Distributed Local Linear Parameter Estimation using Gaussian SPAWN

Mei Leng, Wee Peng Tay, Tony Q.S. Quek, and Hyundong Shin

***Abstract*—We consider the problem of estimating local sensor parameters, where the local parameters and sensor observations are related through linear stochastic models. We study the Gaussian Sum-Product Algorithm over a Wireless Network (gSPAWN) procedure. Compared with the popular diffusion strategies for performing network parameter estimation, whose communication cost at each sensor increases with increasing network density, gSPAWN allows sensors to broadcast a message whose size does not depend on the network size or density, making it more suitable for applications in wireless sensor networks. We show that gSPAWN converges in mean and has mean-square stability under some technical sufficient conditions, and we describe an application of gSPAWN to a network localization problem in non-line-of-sight environments. Numerical results suggest that gSPAWN converges much faster in general than the diffusion method, and has lower communication costs per sensor, with comparable root mean square errors.**

***Index Terms*—Local estimation, distributed estimation, sum-product algorithm, belief propagation, diffusion, wireless sensor network.**

I. Introduction

A wireless sensor network (WSN) consists of many sensors or nodes capable of on-board sensing, computing, and communications. WSNs are used in numerous applications like environmental monitoring, pollution detection, control of industrial machines and home appliances, event detection, and object tracking [1]–[3]. In distributed estimation, sensors cooperate with each other by passing information between neighboring nodes, which removes the necessity of transmitting local data to a central fusion center. Distributed estimation schemes hence have the advantages of being scalable, robust to node failures, and are more energy efficient due to the shorter sensor-to-sensor communication ranges, compared with centralized schemes which require transmission to a fusion center. It also improves local estimation accuracy [4]. For example, by cooperating with each other to perform localization, nodes that have information from an inadequate number of anchors can still successfully self-localize [5].

In this paper, we consider distributed local linear parameter estimation in a WSN. Each sensor cooperates with its neighbors to estimate its own local parameter, which is related to its own observations as well as its neighbors' observations via a linear stochastic model. This is a special case of the more general problem in which sensors cooperate to estimate a *global* vector parameter of interest, as the local parameters can be collected into a single vector parameter. Many distributed estimation algorithms for global parameter estimation have been investigated in the literature, including consensus strategies [6], [7], the incremental least mean square (LMS) algorithm [8], [9], the distributed LMS consensus-based algorithm [10] and diffusion LMS strategies (see the excellent tutorial [11] and the references therein). The consensus strategies are relatively less computationally intensive for each sensor. However, their performance in terms of the convergence rate and mean-square deviation (MSD) are often not as good as the diffusion LMS strategies [12]. The incremental LMS algorithm requires that sensors set up a Hamiltonian path through the network, which is impractical for a large WSN.

To ensure mean and mean-square convergence, the diffusion LMS strategy requires that a pre-defined step size in each diffusion update is smaller than twice the reciprocal of the maximum eigenvalue of the system matrix covariance [11], [13]. Since this eigenvalue is not known a priori, a very small step size is thus typically chosen in the diffusion LMS strategy. This leads to slow convergence rates, resulting in higher communication costs. Furthermore, the diffusion strategy is not specifically designed to perform local sensor parameter estimation in a WSN. For example, when sensors need to estimate their own clock skews and offsets in network synchronization [14], the diffusion LMS strategy either requires that every node estimates the same global parameter, which is a collection of all the sensor local clock skews and offsets, or at least transmits estimates of the clock skews and offsets of all its neighbors (cf. Section III-B for a detailed discussion). In both cases, communication cost for each sensor per iteration increases with the density of the network, and does not make full use of the broadcast nature of wireless communications in a WSN. The distributed LMS consensus-based algorithm of [10] requires the selection of a set of bridge sensors, and the passing of several messages between neighboring sensors, which again does not utilize the broadcast nature of a WSN. It is also more computationally complex than the diffusion algorithm but with better MSD performance.

We therefore ask if there exists a distributed estimation algorithm with similar MSD performance as the diffusion algorithms, and that allows sensors to *broadcast a fixed size*

This work was supported by the Singapore Ministry of Education Academic Research Fund Tier 2 grants MOE2013-T2-2-006 and MOE2014-T2-1-028.

M. Leng and W.P. Tay are with the Nanyang Technological University, Singapore. (e-mail:{lengmei,wptay}@ntu.edu.sg).

T.Q.S. Quek is with the Information Systems Technology and Design Pillar, SUTD, Singapore. (e-mail:qsquek@ieee.org).

H. Shin is with the Department of Electronics and Radio Engineering, Kyung Hee University, Yongin-si, Gyeonggi-do, 446-701, Korea (e-mail: hshin@khu.ac.kr).

message, regardless of the network size or density, to all its neighbors? Our work suggests that the answer is affirmative for a somewhat more restrictive data model than that used in the LMS literature [11], and can be found in the belief propagation (BP) [15], [16] framework, similar to the Sum-Product Algorithm over Wireless Network (SPAWN) proposed by [5]. However, unlike the diffusion algorithms, SPAWN only applies to the case of non-adaptive estimation, which requires the regression matrix to be deterministic, and does not adapt to new observations in each iteration. The reference [5] also does not address the issue of error convergence.

A. Our Contributions

In this paper, we propose an *adaptive* version of SPAWN for distributed local linear parameter estimation, which we call Gaussian SPAWN (gSPAWN) due to the use of Gaussian distributions in the derivation of the messages passed between sensors. We assume that sensors have only local observations with respect to its neighbors, such as the pairwise distance measurement, and the relative clock offset between two nodes. Unlike SPAWN, gSPAWN operates under adaptive model assumptions similar to diffusion algorithms [11], [13], [17], [18], and sensors' observations follow a linear model with a stochastic regression matrix.

It is well known that loopy BP may not converge [19], [20], and its convergence analysis is notoriously difficult. Only a handful of results have established sufficient convergence conditions for Gaussian BP methods [19]–[22]. The gSPAWN procedure is an adaptive cousin of Gaussian BP; one major differing characteristic is that the *same* message is broadcast to all neighboring nodes by each sensor, in contrast to traditional BP where a different message is transmitted to each neighbor. It is of interest to understand the convergence properties of gSPAWN in order for it to have any practical value. Our other main contribution in this paper is the derivation of sufficient conditions for the mean and MSD convergence of gSPAWN. Note that the methods in [19], [20] for analyzing the convergence of Gaussian BP in a loopy graph do not apply to gSPAWN due to the differences in messages transmitted by each node. In fact, because of the broadcast nature of the messages in gSPAWN, our analysis is simpler than that in [19], [20].

We apply gSPAWN to cooperative self-localization in non-line-of-sight (NLOS) multipath environments. We prove that the sensor location estimates given by gSPAWN converges in the mean to the true sensor positions, when all scatterers in the environment are either parallel or orthogonal to each other. To the best of our knowledge, most distributed localization methods [23], [24] consider only line-of-sight (LOS) signals, because NLOS environments introduce non-linearities in the system models. In this application, we adopt a ray tracing model to characterize the relationship between sensor locations, range and angle measurements [25], which leads to a linear relationship between these parameters. We compare the performance of our algorithm with that of a peer-to-peer localization method and the diffusion LMS strategy, with numerical results suggesting that gSPAWN has better average accuracy and convergence rate.

The rest of this paper is organized as follows. In Section II, we define the system model. In Section III, we describe the gSPAWN procedure and compare it to a diffusion algorithm. We provide sufficient conditions for mean convergence and mean-square stability of gSPAWN, and numerical comparison results in Section IV. We then show an application of gSPAWN to network localization in NLOS environments in Section V. Finally, we summarize and conclude in Section VI.

Notations: We use upper-case letters to represent matrices and lower-case letters for vectors and scalars. Bold faced symbols are used to denote random variables. The conjugate, transpose and conjugate transpose of a matrix A are denoted as $\bar{A}$, A^T and A^*, respectively. The minimum and maximum non-zero singular values of A are denoted as $\underline{\delta}(A)$ and $\bar{\delta}(A)$, respectively. The maximum absolute eigenvalue or spectral radius of A is denoted as $r(A)$. When A is Hermitian, we have $r(A) = \bar{\delta}(A)$. We write $A \succeq B$ and $A \succ B$ if $A - B$ is positive semi-definite and positive definite, respectively. If all entries of $B - A$ are non-negative, we write $A \leq B$. The operation $A \otimes B$ is the Kronecker product between A and B. The vector $\mathrm{vec}\{A\}$ is formed by stacking the columns of A together into a column vector. The matrix $\mathrm{diag}(A_1, \ldots, A_n)$ is a block diagonal matrix consisting of the sub-matrices $A_1, \ldots, A_n$ on the main diagonal. The symbol I_m represents a $m \times m$ identity matrix. We use 0_m to denote a $m \times 1$ vector of all zeroes. The operator $\mathbb{E}$ denotes mathematical expectation. The density function of a multivariate Gaussian distribution with mean μ and covariance P is given by $\mathcal{N}(\cdot\ ;\ \mu, \mathrm{P})$.

II. System model

In this section, we describe our system model, and present some assumptions that we make throughout this paper. We consider a network of $n + 1$ sensors $\{0, \ldots, n\}$, where each sensor i wants to estimate a $d \times 1$ parameter s_i. In this paper, we use the terms "sensors" and "nodes" interchangeably. A network consisting of all sensors corresponds to a directed graph $\mathcal{G} = (\mathcal{S}, \mathcal{E})$, where the set of vertices $\mathcal{S} = \{0, \ldots, n\}$, and there is an edge $(i, j) \in \mathcal{E}$ if a broadcast from sensor i can be received at sensor j, in which case we say that sensor i is a neighbor of sensor j. We let the set of neighbors of sensor j be $\mathcal{B}_j$ (which may include sensor j itself or not, depending on the application data model).

Sensors interact with neighbors and estimate their local parameters through in-network processing. In most applications, each sensor can obtain only local measurements w.r.t. its neighbors via communication between each other. For each sensor i, and neighbor $j \in \mathcal{B}_i$, we consider the data model given by

$$\mathbf{d}_{ij}^{(l)} = \mathrm{G}_{ij} s_i - \mathbf{H}_{ij}^{(l)} s_j + \boldsymbol{\omega}_{ij}^{(l)}, \tag{1}$$

where $\mathbf{d}_{ij}^{(l)}$ is the $m \times 1$ vector measurement obtained at sensor i w.r.t. sensor j at time l, G_{ij} is a known $m \times d$ system coefficient matrix, $\mathbf{H}_{ij}^{(l)}$ is an observed $m \times d$ regression matrix at time l, and $\boldsymbol{\omega}_{ij}^{(l)}$ is a measurement noise of dimensions $m \times 1$. We note that this data model is a special case of the widely studied LMS data model (see e.g., [11]), since we can describe (1) by using a global parameter $s = [s_0^T, \ldots, s_n^T]^T$

with appropriate stacking of the measurements $\mathbf{d}_{ij}^{(l)}$. On the other hand, if every sensor i is interested in the same global parameter $s_i = s$, we have from (1) that $\mathbf{d}_{ij}^{(l)} = \mathbf{W}_{ij}^{(l)} s + \boldsymbol{\omega}_{ij}^{(l)}$, where $\mathbf{W}_{ij}^{(l)} = \mathrm{G}_{ij} - \mathbf{H}_{ij}^{(l)}$ is the regression matrix in [11].[1] Although we have assumed for convenience that all quantities have the same dimensions across sensors, our work can be easily generalized to the case where measurements and parameters have different dimensions at each sensor. In the LMS framework, each sensor i seeks to estimate s_i so that the overall network mean square error,

$$\sum_{i=1}^{n} \sum_{j\in\mathcal{B}_i} \mathbb{E}\left\{ \left\| \mathbf{d}_{ij}^{(1)} - \mathrm{G}_{ij} s_i + \mathbf{H}_{ij}^{(1)} s_j \right\|^2 \right\} \tag{2}$$

is minimized. We have the following assumptions similar to [11], [13] on our data model.

