due to space constraints, we do not consider resolution or similarity-based censoring here. We assume the resolution is sufficiently high to avoid quantization artifacts; for example, taking $\beta = 16$ bits is typically more than sufficient. Message censoring can be used to decrease the total number of messages and as a convergence criterion [28], but its overall effect in loopy graphs is difficult to determine [29].

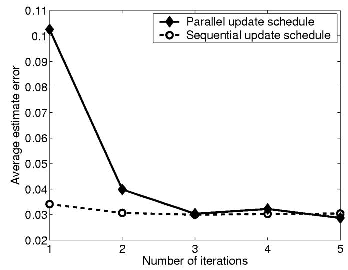
A. Schedule and iterations

The message schedule has a strong influence on BP, affecting the number of iterations until convergence and even potentially the quality of the converged solution [30]. We consider two possible BP message schedules, and analyze performance on the 10-node graph shown in Fig. 3(b). Because we are primarily concerned with the inter-sensor communications required, we enforce a maximum number of messages per sensor, rather than the actual number of iterations.

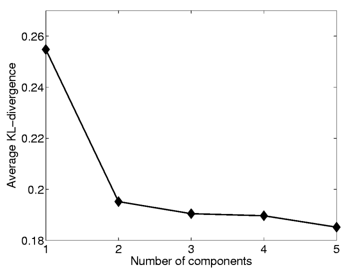
The first BP schedule is a "sequential" schedule, in which each sensor in turn transmits a message to all its neighbors. We determine the order of transmission by beginning with the anchor nodes, and moving outward in sequence based on the shortest observed distance to any anchor. This has similarities to schedules based on spanning trees [31], though (since each sensor is transmitting to *all* neighbors) it is not a tree-structured message ordering. For this schedule, one iteration corresponds to one message from each sensor. Strictly speaking, this ordering is only available given global information (the observed distances of each sensor), but in practice the schedule is robust to small reorderings and thus local or randomized approximations to the sequential schedule could be substituted. Here, however, we will ignore this subtlety.

The second BP schedule we consider is a "parallel" schedule, in which a set of sensors transmit to their neighbors simultaneously. Since initially, large numbers of sensors have no information about their location, we restrict the participating nodes to be those whose belief is well-localized, as determined by some threshold on the entropy of the belief $\hat{p}^n(x_t)$. To provide a fair comparison with the sequential schedule, we limit the number of iterations by allowing each sensor to transmit only a fixed number of messages, terminating when no more sensors are allowed to communicate.

Fig. 8(a) compares the two schedules' performance over 100 Monte Carlo trials, measured by mean error in the location estimates and as a function of the number of message transmissions allowed by each schedule. As can be seen, both schedules produce reasonably similar results, and neither requires more than a few iterations (inter-sensor communications) to converge. Empirically, we find that the sequential schedule performs slightly better on average.



(a)



(b)

Fig. 8. Analyzing the communications cost of NBP. (a) The number of iterations required may depend on the message schedule, but is typically very few (1-3). (b) The transmitted marginal estimates may be compressed by fitting a small Gaussian mixture distribution; a few (1-3) components is usually sufficient.

Faulty communications (nodes' failure to receive some messages) may also be considered in terms of small deletions in the BP message schedule. While the exact effect of these changes is difficult to quantify, it is typically not catastrophic to the algorithm.

B. Message approximation

We may also reduce the communications by approximating each marginal estimate as a small mixture of (diagonal covariance) Gaussians before transmission (instead of sending all particles). Such approximations may be constructed in any number of ways; we use the Kullback-Liebler based approximation of [32] due to its computational efficiency, though more traditional methods such as Expectation-Maximization could also be employed. Note that locally, each node retains its sample-based density estimate (allowing tests for multimodality, etc.) regardless of how coarsely the transmissions are approximated.

In order to observe the effect of this operation on multimodal uncertainty, we performed 100 Monte Carlo trials of NBP with measurement outliers (as in Section VI), but approximated each message by a fixed number of components before transmitting. We apply the sequential schedule described above. Fig. 8(b) shows the resulting marginal estimate errors (measured by KL-divergence from exact message-passing with 1000 particles) as a function of the number of retained components. Single Gaussian (unimodal) approximations to the marginal beliefs resulted in a slight loss in performance, while two-component (potentially bimodal) estimates proved better at capturing the uncertainty. As a benchmark, representing each Gaussian component costs at most 4β bits, so that a two-component mixture at $\beta = 16$ is ≤ 128 bits per message.

IX. Discussion

We proposed a novel approach to sensor localization, applying a graphical model framework and using a nonparametric message-passing algorithm to solve the ensuing inference problem. The methodology has a number of advantages. First, it is easily distributed (exploiting *local* computation and communications between nearby sensors), potentially reducing the amount of communications required. Second, it computes and makes use of estimates of the uncertainty, which may subsequently be used to determine the reliability of each sensor's location estimate. The estimates easily accommodate complex, multi-modal uncertainty. Third, it is straightforward to incorporate additional sources of information, such as a model of the probability of obtaining a distance measurement between sensor pairs. Lastly, in contrast to other methods, it is easily extensible to non-Gaussian noise models, which may be used to model and increase robustness to measurement outliers. In empirical simulations, NBP's performance is comparable to the centralized MAP estimate, while additionally representing the inherent uncertainties.