Assumption 1

(i) The measurement noises $\boldsymbol{\omega}_{ij}^{(l)}$ are stationary with zero mean, and $\mathbb{E}[\boldsymbol{\omega}_{ij}^{(l)}(\boldsymbol{\omega}_{ij}^{(l)})^*] = \mathrm{C}_{ij}$, where C_{ij} is a positive semi-definite matrix.

(ii) For all i, j, the regression matrix $\mathbf{H}_{ij}^{(l)}$ is stationary over l. The regression matrices $\mathbf{H}_{ij}^{(l)}$ are independent over sensor indices i, j, and time l. They are also independent of measurement noises $\boldsymbol{\omega}_{ij}^{(l)}$ for all i, j, and l.

(iii) For every $i = 1, \ldots, n$, the matrix $\sum_{j\in\mathcal{B}_i} \mathrm{G}_{ij}^* \mathrm{G}_{ij}$ is positive definite.

Assumption 1 imposes stationarity constraints on our data model, without which there is no hope of achieving estimation convergence using any adaptive procedure. Assumption 1(iii) ensures that the matrix $\sum_{j\in\mathcal{B}_i} \mathrm{G}_{ij}^* \mathrm{G}_{ij}$ is invertible. This is similar to the assumption in [11] that the regression matrix in its data model is positive definite, and is required to ensure that each local LMS cost in (2) has a unique minimum. We refer the reader to [11] for more details.

Our goal is to design a distributed algorithm to perform local parameter estimation, which makes full use of the broadcast nature of the wireless medium over which sensors communicate. This is achieved by gSPAWN, which we describe in the next section. We make the following assumptions.

Assumption 2

(i) Sensor 0 is the reference sensor, with an a priori estimate $\hat{s}_0 = s_0 + w_0$ of its parameter s_0, where w_0 is a zero mean Gaussian random variable with covariance matrix $\alpha_0 \mathrm{I}_d$.

(ii) The undirected version of the graph $\mathcal{G}$ is connected, and $\mathcal{G}\backslash\{0\}$ consists of strongly connected components.[2]

(iii) Let $\mathrm{H}_{ij} = \mathbb{E}[\mathbf{H}_{ij}^{(l)}]$ for all $l \geq 1$, $\eta_i^2 = \underline{\delta}(\sum_{j\in\mathcal{B}_i} \mathrm{G}_{ij}^* \mathrm{G}_{ij})$, and $\rho_i^2 = \max_{j\in\mathcal{B}_i} \bar{\delta}(\mathrm{H}_{ij}\mathrm{H}_{ij}^*)$. For all $i \geq 1$, we have $\rho_i^2 \leq \eta_i^2/|\mathcal{B}_i|$ if $|\mathcal{B}_i| \leq 2$, and $\rho_i^2 \leq \eta_i^2/3$ if $|\mathcal{B}_i| \geq 3$.

In some applications such as network localization, it is sufficient to estimate the position of each sensor w.r.t. a single reference node. Therefore, we assume, without loss of generality in Assumption 2(i), that sensor 0 is chosen as the reference node, and its local parameter s_0 is known up to an uncertainty. For example, in relative localization, we have $s_0 = 0_d$ and $\alpha_0 = 0$, while s_i are the coordinates of sensor i w.r.t. sensor 0 to be estimated. In applications where the local parameters s_i do not take reference to any baseline value, we can simply take node 0 to be a virtual node with $\mathrm{G}_{i0} = \mathbf{H}_{i0}^{(1)} = 0$ for all nodes $i = 1, \ldots, n$ and all times l, which then does not change the analysis in this paper.

Without loss of generality, we assume in Assumption 2(ii) that the graph $\mathcal{G}$ with the directions on its edges removed is connected, otherwise estimation on the different components of the graph can be done independently of each other. We will see in Section IV-B the assumption that $\mathcal{G}\backslash\{0\}$ consists of strongly connected components is required to ensure convergence for gSPAWN. For applications in which sensor communication links are symmetric, this assumption holds automatically.

Assumption 2(iii) on the other hand restricts our data model to those in which the system coefficient matrices G_{ij} and expected regression matrices H_{ij} do not differ by too much. This is required, along with the additional Assumption 3 (which will be discussed later in Section IV-B), to ensure that gSPAWN converges.

III. The Gaussian SPAWN

In this section, we derive the gSPAWN procedure based on BP. Since BP is a Bayesian estimation approach, we will *temporarily* view the parameters $s_1, \ldots, s_n$ as random variables with uniform priors in this section. Suppose that we make the observations $\mathbf{O}^{(l)} = (\mathbf{d}_{ij}^{(l)}, \mathbf{H}_{ij}^{(l)})_{i\geq 1, j\in\mathcal{B}_i}$ at a time l. In the Bayesian framework, we are interested to find values $s_1, \ldots, s_n$ that maximize the *a posteriori* probability

$$p\left(\{s_i\}_{i=1}^n \mid \mathbf{O}^{(l)} \right),$$

where the notation $p\left(\mathbf{x} \mid \mathbf{y}\right)$ denotes the conditional probability of $\mathbf{x}$ given $\mathbf{y}$.[3] Since the distributions of $\mathbf{d}_{ij}^{(l)}$ and $\mathbf{d}_{ji}^{(l)}$ depend only on s_i and s_j, we have from (1) that

$$\begin{aligned} p\left(\{s_i\}_{i=1}^n \mid \mathbf{O}^{(l)} \right) &\propto \prod_{(i,j)\in\mathcal{E}} p\left(\mathbf{d}_{ij}^{(l)}, \mathbf{d}_{ji}^{(l)} \,\middle|\, s_i, s_j, \mathbf{H}_{ij}^{(l)}, \mathbf{H}_{ji}^{(l)} \right) \\ &= \prod_{(i,j)\in\mathcal{E}} p\left(\mathbf{d}_{ij}^{(l)} \,\middle|\, s_i, s_j, \mathbf{H}_{ij}^{(l)} \right) p\left(\mathbf{d}_{ji}^{(l)} \,\middle|\, s_j, s_i, \mathbf{H}_{ji}^{(l)} \right). \end{aligned} \tag{3}$$

Let $f_{ij}(s_i, s_j) = p\left(\mathbf{d}_{ij}^{(l)} \,\middle|\, s_i, s_j, \mathbf{H}_{ij}^{(l)} \right)$ and $f_{ji}(s_i, s_j) = p\left(\mathbf{d}_{ji}^{(l)} \,\middle|\, s_j, s_i, \mathbf{H}_{ji}^{(l)} \right)$. We construct a factor graph using $\{s_i\}_{i=0}^n$ as the variable nodes, and for each pair of neighboring sensors i and j with $(i, j) \in \mathcal{E}$, we connect their corresponding variable nodes using the two factor nodes f_{ij} and f_{ji}. See Figure 1 for an example, where each random variable is represented by a circle and each factor node is represented

[1] The data model in [11] assumes that $m = 1$, but can be easily extended to $m > 1$.

[2] The notation $\mathcal{G}\backslash\{0\}$ denotes a graph with node 0 and its incident edges removed. A directed graph is said to be strongly connected if for any pair of nodes u and v in the graph, there exists a directed path going from node u to node v.

[3] Note the abuse of notation here: $\mathbf{x}$ is used to denote both the random variable and its realization.

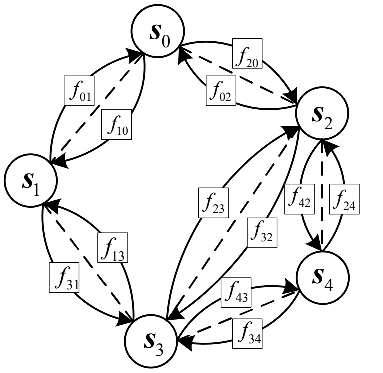


Fig. 1: A factor graph of a network with 5 sensors, where a dashed line indicates that a communication path exists between the corresponding two sensors, and the arrows indicate the direction of message flows.

by a square. We note that our factor graph construction is somewhat untypical compared to those used in traditional loopy BP, where usually a single factor $f_{ij}(s_i, s_j)f_{ji}(s_i, s_j)$ is used between the two variable nodes s_i and s_j. The reason we separate it into two factors is to allow us to design an algorithm that allows sensors to broadcast the same message to all neighboring nodes, as described below.

To apply the sum-product algorithm on a loopy factor graph, we need to define a message passing schedule [16]. We adopt a fully parallel schedule here, a similar scheme has been proposed in [5]. In each iteration, all factor nodes send messages to their neighboring variable nodes in parallel, followed by messages from variable nodes to neighboring factor nodes in parallel. We constrain the updates by allowing messages to flow in only one direction. Specifically, for every i and for all $j \in \mathcal{B}_i$, s_i sends messages only to f_{ji} and not to f_{ij}, and f_{ij} sends messages only to s_i and not to s_j. The two types of messages sent between variable and factor nodes at the kth iteration are the following:

- Each factor node f_{ij} passes a message $h^{(k)}_{f_{ij}\to s_i}(s_i)$ to the variable node s_i representing f_{ij}'s belief of s_i's state. This message is given by

$$h^{(k)}_{f_{ij}\to s_i}(s_i) = \int p\left(\mathbf{d}^{(l)}_{ij} \middle| s_i, s_j, \mathbf{H}^{(l)}_{ij}\right) b^{(k-1)}_j(s_j)\ \mathrm{d}s_j. \quad (4)$$

- Each variable node s_i *broadcasts* a belief to all neighboring factor nodes $\{f_{ji}\}_{j\in\mathcal{B}_i}$, and its belief is given by

$$b^{(k)}_i(s_i) = \prod_{j\in\mathcal{B}_i} h^{(k)}_{f_{ij}\to\mathbf{s}_i}(\mathbf{s}_i). \quad (5)$$

Since the belief $b^{(k)}_i(s_i)$ does not include any messages from f_{ji} in the previous iterations, the same belief message from the variable node s_i is broadcast to all factor nodes $\{f_{ji}\}_{j\in\mathcal{B}_i}$. The communication cost per sensor per iteration hence *does not depend on the size or density of the network.*

In the following, we derive the exact messages passed when $p\left(\mathbf{d}^{(l)}_{ij} \middle| s_i, s_j, \mathbf{H}^{(l)}_{ij}\right)$, for all i, j, are assumed to be Gaussian distributions. In addition, we allow the observations to be updated at each iteration so that the algorithm becomes adaptive in nature. We call this gSPAWN. Although Gaussian distributions are assumed in the derivation of gSPAWN, we show in Section IV that this assumption is not required for mean and MSD convergence of the algorithm.[4]

A. Derivation of Beliefs and Messages for gSPAWN

We first consider the message from the reference node 0 to a neighboring node i. The node 0's belief is held constant over all iterations. From Assumption 2(i), we have $b^{(l)}_0(s_0) = \mathcal{N}\left(s_0\ ;\ \boldsymbol{\mu}^{(l)}_0, \mathrm{P}^{(l)}_0\right)$, where $\boldsymbol{\mu}^{(l)}_0 = \hat{s}_0$ and $\mathrm{P}^{(l)}_0 = \alpha_0\mathrm{I}_d$, for all $l \geq 0$. The message from the factor f_{i0} to sensor i is given by

$$\begin{aligned} h^{(l)}_{f_{i0}\to\mathbf{s}_i}(\mathbf{s}_i) &= \int p\left(\mathbf{d}^{(l)}_{i0} \middle| s_i, s_0, \mathbf{H}^{(l)}_{i0}\right) b^{(l)}_0(s_0)\ \mathrm{d}s_0 \\ &\underset{\sim}{\propto} \mathcal{N}\left(\mathrm{G}_{i0}s_i\ ;\ \boldsymbol{\nu}^{(l)}_{i0}, \Pi^{(l)}_{i0}\right). \end{aligned} \quad (6)$$

where $\boldsymbol{\nu}^{(l)}_{i0} = \mathbf{d}^{(l)}_{i0} + \mathbf{H}^{(l)}_{i0}\boldsymbol{\mu}^{(l)}_0$, $\Pi^{(l)}_{i0} = \mathrm{C}_{i0} + \mathrm{H}_{i0}\mathrm{P}^{(l)}_0\mathrm{H}^*_{i0}$, and the symbol $\underset{\sim}{\propto}$ means approximately proportional to. The approximation in (6) arises because we have replaced $\mathbf{H}^{(l)}_{i0}$ with H_{i0} in $\Pi^{(l)}_{i0}$.

For all nodes i other than the reference node 0, we set the initial belief $b^{(0)}_i(s_i)$ of node i to be a Gaussian distribution with mean $\mu^{(0)}_j = 0_d$ and covariance

$$\mathrm{P}^{(0)}_i = \begin{cases} 4\alpha\mathrm{I}_d, & \text{if } |\mathcal{B}_i| = 1, \\ 3\alpha\mathrm{I}_d, & \text{if } |\mathcal{B}_i| = 2, \\ 5\alpha/3\mathrm{I}_d, & \text{otherwise}, \end{cases} \quad (7)$$

where α is a positive number chosen to be sufficiently large so that $\alpha \geq \alpha_0$, and $\alpha \geq \frac{\max_{i,j} r(\mathrm{C}_{ij})}{\min_i \rho^2_i}$, where $\rho^2_i = \max_{j\in\mathcal{B}_i} \bar{\delta}(\mathrm{H}_{ij}\mathrm{H}^*_{ij})$ (cf. Assumption 2(iii)). A larger initial variance is used for those nodes with less neighbors because such nodes have relatively less information than other nodes in the network. We will see later in the proof of Theorem 1 that the chosen multipliers for the cases where there are not more than two neighbors are the right ones.

Consider the lth iteration of gSPAWN. After $l-1$ iterations, the belief of node j is $b^{(l-1)}_j(s_j) = \mathcal{N}\left(s_j\ ;\ \boldsymbol{\mu}^{(l-1)}_j, \mathrm{P}^{(l-1)}_j\right)$. The message $b^{(l-1)}_j(s_j)$ is a Gaussian distribution, so only the mean $\boldsymbol{\mu}^{(l-1)}_j$ and covariance matrix $\mathrm{P}^{(l-1)}_j$ need to be passed to node i, which then computes

$$\begin{aligned} h^{(l)}_{f_{ij}\to s_i}(s_i) &= \int p\left(\mathbf{d}^{(l)}_{ij} \middle| s_i, s_j, \mathbf{H}^{(l)}_{ij}\right) b^{(l-1)}_j(s_j)\ \mathrm{d}\mathbf{s}_j \\ &\underset{\sim}{\propto} \mathcal{N}\left(\mathrm{G}_{ij}s_i\ ;\ \boldsymbol{\nu}^{(l-1)}_{ij}, \Pi^{(l-1)}_{ij}\right), \end{aligned}$$

where

$$\boldsymbol{\nu}^{(l-1)}_{ij} = \mathbf{d}^{(l)}_{ij} + \mathbf{H}^{(l)}_{ij}\boldsymbol{\mu}^{(l-1)}_j, \quad (8)$$

$$\Pi^{(l-1)}_{ij} = \mathrm{C}_{ij} + \mathrm{H}_{ij}\mathrm{P}^{(l-1)}_j\mathrm{H}^*_{ij}. \quad (9)$$

[4] This is analogous to the philosophy of using the B.L.U.E. for parameter estimation with non-Gaussian system models.

Algorithm 1 gSPAWN: Gaussian Sum-Product Algorithm over a Wireless Network

```
Initialization:
For the reference node 0, set $\boldsymbol{\mu}_0^{(l)} = \hat{s}_0$ and $\mathrm{P}_0^{(l)} = \alpha_0 \mathrm{I}_d$, for all $l \geq 0$.
For $i = 1, \ldots, n$, let $\boldsymbol{\mu}_i^{(0)} = 0_d$ and set $\mathrm{P}_i^{(0)}$ using (7).
for the $l^{\text{th}}$ iteration do
    nodes $i = 1 : n$ in parallel
    broadcast the current belief $b_i^{(l-1)}(s_i)$ to neighboring nodes;
    receive $b_j^{(l-1)}(s_j)$ from neighboring nodes $j$, $j \in \mathcal{B}_i$;
    update its belief as $b_i^{(l)}(\mathbf{s}_i) \sim \mathcal{N}\left(s_i \ ; \ \boldsymbol{\mu}_i^{(l)}, \mathrm{P}_i^{(l)}\right)$ with
        $\mathrm{P}_i^{(l)} = \left( \sum_{j \in \mathcal{B}_i} \mathrm{G}_{ij}^* \left(\Pi_{ij}^{(l-1)}\right)^{-1} \mathrm{G}_{ij} \right)^{-1},$
    and
        $\boldsymbol{\mu}_i^{(l)} = \mathrm{P}_i^{(l)} \sum_{j \in \mathcal{B}_i} \mathrm{G}_{ij}^* \left(\Pi_{ij}^{(l-1)}\right)^{-1} \boldsymbol{\nu}_{ij}^{(l-1)},$
    where $\boldsymbol{\nu}_{ij}^{(l-1)}$ and $\Pi_{ij}^{(l-1)}$ are given in (8) and (9) respectively.
    end parallel
end for
```

We obtain the belief of node i at the lth iteration from (5). Since all the messages in the product in (5) are Gaussian distributions, we have $b_i^{(l)}(s_i) = \mathcal{N}\left(s_i \ ; \ \boldsymbol{\mu}_i^{(l)}, \mathrm{P}_i^{(l)}\right)$ with

$$\mathrm{P}_i^{(l)} = \left(\sum_{j \in \mathcal{B}_i} \mathrm{G}_{ij}^* \left(\Pi_{ij}^{(l-1)}\right)^{-1} \mathrm{G}_{ij} \right)^{-1} \tag{10}$$

and

$$\boldsymbol{\mu}_i^{(l)} = \mathrm{P}_i^{(l)} \sum_{j \in \mathcal{B}_i} \mathrm{G}_{ij}^* \left(\Pi_{ij}^{(l-1)}\right)^{-1} \boldsymbol{\nu}_{ij}^{(l-1)}. \tag{11}$$

We will show in Lemma 3 that $\mathrm{P}_j^{(l)}$ is positive definite for all $j = 1, \ldots, n$ and $l \geq 0$, so (10) is well defined.

The gSPAWN procedure is formally presented in Algorithm 1. Note that factor nodes are virtual, and are introduced only to facilitate the derivation of the algorithm. In practice, the update is performed at each individual sensor directly. At the end of l iterations, sensor i estimates its parameter s_i by maximizing the belief $b_i^{(l)}(s_i)$ with respect to s_i. Since $b_i^{(l)}(s_i)$ is a Gaussian distribution, the local estimator for s_i is given by $\hat{s}_i^{(l)} = \boldsymbol{\mu}_i^{(l)}$, and has a covariance of $\mathrm{P}_i^{(l)}$.

To give an intuitive interpretation of the broadcast messages, assume $d = 1$ and $\mathrm{G}_{ij} = 1$, and $\sigma_{ij}^{(l-1)} = (\Pi_{ij}^{(l-1)})^{-1}$. It can be shown that

$$\boldsymbol{\mu}_i^{(l)} = \sum_{j \in \mathcal{B}_i} \frac{\sigma_{ij}^{(l-1)}}{\sum_{j \in \mathcal{B}_i} \sigma_{ij}^{(l-1)}} \boldsymbol{\nu}_{ij}^{(l-1)}.$$

From (8), $\boldsymbol{\nu}_{ij}^{(l-1)}$ corresponds to the a posterior expectation of sensor j's estimate for sensor i's parameter. Sensor i thus updates its belief mean as a weighted average of expectations from neighboring sensors, where expectations associated with smaller variances have a higher weightage. In the limiting case where $\sigma_{ij}^{(l-1)} = 0$, i.e., sensor j's estimate has extremely poor accuracy, the message from sensor j to sensor i will have no impact on the update of $\boldsymbol{\mu}_i^{(l)}$.

Since each sensor performs local estimation by relying solely on message exchanges with neighbors, in the same way as the diffusion algorithms gSPAWN can be adapted to dynamically changing network topologies in which local sensor parameters change with topology changes, as long as the time-scale at which the network topology changes is larger than that for the message broadcasting iterations [11].

B. Comparison with Adapt-Then-Combine Diffusion

The gSPAWN procedure is reminiscent of the Adapt-Then-Combine (ATC) diffusion algorithm, first proposed by [17]. We present the ATC algorithm in our context of local parameter estimation,[5] and discuss the differences with gSPAWN. As the ATC algorithm assumes that the data model at every sensor is based on a common global parameter, we stack the local sensor parameters into a single global parameter $s = [s_0^T, \ldots, s_n^T]^T$, with the data model at sensor i given by

$$\mathbf{d}_i^{(l)} = \mathbf{W}_i^{(l)} s + (\boldsymbol{\omega}_i^{(l)})^T, \tag{12}$$

where $\boldsymbol{\omega}_i^{(l)} = \left[(\boldsymbol{\omega}_{i1}^{(l)})^T, \ldots, (\boldsymbol{\omega}_{in}^{(l)})^T\right]$, and $\mathbf{W}_i^{(l)}$ is a $(n+1) \times (n+1)$ block matrix consisting of $m \times d$ blocks, with the (j,k)-th block for $0 \leq j, k \leq n$ being

$$\mathbf{W}_i^{(l)}(j,k) = \begin{cases} \mathrm{G}_{ij}, & \text{if } k = i \text{ and } j \in \mathcal{B}_i \backslash \{i\}, \\ \mathrm{G}_{ii} - \mathbf{H}_{ii}^{(l)}, & \text{if } k = i \text{ and } j = i \in \mathcal{B}_i, \\ -\mathbf{H}_{ij}^{(l)}, & \text{if } k = j \text{ and } j \in \mathcal{B}_i \backslash \{i\}, \\ 0 \cdot \mathrm{I}_d, & \text{otherwise.} \end{cases}$$

The ATC update at time l at sensor i consists of the following steps:

$$\boldsymbol{\psi}_i^{(l)} = \mathbf{s}_i^{(l-1)} + \xi_i \cdot (\mathbf{W}_i^{(l)})^* (\mathbf{d}_i^{(l)} - \mathbf{W}_i^{(l)} \mathbf{s}_i^{(l-1)}), \tag{13}$$

$$\mathbf{s}_i^{(l)} = \sum_{j \in \mathcal{B}_i} a_{ij} \boldsymbol{\psi}_j^{(l)}, \tag{14}$$

where $\boldsymbol{\psi}_i^{(l)}$ and $\mathbf{s}_i^{(l)}$ are the intermediate and local estimates of s at sensor i respectively, ξ_i is a step size chosen to be sufficiently small, and (a_{ij}) are non-negative weights with $\sum_j a_{ij} = 1$ for all i. The following remarks summarize the major differences of gSPAWN and ATC.