We have also shown how modifications to the NBP algorithm can result in improved performance. The NBP framework easily accommodates an outlier process model, increasing the method's robustness to a few large errors in distance measurements for little to no computation and communication overhead. Also, carefully chosen proposal distributions can result in improved small-sample performance, reducing the computational costs associated with calibration. Finally, appropriate message schedules require very few message transmissions, and reduced-complexity representations may be applied to lessen the cost of each message transmission with little or no impact on the final solution.

There remain many open directions for continued research. First, other message-passing inference algorithms (e.g. max-product) might improve performance if adapted to high-dimensional non-Gaussian problems. Also, alternative graphical model representations may bear investigating; it may be possible to retain fewer edges, or improve the accuracy of BP by clustering nodes [16]. Given its promising initial performance and many possible avenues of improvement, NBP appears to provide a useful tool for estimating unknown sensor locations in large *ad-hoc* networks.

References

[1] S. Kumar, F. Zhao, and D. Shepherd (Eds.), "Special issue on collaborative information processing," *IEEE Signal Processing Mag.*, vol. 19, no. 2, Mar. 2002.

[2] H. Gharavi and S. Kumar (Eds.), "Special issue on sensor networks and applications," *Proc. IEEE*, vol. 91, no. 8, Aug. 2003.

[3] R. Moses and R. Patterson, "Self-calibration of sensor networks," in *SPIE vol. 4743: Unattended Ground Sensor Technologies and Applications IV*, 2002.

[4] E. B. Sudderth, A. T. Ihler, W. T. Freeman, and A. S. Willsky, "Nonparametric belief propagation," in *CVPR*, 2003.

[5] J. Pearl, *Probabilistic Reasoning in Intelligent Systems*. San Mateo: Morgan Kaufman, 1988.
[6] R. Moses, D. Krishnamurthy, and R. Patterson, "Self-localization for wireless networks," *Eurasip Journal on Applied Signal Processing*, 2003.
[7] K. Whitehouse, "The design of calamari: an ad-hoc localization system for sensor networks," Master's thesis, U. C. Berkeley, 2002.
[8] M. W. Trosset, "The formulation and solution of multidimensional scaling problems," Rice University, Tech. Rep. TR93-55, 1993.
[9] K. Langendoen and N. Reijers, "Distributed localization in wireless sensor networks: a quantitative comparison," *Computer Networks*, vol. 43, no. 4, pp. 499–518, Nov. 2003.
[10] N. Patwari and A. Hero, "Relative location estimation in wireless sensor networks," *IEEE Trans. Signal Processing*, vol. 51, no. 8, pp. 2137–2148, Aug. 2003.
[11] L. Doherty, L. E. Ghaoui, and K. S. J. Pister, "Convex position estimation in wireless sensor networks," in *Infocom*, Apr 2001.
[12] A. Savvides, H. Park, and M. B. Srivastava, "The bits and flops of the n-hop multilateration primitive for node localization problems," in *ACM Workshop on Wireless Sensor Networks and Applications*. ACM, 2003, pp. 112 – 121.
[13] N. Priyantha, H. Balakrishnan, E. Demaine, and S. Teller, "Anchor-free distributed localization in sensor networks," MIT LCS, Tech. Rep. 892, 2003.
[14] M. Fazel, H. Hindi, and S. P. Boyd, "Log-det heuristic for matrix rank minimization with applications to Hankel and Euclidean distance matrices," in *Proceedings, American Control Conference*, 2003.
[15] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 6, no. 6, pp. 721–741, Nov. 1984.
[16] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Constructing free energy approximations and generalized belief propagation algorithms," MERL, Tech. Rep. 2004-040, May 2004.
[17] M. J. Wainwright and M. I. Jordan, "Graphical models, exponential families, and variational inference," UC Berkeley Dept. of Statistics, Tech. Rep. 629, Sept. 2003.
[18] A. T. Ihler, J. W. Fisher III, R. L. Moses, and A. S. Willsky, "Nonparametric belief propagation for self-calibration in sensor networks," in *IPSN*, 2004.
[19] K. Murphy, Y. Weiss, and M. Jordan, "Loopy-belief propagation for approximate inference: An empirical study," in *UAI 15*, July 1999, pp. 467–475.
[20] D. Koller, U. Lerner, and D. Angelov, "A general algorithm for approximate inference and its application to hybrid Bayes nets," in *UAI 15*, 1999, pp. 324–333.
[21] J. M. Coughlan and S. J. Ferreira, "Finding deformable shapes using loopy belief propagation," in *ECCV 7*, May 2002.
[22] P. Felzenszwalb and D. Huttenlocher, "Belief propagation for early vision," in *CVPR*, 2004.
[23] B. Silverman, *Density Estimation for Statistics and Data Analysis*. New York: Chapman and Hall, 1986.
[24] M. Isard, "PAMPAS: Real–valued graphical models for computer vision," in *CVPR*, 2003.
[25] A. T. Ihler, E. B. Sudderth, W. T. Freeman, and A. S. Willsky, "Efficient multiscale sampling from products of Gaussian mixtures," in *NIPS 17*, 2003.
[26] A. Doucet, N. de Freitas, and N. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*. New York: Springer-Verlag, 2001.
[27] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian bayesian tracking," *IEEE Trans. Signal Processing*, vol. 50, no. 2, pp. 174–188, Feb. 2002.
[28] L. Chen, M. Wainwright, M. Cetin, and A. Willsky, "Data association based on optimization in graphical models with application to sensor networks," Submitted to *Mathematical and Computer Modeling*, 2004.
[29] A. T. Ihler, J. W. Fisher III, and A. S. Willsky, "Message errors in belief propagation," Laboratory for Information and Decision Systems, Tech. Rep. TR-2602, 2004.
[30] Y. Mao and A. Banihashemi, "Decoding low-density parity check codes with probabilistic scheduling," *IEEE Commun. Lett.*, vol. 5, no. 10, pp. 414–416, Oct. 2001.
[31] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky, "Tree–based reparameterization analysis of sum–product and its generalizations," *IEEE Trans. Inform. Theory*, vol. 49, no. 5, May 2003.
[32] A. T. Ihler, J. W. Fisher III, and A. S. Willsky, "Communication-constrained inference," Laboratory for Information and Decision Systems, Tech. Rep. TR-2601, 2004.