(i) In the l-th iteration of the ATC update (13)-(14), each sensor i first computes $\boldsymbol{\psi}_i^{(l)}$ and broadcasts it so that its neighbors can perform the combination step (14). The message $\boldsymbol{\psi}_i^{(l)}$ is an intermediate estimate of the global parameter s, and is of size $dn \times 1$. Each sensor i needs to know the total number of nodes n in the network, which determines the size of its broadcast message. Its

[5] We consider only the version of ATC in which there is no exchange of sensor observations with neighbors. We also do not discuss the Combine-Then-Adapt diffusion algorithm as this is known to perform worse than ATC in general [17].

communication cost therefore increases linearly with n. In contrast, in gSPAWN, sensor i broadcasts a fixed size message $(\boldsymbol{\mu}_i^{(l)}, \mathrm{P}_i^{(l)})$, which does not depend on the size of the network.

(ii) Alternatively, instead of broadcasting the message $\boldsymbol{\psi}_i^{(l)}$ in full, sensor i can choose to broadcast only those components of $\boldsymbol{\psi}_i^{(l)}$ that it updates, which correspond to the parameters $\{s_j : j \in \mathcal{B}_i\}$. In this case, the sensor has to add identity headers in its broadcast message so that the receiving sensors know which parameters each component corresponds to. In a static network, this can be done once but in a mobile network, the header information has to be re-broadcast each time the network topology changes. The message itself also has size linear in the number of neighbors of the sensor. Therefore, the communication cost per sensor per iteration increases with the density of the network. We note that the communication cost per sensor per iteration for gSPAWN does not depend on the network node density.

(iii) To compare the computational complexity, we evaluate the number of floating point operations (additions and multiplications) required for both algorithms.[6] It can be shown that the computational complexity for sensor i using gSPAWN is upper bounded by $\mathcal{O}(|\mathcal{B}_i|(m^3+d^3))$. On the other hand, assuming remark (ii) above, the ATC algorithm has complexity $\mathcal{O}(|\mathcal{B}_i|^2 md)$ for sensor i. Therefore, since m and d are fixed constants, ATC is less computationally intensive than gSPAWN for sensors with a small number of neighbors and vice versa. We see that in general, gSPAWN has lower computational complexity per sensor for dense networks.

(iv) In gSPAWN, the matrix weights in (11) need not be non-negative, unlike those in (14). Moreover, the weights in (11) evolve over the iterations, and reflect each sensor's confidence in its belief. The weights for the ATC algorithm we have presented are fixed, although adaptive combination weights for ATC have also been proposed in [11] but no convergence analysis is available.

(v) In (13), to ensure mean and mean-square convergence, the step size ξ_i is chosen so that

$$0 < \xi_i < \frac{2}{r(\mathrm{R}_i)}, \tag{15}$$

where $\mathrm{R}_i = \mathbb{E}[\mathbf{W}_i^* \mathbf{W}_i]$ [13]. Since $r(\mathrm{R}_i)$ is in general not known a priori, the step size ξ_i is chosen to be very small in most applications, which usually leads to a very slow convergence rate. In gSPAWN, the parameter α in (7) plays an analogous role, but it, in effect, does not control the convergence rate since it appears in both $\Pi_{ij}^{(l-1)}$ and $\mathrm{P}_i^{(l)}$ in (11), and can be chosen as large as desired (see Section III-A).

(vi) In Section IV, we show the mean convergence and mean-square stability of gSPAWN under some technical assumptions, which make our model less general than the LMS model used in the ATC algorithm. The advantages of gSPAWN over ATC as described above are only valid under these assumptions. Nevertheless, these assumptions hold for several important sensor network applications including distributed sensor localization, which we discuss in Section V, and distributed sensor clock synchronization [14]. We also remark that gSPAWN is more suitable for *local* parameter estimation, while ATC has lower communication costs in the case of *global* network parameter estimation if the number of parameters is fixed, since gSPAWN needs to broadcast covariance information as well as parameter estimates.

IV. Convergence analysis of gSPAWN

In this section, we show that the covariance matrices $\mathrm{P}_i^{(l)}$ in gSPAWN converge, and establish sufficient conditions for the belief means to converge to the true parameter values. We also show that the MSD converges under additional assumptions. Although gSPAWN follows the framework of BP, its message passing is different from traditional BP. Moreover, we consider an adaptive version of the algorithm in this paper. Therefore, the general convergence analysis in [19], [20] cannot be applied here.

A. Convergence of Covariance Matrices

We first state several elementary results. The first lemma is from [26], and shows that a non-increasing sequence of positive definite matrices (in the order defined by $\succeq$) converges. The proof of Lemma 2 can be found in [27].

Lemma 1 *If the sequence* $\{\mathrm{X}^{(l)}\}_{l=1}^{+\infty}$ *of positive definite matrices is non-increasing, i.e.,* $\mathrm{X}^{(l)} \succeq \mathrm{X}^{(l+1)}$ *for* $l = 1, 2, \cdots$, *then this sequence converges to a positive semi-definite matrix.*

Lemma 2 *If the matrices* X *and* Y *are positive definite, then* $\mathrm{X} \succeq \mathrm{Y}$ *if and only if* $\mathrm{Y}^{-1} \succeq \mathrm{X}^{-1}$.

Lemma 3 *Suppose that Assumption 1*(iii) *holds. For all* $i = 1, \ldots, N$, *and for all* $l \geq 0$, $\mathrm{P}_i^{(l)}$ *is positive definite.*

Proof: We prove the claim by induction on l. From (7), the claim holds for $l = 0$. Suppose the claim holds for $l-1$, we have that $\Pi_{ij}^{(l-1)}$ is positive definite for all $j \in \mathcal{B}_i$. Therefore, the inverse $(\Pi_{ij}^{(l-1)})^{-1}$ is also positive definite, and from (10) and Assumption 1(iii), since $\sum_{j \in \mathcal{B}_i} \mathrm{G}_{ij}^* \mathrm{G}_{ij}$ has full rank, we obtain that $\mathrm{P}_i^{(l)}$ is positive definite. The induction is now complete, and the lemma is proved. ■

The following result shows that the covariance matrices of the beliefs at each variable node in the factor graph converges.

Theorem 1 *Suppose that Assumptions 1 and 2 hold. Then, the covariance matrices* $\mathrm{P}_i^{(l)}$ *at every node* i *in gSPAWN converges, i.e., there exists unique positive semi-definite matrices* $\{\mathrm{P}_i^{(\infty)}\}_{i=1}^n$ *such that for all* $i = 1, \ldots, n$,

$$\lim_{l \to \infty} \mathrm{P}_i^{(l)} = \mathrm{P}_i^{(\infty)}. \tag{16}$$

Proof: All nodes are updated using (10) and (11). Since the covariance matrix C_{ij} is positive definite, we can perform

[6] We say that $f(n) = \mathcal{O}(g(n))$ if $\limsup_{n\to\infty} |f(n)/g(n)| < \infty$.

eigen decomposition on C_{ij}. Let $\mathrm{C}_{ij} = \mathrm{V}_{ij}^* \mathrm{D}_{ij} \mathrm{V}_{ij}$, where V_{ij} is a unitary matrix, and D_{ij} is a diagonal matrix with positive entries. Let $\mathrm{K}_i^{(l)} = \left(\mathrm{P}_i^{(l)}\right)^{-1}$. From (9) and (10), it can be shown that

$$\mathrm{K}_i^{(l)} = \sum_{j \in \mathcal{B}_i} \mathrm{G}_{ij}^* \mathrm{V}_{ij}^* \left(\mathrm{D}_{ij} + \mathrm{V}_{ij} \mathrm{H}_{ij} \mathrm{P}_j^{(l-1)} \mathrm{H}_{ij}^* \mathrm{V}_{ij}^*\right)^{-1} \mathrm{V}_{ij} \mathrm{G}_{ij}. \tag{17}$$

Further performing singular value decomposition on G_{ij} and H_{ij}, we have $\mathrm{G}_{ij} = \mathrm{U}_{ij} \mathrm{A}_{ij} \tilde{\mathrm{U}}_{ij}^*$ and $\mathrm{H}_{ij} = \mathrm{W}_{ij} \mathrm{B}_{ij} \tilde{\mathrm{W}}_{ij}^*$, where U_{ij}, $\tilde{\mathrm{U}}_{ij}$, W_{ij}, and $\tilde{\mathrm{W}}_{ij}$ are unitary matrices, and A_{ij} and B_{ij} are diagonal matrices with corresponding singular values on the diagonal. We have,

$$\begin{aligned}\mathrm{K}_i^{(1)} = \sum_{j \in \mathcal{B}_i} & \tilde{\mathrm{U}}_{ij} \mathrm{A}_{ij}^* \mathrm{U}_{ij}^* \mathrm{V}_{ij}^* \\ & \left(\mathrm{D}_{ij} + \mathrm{V}_{ij} \mathrm{W}_{ij} \mathrm{B}_{ij} \mathrm{P}_j^{(l-1)} \mathrm{B}_{ij}^* \mathrm{W}_{ij}^* \mathrm{V}_{ij}^*\right)^{-1} \mathrm{V}_{ij} \mathrm{U}_{ij} \mathrm{A}_{ij} \tilde{\mathrm{U}}_{ij}^*.\end{aligned} \tag{18}$$

We show by induction on l that $\mathrm{K}_i^{(l)}$ is non-decreasing for all $i = 0, \ldots, N$. Since the reference node 0 does not update its belief, this clearly holds for $i = 0$. In the following, we prove the claim for $i = 1, \ldots, n$. Since $\alpha \geq \max_{i,j} r(\mathrm{D}_{ij})/\rho_i^2$, it follows that $\mathrm{D}_{ij} \preceq \alpha \rho_i^2 \mathrm{I}_m$ for all i, j. From Assumption 2(iii), we also have $\bar{\delta}(\mathrm{H}_{ij} \mathrm{H}_{ij}^*) \leq \rho_i^2$, and hence $\mathrm{B}_{ij} \mathrm{B}_{ij}^* \preceq \rho_i^2 \mathrm{I}_m$ for all i, j. Using these two properties, we now consider different cases for the size of $\mathcal{B}_i$.

Case $|\mathcal{B}_i|= 1$: From Assumption 2(iii), we have $\rho_i^2 \leq \eta_i^2$. We consider the cases where the reference node 0 is a neighbor of node i or not separately. Suppose that node 0 is the neighbor of node i. Then, since $\mathrm{P}_0^{(0)} = \alpha_0 \mathrm{I}_2$, we have from (18) that

$$\mathrm{K}_i^{(1)} \succeq \frac{\eta_i^2}{(\alpha + \alpha_0)\rho_i^2} \mathrm{I}_d \ \succeq \frac{\eta_i^2}{2\alpha\rho_i^2} \mathrm{I}_d \ \succeq \frac{1}{4\alpha} \mathrm{I}_d = \mathrm{K}_i^{(0)},$$

where the second inequality follows from $\alpha_0 \leq \alpha$. If node 0 is not a neighbor of node i, suppose that $\mathcal{B}_i = \{j\}$, where $j \neq 0$. From Assumption 2(ii), we have $j \neq i$ and $|\mathcal{B}_j|\geq 2$, yielding

$$\mathrm{K}_i^{(1)} \succeq \frac{\eta_i^2}{\alpha(\rho_i^2 + 3\rho_i^2)} \mathrm{I}_d \ \succeq \frac{1}{4\alpha} \mathrm{I}_d = \mathrm{K}_i^{(0)}.$$

Case $|\mathcal{B}_i|= 2$: From Assumption 2(iii), we have $\rho_i^2 \leq \eta_i^2/2$. Denote $\mathcal{B}_i = \{j_1, j_2\}$, we have $|\mathcal{B}_{j_1}|= 1$ and $|\mathcal{B}_{j_2}|= 2$ in the worst case, and it follows from (18) that

$$\begin{aligned}\mathrm{K}_i^{(1)} &\succeq \frac{1}{\alpha} \frac{\mathrm{G}_{ij_1}^* \mathrm{G}_{ij_1}}{\rho_i^2 + 4\rho_i^2} + \frac{1}{\alpha} \frac{\mathrm{G}_{ij_2}^* \mathrm{G}_{ij_2}}{\rho_i^2 + 3\rho_i^2} \\ &= \frac{1}{\alpha} \frac{4\mathrm{G}_{ij_1}^* \mathrm{G}_{ij_1} + 5\mathrm{G}_{ij_2}^* \mathrm{G}_{ij_2}}{20\rho_i^2} \\ &\succeq \frac{1}{\alpha} \frac{\eta_i^2}{5\rho_i^2} \mathrm{I}_d \ \succeq \frac{1}{3\alpha} \mathrm{I}_d = \mathrm{K}_i^{(0)}.\end{aligned}$$

Case $|\mathcal{B}_i| \geq 3$: From (18), we have

$$\begin{aligned}\mathrm{K}_i^{(1)} \succeq \sum_{j \in \mathcal{B}_i} & \tilde{\mathrm{U}}_{ij} \mathrm{A}_{ij}^* \mathrm{U}_{ij}^* \mathrm{V}_{ij}^* \\ & (\mathrm{D}_{ij} + 4\alpha \mathrm{V}_{ij} \mathrm{W}_{ij} \mathrm{B}_{ij} \mathrm{B}_{ij}^* \mathrm{W}_{ij}^* \mathrm{V}_{ij}^*)^{-1} \mathrm{V}_{ij} \mathrm{U}_{ij} \mathrm{A}_{ij} \tilde{\mathrm{U}}_{ij}^*.\end{aligned}$$

From Assumption 2(iii), we have $\rho_i^2 \leq \eta_i^2/3$, and we obtain

$$\mathrm{K}_i^{(1)} \succeq \frac{\sum_{j \in \mathcal{B}_i} \tilde{\mathrm{U}}_{ij} \mathrm{A}_{ij}^* \mathrm{A}_{ij} \tilde{\mathrm{U}}_{ij}^*}{5\alpha\rho_i^2} \ \succeq \frac{1}{\alpha} \frac{\eta_i^2}{5\rho_i^2} \mathrm{I}_d \ \succeq \frac{3}{5\alpha} \mathrm{I}_d = \mathrm{K}_i^{(0)}.$$

Therefore, we have shown that $\mathrm{K}_i^{(1)} \succeq \mathrm{K}_i^{(0)}$ for all nodes $i \geq 1$.

Suppose that $\mathrm{K}_j^{(l)} \succeq \mathrm{K}_j^{(l-1)}$ for all j. From Lemmas 2 and 3, we have $\mathrm{P}_j^{(l-1)} \succeq \mathrm{P}_j^{(l)}$, and it follows that $\mathrm{V}_{ij} \mathrm{H}_{ij} \mathrm{P}_j^{(l-1)} \mathrm{H}_{ij}^* \mathrm{V}_{ij}^* \succeq \mathrm{V}_{ij} \mathrm{H}_{ij} \mathrm{P}_j^{(l)} \mathrm{H}_{ij}^* \mathrm{V}_{ij}^*$. Applying Lemma 2, we have

$$\begin{aligned}&\left(\mathrm{D}_{ij} + \mathrm{V}_{ij} \mathrm{H}_{ij} \mathrm{P}_j^{(l)} \mathrm{H}_{ij}^* \mathrm{V}_{ij}^*\right)^{-1} \\ &\quad \succeq \left(\mathrm{D}_{ij} + \mathrm{V}_{ij} \mathrm{H}_{ij} \mathrm{P}_j^{(l-1)} \mathrm{H}_{ij}^* \mathrm{V}_{ij}^*\right)^{-1},\end{aligned}$$

which together with (17) yields

$$\mathrm{K}_i^{(l+1)} \succeq \mathrm{K}_i^{(l)}.$$

This completes the induction, and the claim that $\mathrm{K}_i^{(l)}$ is non-decreasing in l for each i is now proven. This implies that $\mathrm{P}_i^{(l)}$ is non-increasing in l. The theorem now follows from Lemma 1, and the proof is complete. ■

Theorem 1 shows that the covariance matrices at every node converges to the positive semi-definite matrices $\{\mathrm{P}_i^{(\infty)}\}_{i=1}^n$, which is the fixed point solution to the set of equations

$$\left(\mathrm{P}_i^{(\infty)}\right)^{-1} = \sum_{j \in \mathcal{B}_i} \mathrm{G}_{ij}^* \left(\mathrm{C}_{ij} + \mathrm{H}_{ij} \mathrm{P}_j^{(\infty)} \mathrm{H}_{ij}^*\right)^{-1} \mathrm{G}_{ij}, \tag{19}$$

for all $i \geq 1$, with $\mathrm{P}_0^{(\infty)} = \alpha_0 \mathrm{I}_d$.

B. Convergence of Belief Means

In this subsection, we provide a technical assumption that together with Assumptions 1 and 2 ensure that gSPAWN converges in mean. We then provide sufficient conditions that satisfy the given technical assumption. Using the covariance matrices $\{\mathrm{P}_i^{(\infty)}\}_{i=0}^n$, which can be derived using (19), we let $\mathrm{Q}^{(\infty)}$ be a $dn \times dn$ matrix consisting of $n \times n$ blocks of $d \times d$ submatrices, where for $1 \leq i, j \leq n$, the (i, j)-th block is

$$\mathrm{Q}^{(\infty)}(i,j) = \begin{cases} \mathrm{P}_i^{(\infty)} \mathrm{G}_{ij}^* \left(\mathrm{C}_{ij} + \mathrm{H}_{ij} \mathrm{P}_j^{(\infty)} \mathrm{H}_{ij}^*\right)^{-1} \mathrm{H}_{ij}, & \text{if } j \in \mathcal{B}_i, \\ 0 \cdot \mathrm{I}_d, & \text{otherwise.} \end{cases} \tag{20}$$

Assumption 3 *The spectral radius $r(\mathrm{Q}^{(\infty)})$ is less than 1.*

Assumption 3, together with Assumptions 1 and 2, are sufficient for the mean convergence (Theorem 2) of gSPAWN. If the observations $(\mathbf{d}_{ij}^{(l)}, \mathbf{H}_{ij}^{(l)})_{i,j,l}$ are scalars, then it can be shown that Assumption 3 holds under quite general conditions, see [14]. However, for vector observations, $\mathrm{Q}^{(\infty)}$ in general does not have spectral radius less than 1 even when $\mathrm{H}_{ij} = \mathrm{G}_{ij}$ for all i, j (a numerical example is provided in Appendix A). To verify Assumption 3 requires a priori knowledge of the sensor network architecture, which may not be practical. To overcome this difficulty, we present local sufficient conditions in the following proposition that ensure Assumption 3 is satisfied.

Proposition 1 *Suppose that Assumptions 1 and 2 hold. In addition, there exists a unitary matrix* U *so that for all* $i, j = 0, \ldots, n$, *there exists a unitary matrix* V_{ij} *such that*

(i) $\mathrm{G}_{ij} = \mathrm{V}_{ij}\mathrm{A}_{ij}\mathrm{U}^*$, *where* A_{ij} *is a diagonal matrix with non-negative real numbers on the diagonal;*

(ii) $\mathrm{H}_{ij} = \mathrm{V}_{ij}\mathrm{B}_{ij}\mathrm{U}^*$, *where* B_{ij} *is a diagonal matrix with non-negative real numbers on the diagonal, with* $\mathrm{B}_{ij} \leq \mathrm{A}_{ij}$*; and*

(iii) $\mathrm{C}_{ij} = \mathrm{V}_{ij}\mathrm{D}_{ij}\mathrm{V}_{ij}^*$, *where* D_{ij} *is a diagonal matrix with positive real numbers on the diagonal.*

Then, Assumption 3 holds.

Proof: We first show, by induction on l, that for all $i \geq 0$ and $l \geq 0$, $\mathrm{P}_i^{(l)}$ has the form $\mathrm{U}\Lambda_i^{(l)}\mathrm{U}^*$, where $\Lambda_i^{(l)}$ is a non-negative diagonal matrix. This is trivially true for all $i \geq 0$ when $l = 0$. It is also true for $i = 0$ and for all $l \geq 1$ since the reference node 0 does not update its belief. Suppose that the claim holds for $l - 1$. Then we have for $i \geq 1$,

$$\begin{aligned}\left(\mathrm{P}_i^{(l)}\right)^{-1} &= \sum_{j\in\mathcal{B}_i} \mathrm{G}_{ij}^* \left(\mathrm{C}_{ij} + \mathrm{H}_{ij}\mathrm{P}_j^{(l-1)}\mathrm{H}_{ij}^*\right)^{-1} \mathrm{G}_{ij} \\ &= \mathrm{U} \underbrace{\sum_{j\in\mathcal{B}_i} \mathrm{A}_{ij} \left(\mathrm{D}_{ij} + \mathrm{B}_{ij}\Lambda_j^{(l-1)}\mathrm{B}_{ij}\right)^{-1} \mathrm{A}_{ij}}_{\triangleq \Lambda_i^{(l)}} \mathrm{U}^*, \quad (21)\end{aligned}$$

so the claim holds since $\mathrm{A}_{ij}\left(\mathrm{D}_{ij} + \mathrm{B}_{ij}\Lambda_j^{(l-1)}\mathrm{B}_{ij}\right)^{-1}\mathrm{A}_{ij}$ is a diagonal matrix for all i, j. This implies that for all $i \geq 0$, $\mathrm{P}_i^{(\infty)} = \mathrm{U}\Lambda_i^{(\infty)}\mathrm{U}^*$, for some non-negative diagonal matrix $\Lambda_i^{(\infty)}$ as $l \to \infty$. We hence have from (20) for $j \in \mathcal{B}_i$,

$$\begin{aligned}\mathrm{Q}^{(\infty)}(i,j) &= \mathrm{P}_i^{(\infty)}\mathrm{G}_{ij}^* \left(\mathrm{C}_{ij} + \mathrm{H}_{ij}\mathrm{P}_j^{(\infty)}\mathrm{H}_{ij}^*\right)^{-1} \mathrm{H}_{ij} \\ &= \mathrm{U}\left(\Lambda_i^{(\infty)}\right)^{-1} \mathrm{A}_{ij} \left(\mathrm{D}_{ij} + \mathrm{B}_{ij}\Lambda_j^{(\infty)}\mathrm{B}_{ij}\right)^{-1} \mathrm{B}_{ij}\mathrm{U}^*. \quad (22)\end{aligned}$$