Alexander T. Ihler Alexander Ihler (S-01) received the B.S. degree from the California Institute of Technology in 1998 and the S.M. from the Massachusetts Institute of Technology in 2000. He is currently pursuing his doctoral degree at MIT in the Stochastic Systems Group. His research interests are in statistical signal processing, machine learning, nonparametric statistics, distributed systems, and sensor networks.

John W. Fisher III John W. Fisher III received a Ph.D. degree in electrical and computer engineering from the University of Florida, Gainesville, Florida in 1997. He is currently a Principal Research Scientist in the Computer Science and Artificial Intelligence Laboratory and affiliated with the Laboratory for Information and Decision Systems, both at MIT. Prior to joining MIT he was affiliated with the University of Florida, as both a faculty member and graduate student since 1987, during which time he conducted research in the areas of ultra-wideband radar for ground penetration and foliage penetration applications, radar signal processing, and automatic target recognition algorithms. His current area of research focus includes information theoretic approaches to signal processing, multi-modal data fusion, machine learning and computer vision.

Randolph L. Moses Randolph L. Moses (S'78–M'85–SM'90) received the B.S., M.S., and Ph.D. degrees in electrical engineering from Virginia Polytechnic Institute and State University in 1979, 1980, and 1984, respectively. During the summer of 1983 he was a SCEEE Summer Faculty Research Fellow at Rome Air Development Center, Rome, NY. From 1984 to 1985 he was with the Eindhoven University of Technology, Eindhoven, The Netherlands, as a NATO Postdoctoral Fellow. Since 1985 he has been with the Department of Electrical Engineering, The Ohio State University, and is currently a Professor there. During 1994–95 he was on sabbatical leave as a visiting researcher at the System and Control Group at Uppsala University in Sweden. His research interests are in digital signal processing, and include parametric time series analysis, radar signal processing, sensor array processing, and communications systems. Dr. Moses served on the Technical Committee on Statistical Signal and Array Processing of the IEEE Signal Processing Society from 1991–94. He is a member of Eta Kappa Nu, Tau Beta Pi, Phi Kappa Phi, and Sigma Xi.

Alan S. Willsky Dr. Alan Willsky (S'70, M'73, SM'82, F'86) joined the M.I.T. faculty in 1973 and is currently the Edwin Sibley Webster Professor of Electrical Engineering. He is a founder, member of the Board of Directors, and Chief Scientific Consultant of Alphatech, Inc. From 1998-2002 he served as a member of the US Air Force Scientific Advisory Board. He has received several awards including the 1975 American Automatic Control Council Donald P. Eckman Award, the 1979 ASCE Alfred Noble Prize and the 1980 IEEE Browder J. Thompson Memorial Award. Dr. Willsky has held visiting positions in England and France and various leadership positions in the IEEE Control Systems Society (which made him a Distinguished Member in 1988). He has delivered numerous keynote addresses and is co-author of the undergraduate text Signals and Systems. His research interests are in the development and application of advanced methods of estimation and statistical signal and image processing. Methods he has developed have been successfully applied in a variety of applications including failure detection, surveillance systems, biomedical signal and image processing, and remote sensing.