Let $\tilde{\mathrm{Q}}$ be a a $dn \times dn$ matrix consisting of $n \times n$ blocks of $d \times d$ submatrices, where the (i,j)-th block is $(\Lambda_i^{(\infty)})^{-1}\mathrm{A}_{ij}(\mathrm{D}_{ij} + \mathrm{B}_{ij}\Lambda_j^{(\infty)}\mathrm{B}_{ij})^{-1}\mathrm{B}_{ij}$ for $1 \leq i, j \leq n$ and $j \in \mathcal{B}_i$. Then from (22), $r(\mathrm{Q}^{(\infty)}) = r(\tilde{\mathrm{Q}})$. We have

$$\sum_{\substack{j\in\mathcal{B}_i \\ j\neq 0}} \tilde{\mathrm{Q}}(i,j) \leq \left(\Lambda_i^{(\infty)}\right)^{-1} \sum_{\substack{j\in\mathcal{B}_i \\ j\neq 0}} \mathrm{A}_{ij} \left(\mathrm{D}_{ij} + \mathrm{B}_{ij}\Lambda_j^{(\infty)}\mathrm{B}_{ij}\right)^{-1} \mathrm{A}_{ij}. \quad (23)$$

From (21), the right hand side (R.H.S.) of (23) is equal to I_d if $0 \notin \mathcal{B}_i$, and $\prec \mathrm{I}_d$ if $0 \in \mathcal{B}_i$. This implies that $\tilde{\mathrm{Q}}$ is a non-negative sub-stochastic matrix. From Assumption 2, $\tilde{\mathrm{Q}}$ can be expressed, by relabeling the nodes if necessary, as a block diagonal matrix where each diagonal block is irreducible (when the reference node 0 is removed from the network, the network breaks into irreducible components represented by each diagonal block). Applying the Perron-Frobenius theorem [28] on each block, we obtain the proposition. ■

Let $\tilde{\boldsymbol{\mu}}_i^{(l)} = \boldsymbol{\mu}_i^{(l)} - s_i$. Substituting (8) and (9) into (11), we obtain

$$\begin{aligned}\tilde{\boldsymbol{\mu}}_i^{(l)} &= \mathrm{P}_i^{(l)} \sum_{j\in\mathcal{B}_i} \mathrm{G}_{ij}^* \left(\Pi_{ij}^{(l-1)}\right)^{-1} \mathbf{H}_{ij}^{(l)} \tilde{\boldsymbol{\mu}}_j^{(l-1)} \\ &\quad + \mathrm{P}_i^{(l)} \sum_{j\in\mathcal{B}_i} \mathrm{G}_{ij}^* \left(\Pi_{ij}^{(l-1)}\right)^{-1} \boldsymbol{\omega}_{ij}^{(l)}.\end{aligned}$$

Denote $\tilde{\boldsymbol{\mu}}^{(l)} = [\tilde{\boldsymbol{\mu}}_1^{(l)}, \ldots, \tilde{\boldsymbol{\mu}}_n^{(l)}]^T$, and $\boldsymbol{\Omega}^{(l)} = [\boldsymbol{\omega}_1^{(l)}, \cdots, \boldsymbol{\omega}_n^{(l)}]^T$, where $\boldsymbol{\omega}_i^{(l)} = [(\boldsymbol{\omega}_{i1}^{(l)})^T, \ldots, (\boldsymbol{\omega}_{in}^{(l)})^T]$, we have

$$\tilde{\boldsymbol{\mu}}^{(l)} = \mathbf{Q}^{(l)}\tilde{\boldsymbol{\mu}}^{(l-1)} + \mathrm{R}^{(l)}\boldsymbol{\Omega}^{(l)}, \quad (24)$$

where $\mathbf{Q}^{(l)}$ is a $dn \times dn$ matrix consisting of $n \times n$ blocks of $d \times d$ submatrices, with the (i,j)-th block for $1 \leq i, j \leq n$ being

$$\mathbf{Q}^{(l)}(i,j) = \begin{cases} \mathrm{P}_i^{(l)}\mathrm{G}_{ij}^* \left(\Pi_{ij}^{(l-1)}\right)^{-1} \mathbf{H}_{ij}^{(l)}, & \text{if } j \in \mathcal{B}_i, \\ 0 \cdot \mathrm{I}_d, & \text{otherwise.} \end{cases} \quad (25)$$

The matrix $\mathrm{R}^{(l)} = \mathrm{E}(\mathrm{I}_n \otimes \tilde{\mathrm{Q}}^{(l)})$, where $\tilde{\mathrm{Q}}^{(l)}$ is the same as $\mathbf{Q}^{(l)}$ except that it is missing the $\mathbf{H}_{ij}^{(l)}$ terms on the R.H.S. of (25), and E is a selection matrix defined as

$$\mathrm{E} = \begin{bmatrix} e_1^T \otimes \mathrm{I}_d \\ \vdots \\ e_{(i-1)n+i}^T \otimes \mathrm{I}_d \\ \vdots \\ e_{n^2}^T \otimes \mathrm{I}_d \end{bmatrix},$$

with e_i being a $n^2 \times 1$ vector where there is a 1 at the i-th component, and all other entries are zero. From Theorem 1, the following lemma follows easily.

Lemma 4 *Let* $\mathrm{Q}^{(l)} = \mathbb{E}\left\{\mathbf{Q}^{(l)}\right\}$. *Suppose that Assumptions 1-3 hold. Then,* $\lim_{l\to\infty} \mathrm{Q}^{(l)} = \mathrm{Q}^{(\infty)}$ *in* (20)*, and for all* l *sufficiently large, we have* $r(\mathrm{Q}^{(l)}) < 1$. *Furthermore, there exists a finite constant* c *such that for all* l, $\left\|\mathrm{R}^{(l)}\right\| \leq c$.

Theorem 2 *Suppose that Assumptions 1-3 hold. Then, for all* $i \geq 0$, *we have*

$$\lim_{l\to\infty} \mathbb{E}\left\{\tilde{\boldsymbol{\mu}}_i^{(l)}\right\} = 0_d.$$

Proof: Let $\tilde{u}^{(l)} = \mathbb{E}\left\{\tilde{\boldsymbol{\mu}}^{(l)}\right\}$, then taking expectation on both sides of (24), we have $\tilde{u}^{(l)} = \mathrm{Q}^{(l)}\tilde{u}^{(l-1)}$, where $\mathrm{Q}^{(l)} = \mathbb{E}\left\{\mathbf{Q}^{(l)}\right\}$, and we have used the independence of $\mathbf{Q}^{(l)}$ and $\tilde{\boldsymbol{\mu}}^{(l-1)}$. The theorem now follows from Lemma 4. ■

C. MSD Convergence

Since (24) is a linear recursive equation, its mean-square stability analysis follows from similar methods used in Section 6.5 of [11]. We note however that our analysis is complicated by the fact that $\mathbf{Q}^{(l)}$ in (24) does not have the same form as that in [11], which allows decomposition of the recursion into components that then makes it possible to replace the random matrix $\mathbf{Q}^{(l)}$ with $\mathrm{Q}^{(l)} = \mathbb{E}\left\{\mathbf{Q}^{(l)}\right\}$. In addition, we do not have the luxury of a step size parameter that can be chosen sufficiently small so that higher order terms can be ignored. Therefore, we require stronger assumptions on the distributions of $\mathbf{H}_{ij}^{(l)}$ in order to ensure MSD convergence for gSPAWN.

The MSD at iteration l is defined to be $\mathbb{E}\left\{\left\|\tilde{\boldsymbol{\mu}}^{(l)}\right\|^2\right\}$. Consider a positive definite matrix W. Let $w = \mathrm{vec}(W)$ be the vector formed by stacking the columns of W together. For any vector a, let $\|a\|_w^2 = \|a\|_W^2 = a^* W a$. Following the same derivation as in [11], it can be shown that

$$\mathbb{E}\left\{\left\|\tilde{\boldsymbol{\mu}}^{(l)}\right\|_w^2\right\} = \mathbb{E}\left\{\left\|\tilde{\boldsymbol{\mu}}^{(l-1)}\right\|_{\mathcal{F}^{(l)}w}^2\right\} + \mathcal{R}^{(l)}w, \tag{26}$$

where

$$\mathcal{F}^{(l)} = \mathbb{E}\left\{\left(\mathbf{Q}^{(l)}\right)^T \otimes \left(\mathbf{Q}^{(l)}\right)^*\right\}, \tag{27}$$

and

$$\mathcal{R}^{(l)} = \left(\mathrm{vec}\left(\mathrm{R}^{(l)}\mathrm{C}\left(\mathrm{R}^{(l)}\right)^T\right)\right)^T,$$

with $\mathrm{C} = \mathrm{diag}(\mathrm{C}_{11}, \ldots, \mathrm{C}_{n1}, \ldots, \mathrm{C}_{1n}, \ldots, \mathrm{C}_{nn})$. The MSD at any iteration l can be found by setting $W = \mathrm{I}_{dn}$. We say that gSPAWN is mean-square stable if the MSD converges as $l \to \infty$.[7]

By iterating the recursive relationship (26), we have

$$\begin{aligned}\mathbb{E}\left\{\left\|\tilde{\boldsymbol{\mu}}^{(l)}\right\|^2\right\} &= \mathbb{E}\left\{\left\|\tilde{\boldsymbol{\mu}}^{(0)}\right\|_{\prod_{i=1}^{l}\mathcal{F}^{(i)}\mathrm{vec}\{\mathrm{I}_{dn}\}}^2\right\} \\ &\quad + \sum_{i=1}^{l}\mathcal{R}^{(i)}\prod_{j=i+1}^{l}\mathcal{F}^{(j)}\mathrm{vec}\left\{\mathrm{I}_{dn}\right\}.\end{aligned}$$

Therefore, to show mean square stability, it suffices to show that the matrices $\mathcal{F}^{(l)}$ are stable for all l sufficiently large because of Lemma 4. Without any restrictions on the distributions of $\mathbf{H}_{ij}^{(l)}$, $\mathcal{F}^{(l)}$ may not be stable. In the following, we present some sufficient conditions for mean-square stability, which apply in various practical applications.

Proposition 2 *Suppose that Assumptions 1-3 hold, and $\mathbf{H}_{ij}^{(l)} = \mathrm{H}_{ij}$ is non-random for all i, j and l. Then, the MSD in gSPAWN converges.*

Proof: From (27), the eigenvalues of $\mathcal{F}^{(l)}$ are of the form $\phi\xi^*$, where ϕ and ξ are eigenvalues of $\mathbf{Q}^{(l)}$, which is assumed here to be non-random. The proposition now follows from Assumption 3. ■

Proposition 3 *Suppose that Assumptions 1 and 2 hold. In addition, there exists a unitary matrix* U *so that for all* $i, j = 0, \ldots, n$*, there exists a unitary matrix* V_{ij} *such that*

(i) $\mathrm{G}_{ij} = \mathrm{V}_{ij}\mathrm{A}_{ij}\mathrm{U}^*$*, where* A_{ij} *is a diagonal matrix with non-negative real numbers on the diagonal;*

(ii) $\mathbf{H}_{ij}^{(l)} = \mathrm{V}_{ij}\mathbf{B}_{ij}^{(l)}\mathrm{U}^*$ *almost surely for all* l*, where* $\mathbf{B}_{ij}^{(l)}$ *is a diagonal matrix. Furthermore,* $\mathrm{B}_{ij} = \mathbb{E}\left\{\mathbf{B}_{ij}^{(l)}\right\}$ *and* $\tilde{\mathrm{B}}_{ij} = \mathbb{E}\left\{(\mathbf{B}_{ij}^{(l)})^2\right\}$ *are diagonal matrices with non-negative real numbers on the diagonal, with* $\tilde{\mathrm{B}}_{ij} \leq \mathrm{A}_{ij}^2$*; and*

(iii) $\mathrm{C}_{ij} = \mathrm{V}_{ij}\mathrm{D}_{ij}\mathrm{V}_{ij}^*$*, where* D_{ij} *is a diagonal matrix with positive real numbers on the diagonal.*

Then, the MSD in gSPAWN converges.

Proof: The same proof in Proposition 1 shows that for all $i \geq 0$ and $l \geq 0$, $\mathrm{P}_i^{(l)}$ has the form $\mathrm{U}\Lambda_i^{(l)}\mathrm{U}^*$, where $\Lambda_i^{(l)}$ is a non-negative diagonal matrix. For fixed $l \geq 0$ and $a \geq 0$, we have

$$\prod_{k=a}^{l}\mathcal{F}^{(k)}\mathrm{vec}\left\{\mathrm{I}_{dn}\right\} = \mathrm{vec}\left\{\mathrm{F}_l^{(a)}\right\},$$

where $\mathrm{F}_l^{(k)}$ is defined recursively as $\mathrm{F}_l^{(k)} = \mathbb{E}\left\{\left(\mathbf{Q}^{(k)}\right)^*\mathrm{F}_l^{(k+1)}\mathbf{Q}^{(k)}\right\}$, with $\mathrm{F}_l^{(l+1)} = \mathrm{I}_{dn}$. Let $\bar{\mathrm{U}} = \mathrm{diag}(\mathrm{U}, \ldots, \mathrm{U})$ and $\mathrm{J}^{(k)}$ be matrices consisting of $n \times n$ blocks of size $d \times d$, where the (i, j)-th block of $\mathrm{J}^{(k)}$ is given by

$$\mathrm{J}^{(k)}(i,j) = \begin{cases}\Lambda_i^{(k)}\mathrm{A}_{ij}\left(\mathrm{D}_{ij} + \mathrm{B}_{ij}\Lambda_j^{(k-1)}\mathrm{B}_{ij}\right)^{-1}\mathrm{A}_{ij}, & \text{if } j \in \mathcal{B}_i, \\ 0 \cdot \mathrm{I}_d, & \text{otherwise.}\end{cases}$$

Viewing $\mathrm{F}_l^{(k)}$ as a $n \times n$ block matrix, we show by backward induction on k that each block $\mathrm{U}^*\mathrm{F}_l^{(k)}(i,j)\mathrm{U}$ is a non-negative diagonal matrix and $\bar{\mathrm{U}}^*\mathrm{F}_l^{(k)}\bar{\mathrm{U}} \leq (\mathrm{J}^{(k)})^*\bar{\mathrm{U}}^*\mathrm{F}_l^{(k+1)}\bar{\mathrm{U}}\mathrm{J}^{(k)}$. It then follows from the Perron-Frobenius theorem [28] that since $\mathrm{J}^{(k)}$ has spectral radius less than one, $\mathrm{F}_l^{(a)} \to 0 \cdot \mathrm{I}_{dn \times dn}$ exponentially fast as $l \to \infty$ so that the proposition holds.

Suppose the claim holds for $k + 1$. Let $\tilde{\Pi}_{ij}^{(k)} = \left(\mathrm{D}_{ij} + \mathrm{B}_{ij}\Lambda_j^{(k)}\mathrm{B}_{ij}\right)^{-1}$, which is a non-negative diagonal matrix for all k. It can be shown algebraically that

$$\begin{aligned}&\mathrm{U}^*\mathrm{F}_l^{(k)}(i,j)\mathrm{U} = \\ &\sum_{q,q'=1}^{n}\mathbb{E}\left\{\mathbf{B}_{iq}^{(k)}\tilde{\Pi}_{iq}^{(k-1)}\mathrm{A}_{iq}\mathrm{U}^*\mathrm{F}_l^{(k+1)}(q,q')\mathrm{U}\mathrm{A}_{jq'}\tilde{\Pi}_{jq'}^{(k-1)}\mathbf{B}_{jq'}^{(k)}\right\} \\ &\leq \sum_{q=1}^{n}\sum_{q'=1}^{n}\mathrm{A}_{iq}^{(k)}\tilde{\Pi}_{iq}^{(k-1)}\mathrm{A}_{iq}\mathrm{U}^*\mathrm{F}_l^{(k+1)}(q,q')\mathrm{U}\mathrm{A}_{jq'}\tilde{\Pi}_{jq'}^{(k-1)}\mathrm{A}_{jq'}^{(k)} \\ &= \left((\mathrm{J}^{(k)})^*\bar{\mathrm{U}}^*\mathrm{F}_l^{(k+1)}\bar{\mathrm{U}}\mathrm{J}^{(k)}\right)(i,j),\end{aligned}$$

where the inequality holds because all the matrices are non-negative and diagonal, $\mathbb{E}\left\{\mathbf{B}_{iq}^{(k)}\mathbf{B}_{jq'}^{(k)}\right\} = \mathbb{E}\left\{\mathbf{B}_{iq}^{(k)}\right\}\mathbb{E}\left\{\mathbf{B}_{jq'}^{(k)}\right\} \leq \mathrm{A}_{iq}\mathrm{A}_{jq'}$ if $i \neq j$ or $q \neq q'$, and $\mathbb{E}\left\{(\mathbf{B}_{iq}^{(k)})^2\right\} \leq \mathrm{A}_{iq}^2$ by assumption. The claim trivially holds for $k = l$, and the induction is complete. The proposition is now proved. ■

D. Numerical Examples and Discussions

In this section, we provide numerical results to compare the performance of gSPAWN and the ATC algorithm. For the network graph $\mathcal{G}$, we use a k-regular graph with 20 nodes

[7] Similar to [11], the mean-square error and excess mean-square error can be defined, but we do not discuss those here.

generated using results from [29]. Each sensor i has a 2×1 local parameter s_i to be estimated. The matrices G_{ij} and $\mathbf{H}_{ij}^{(l)}$ are 1×2 vectors. For each $i, j = 0, \ldots, 7$, we let G_{ij} to be either $[15, 0]$ or $[0, 15]$ randomly so that Assumption 1(iii) is satisfied. The expectation H_{ij} is chosen to be either $[10, 0]$ or $[0, 10]$ randomly, and each $\mathbf{H}_{ij}^{(l)}$ is generated from a Gaussian distribution with covariance $2\mathrm{I}_2$. The parameter α in gSPAWN is chosen to be 10. The relative degree-variance in [13] is used as the combination weight in the ATC algorithm. The step-size in the ATC algorithm for sensor i (cf. equation (15)) is chosen to be $\eta / r(\mathrm{R}_i)$, where $0 < \eta < 2$.

The root mean squared error (RMSE) for each sensor i is given by $\sqrt{\mathbb{E}\left\{\left\|s_i - \hat{\mathbf{s}}_i^{(l)}\right\|_2^2\right\}}$, where $\hat{\mathbf{s}}_i^{(l)}$ is the local estimate of s_i in both the gSPAWN and ATC algorithm. The iteration l is chosen to be sufficiently large so that the RMSE is within 10^{-5} of the steady state value, at which point we say that the algorithm has converged. We use the average RMSE over all sensors as the performance measure. The Cramer-Rao lower bound (CRLB) is used as a benchmark. The likelihood function of the sensor observations at iteration l can be shown using (1) to be proportional to

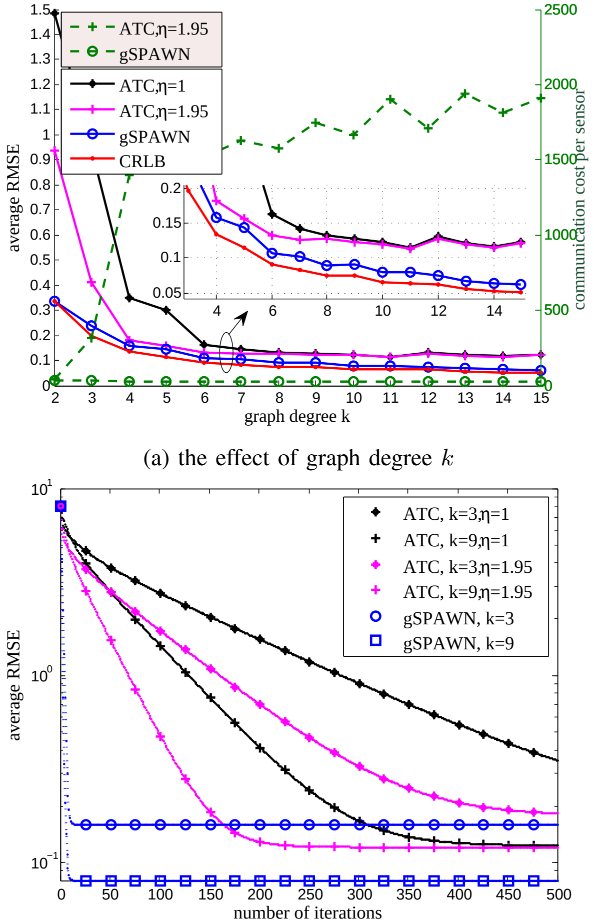
$$\exp\left\{-\frac{1}{2}(\mathbf{E}^{(l)}s - \mathbf{D}^{(l)})^* \mathrm{C}^{-1}(\mathbf{E}^{(l)}s - \mathbf{D}^{(l)})\right\}, \qquad (28)$$

where $s = [s_1^T, \cdots, s_n^T]^T$, $\mathbf{E}^{(l)}$ is a selection matrix consisting of all system matrices $\{\mathrm{G}_{ij}, \mathbf{H}_{ij}^{(l)}\}$ at iteration l, and $\mathbf{D}^{(l)}$ and C are the matrices obtained by stacking all measurements $\mathbf{d}_{ij}^{(l)}$ and noise covariances C_{ij} together in the order dictated by $\mathbf{E}^{(l)}$. It is well known that Fisher information matrix for (28) is given by $\mathbf{E}^{(l)*}\mathrm{C}^{-1}\mathbf{E}^{(l)}$ [30]. The average CRLB is obtained by averaging over l.

In Figure 2(a), we see that gSPAWN and ATC have comparable average RMSE when the graph degree is greater than 3. We also observe that the total number of messages transmitted per sensor for gSPAWN is much smaller than that for ATC. This is because gSPAWN not only converges faster, but also passes fewer messages in each iteration. To be specific, the number of messages per sensor per iteration for gSPAWN is 2, and that for ATC is the graph degree k.

In Figure 2(b), we observe that gSPAWN converges within 30 iterations, while the ATC algorithm convergence rate depends not only on the graph degree but also on the value of η, with a lower graph degree and a smaller value of η leading to a slower convergence rate. This can also be seen by comparing the average spectral radius[8]. In our simulations, the spectral radius for ATC with $\eta = 1.95$ is between 0.85 and 0.98, while that of gSPAWN is between 0.54 and 0.72.

We also evaluate the effect of observation noise variance and network size in Figure 3. We set $n = 20$ and $k = 5$ in Figure 3(a), where the noise standard deviation ranges from 0.01 to 10. It is clear that the performance deteriorates as the noise variance increases for both algorithms. In Figure 3(b), we set the noise standard deviation to 3, each node to have fixed degree $k = 5$, and increase the network size from 6 to 20. The communication cost per sensor for ATC increases because it takes longer to converge in a large network, but the communication cost per sensor for gSPAWN remains relatively constant.



(a) the effect of graph degree k

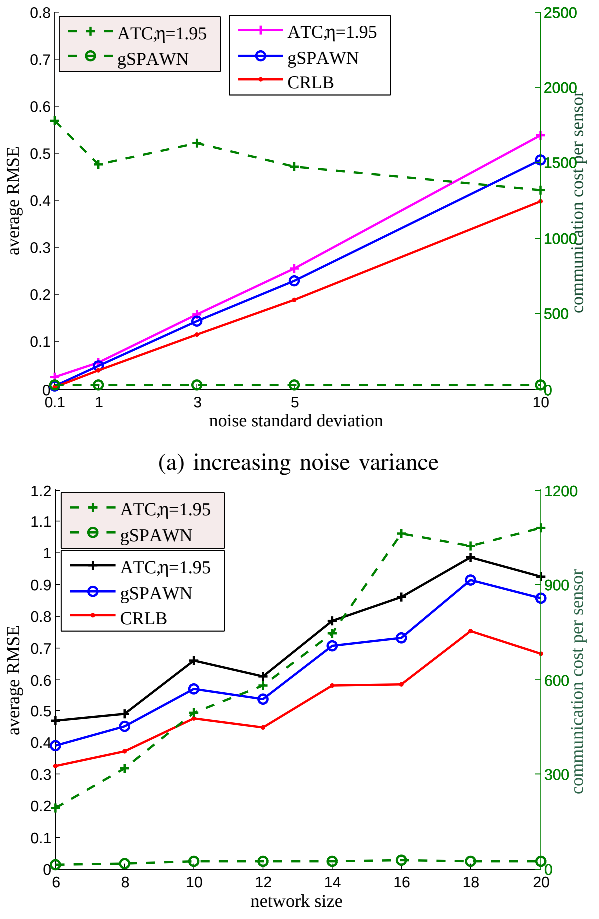
(b) convergence performance

Fig. 2: RMSE performance comparison between gSPAWN and ATC for regular graphs.

V. Application to Cooperative Localization in NLOS Environments

In this section, we apply gSPAWN to the problem of distributed sensor localization in a NLOS environment. Suppose that there are $n+1$ sensors $0, \ldots, n$ in a network, where the position of sensor i is $s_i = [x_i, y_i]^T$ with x_i and y_i being the x and y coordinates respectively. Without loss of generality, we assume that the position of the reference sensor 0 is known up to some estimation accuracy. The objective of each sensor i is to perform self-localization relative to sensor 0 by measuring time-of-arrival (TOA) or received signal strength, and angle-of-arrival (AOA) of signals broadcast by each sensor. In most urban and indoor environments, sensors may not have direct LOS to each other. Signals transmitted by one sensor may be propagated to another sensor through multiple paths. The multipath propagation between sensors can be characterized by a ray tracing model [25]. In this application, we restrict ourselves to paths that have at most one reflection. An example

[8] The spectral radius for gSPAWN is calculated for $\mathbf{Q}^{(l)}$ in (25), and that for ATC is calculated for the recursion matrix in equation (246) of [13].



(a) increasing noise variance

(b) increasing network size

Fig. 3: The effect of network size and noise variance.

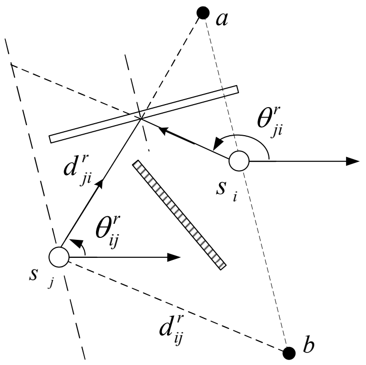


Fig. 4: An example for single-bounce scattering path between sensor i and sensor j.

of a single-bounce scattering path is shown in Figure 4, where a signal transmitted from sensor j to sensor i is reflected at a nearby scatterer. Suppose that there are k_{ij} LOS or NLOS paths between sensor i and sensor j, and that individual paths can be resolved from the received signal at each node. This is the case if, for example, the signal bandwidth is sufficiently large.

Consider the kth path between sensor i and j as shown in Figure 4. Let $\mathbf{d}_{ij}^k$ be the distance measured by sensor i using the TOA or received signal strength along this path from sensor j, and θ_{ij}^k be the corresponding AOA of the signal. We assume that a particular direction has been fixed to be the horizontal direction, and all angles are measured w.r.t. this direction. In addition, sensors do not have prior knowledge of the positions of the scatterers. It can be shown using geometrical arguments [25], [31] that

$$\mathbf{d}_{ij}^k = g(\theta_{ij}^k, \theta_{ij}^k)^T \left(s_i - s_j\right), \tag{29}$$

where

$$g(\theta, \phi) = \begin{bmatrix} \frac{\sin\theta + \sin\phi}{\sin(\theta-\phi)} \\ -\frac{\cos\theta + \cos\phi}{\sin(\theta-\phi)} \end{bmatrix}. \tag{30}$$

When the signal path between sensor i and j is a LOS path, we have $|\theta_{ij}^k - \theta_{ij}^k| = \pi$ and $d_{ij}^k = \|s_i - s_j\|$, therefore we define $g\left(\theta_{ij}^k, \theta_{ij}^k\right) = [\cos(\theta_{ij}^k), \sin(\theta_{ij}^k)]^T$ in (29).

We stack the measurements from all k_{ij} paths into vectors $\tilde{\mathbf{d}}_{ij} = [\mathbf{d}_{ij}^1, \ldots, \mathbf{d}_{ij}^{k_{ij}}]^T$ and $\mathrm{T}_{ij} = [g(\theta_{ij}^1, \theta_{ij}^1), \ldots, g(\theta_{ij}^{k_{ij}}, \theta_{ij}^{k_{ij}})]^T$. We assume that $k_{ij} > 1$ and T_{ij} has linearly independent columns, otherwise some of the paths are redundant as they do not provide additional information to perform localization. Let $\mathbf{d}_{ij} = \mathrm{T}_{ij}^\dagger \tilde{\mathbf{d}}_{ij}$, where $\mathrm{T}_{ij}^\dagger$ is the Moore-Penrose pseudo-inverse of T_{ij}. We adopt the data model

$$\mathbf{d}_{ij} = s_i - s_j + \boldsymbol{\omega}_{ij}, \tag{31}$$

where we assume that the AOA of each arriving signal can be measured sufficiently accurately [32] so that the distance and AOA measurement noise can be included together in $\boldsymbol{\omega}_{ij}$, which is a 2×1 measurement noise vector with covariance $\mathrm{C}_{ij} = \sigma_{ij}^2 (\mathrm{T}_{ij}^\dagger)(\mathrm{T}_{ij}^\dagger)^T = \sigma_{ij}^2 (\mathrm{T}_{ij}^T \mathrm{T}_{ij})^{-1}$. Note that C_{ij} is a function of the AOA of the signal from sensor j to sensor i.

This corresponds to the general model in (1), where sensor observations are the same over all iterations l, and $\mathrm{G}_{ij} = \mathbf{H}_{ij}^{(l)} = \mathrm{I}_2$ for all l. The gSPAWN procedure in Section III can now be used to estimate s_i in a distributed manner. In this non-adaptive application, we assume that only one measurement is observed at each sensor, but the same algorithm can also be applied if new observations are available at each iteration of gSPAWN as shown in Section III. The local updates (10) and (11) at sensor i in the lth iteration are given by

$$\mathrm{P}_i^{(l)} = \left(\sum_{j \in \mathcal{B}_i} \left(\mathrm{C}_{ij} + \mathrm{P}_j^{(l-1)} \right)^{-1} \right)^{-1}, \tag{32}$$

$$\boldsymbol{\mu}_i^{(l)} = \mathrm{P}_i^{(l)} \sum_{j \in \mathcal{B}_i} \left(\mathrm{C}_{ij} + \mathrm{P}_j^{(l-1)} \right)^{-1} \left(\boldsymbol{\mu}_j^{(l-1)} + \mathbf{d}_{ij} \right), \tag{33}$$

where $\boldsymbol{\mu}_i^{(l)}$ is the sensor's local estimate of its position after l iterations.

A. Convergence of Location Estimates

Suppose that all scatterers in the environment are either parallel or orthogonal to each other (e.g., in indoor environments, the strongest scatterers are the walls, floor and ceiling). The convergence of gSPAWN in distributed localization of sensors follows from results in Section IV.

Corollary 1 *In gSPAWN for the NLOS localization problem in a strongly connected sensor network using data model* (31)*, we have*

(i) the covariance matrix in (32) *converges;*

(ii) the belief means in (33) *converges in mean to the true sensor locations when all scatterers are either parallel or orthogonal to each other, and*

(iii) the location MSDs converge when all scatterers are either parallel or orthogonal to each other.

Proof: From (31), it can be checked that Assumptions 1 and 2 are satisfied with $\mathrm{G}_{ij} = \mathrm{H}_{ij} = \mathbf{H}_{ij}^{(l)} = \mathrm{I}_2$. Claim (i) now follows from Theorem 1. We next show that Assumption 3 holds when all scatterers are either parallel or orthogonal to each other. In this case, there exists a unitary matrix U such that after applying U to the system model, all the scatterers are either horizontal or vertical. Then for all $i = 1, \ldots, n$, $j \in \mathcal{B}_i$, and each path k, we have $(\theta_{ij}^k + \theta_{ij}^k - 2\rho) \pmod{2\pi} = \pi$, where $\rho = 0$ or $\pi/2$ depending on whether the path is reflected by a horizontal or vertical scatterer, respectively. From (30), we obtain

$$\mathrm{U}^*\mathrm{C}_{ij}\mathrm{U} = \sigma_{ij}^2 \begin{bmatrix} \sum_{k\in\mathcal{K}_h} [\sec(\theta_{ij}^k)]^2 & 0 \\ 0 & \sum_{k\in\mathcal{K}_v} [\csc(\theta_{ij}^k)]^2 \end{bmatrix}^{-1},$$

where $\mathcal{K}_h$ and $\mathcal{K}_v$ are the index sets of paths from horizontal and vertical scatterers, respectively. Assumption 3 now follows from Proposition 1, and claims (ii) and (iii) then follow from Theorem 2 and Proposition 2 respectively. The corollary is now proved. ■

B. Simulation Results and Discussions

In this section, we present simulation results for the distributed localization problem. We consider a network with 25 nodes scattered in an enclosed area of 100m × 100m. The communication links between nodes are shown in Figure 5. Scatterers are placed throughout the area. Simulations are conducted with different values of the measurement noise parameter $\sigma_{ij} = \sigma$ for all i, j. Each simulation run consists of 1000 independent trials. We use the average RMSE over all sensors as the performance measure, and we also use CRLB as a benchmark, which are calculated in a way similar to that in Section IV-D.

We compare the performance of gSPAWN with that of a peer-to-peer localization method and the ATC method. In the peer-to-peer localization approach, sensors that have direct measurements w.r.t. the reference node are localized first, and then other sensors are localized by treating previously localized neighbors as virtual anchor nodes. In the ATC strategy, the step-size parameter for s_i is set to be $1/r(\mathrm{R}_i)$, where $r(\mathrm{R}_i)$ is as defined in (15). The relative degree-variance is used as combination weights.

In Figure 6(a), we show the average RMSE with respect to different values of σ. It can be seen that gSPAWN outperforms both ATC and the peer-to-peer localization method, with average RMSE close to that of the ATC strategy. The convergence performance is shown in Figure 6(b), where $\sigma = 1$. gSPAWN converges within 100 iterations, while the ATC converges only after 350 iterations. In this numerical example, the peer-to-peer method uses one broadcast per sensor per iteration, gSPAWN uses two broadcasts, i.e., one for the estimated mean and the other for the covariance information, per sensor per iteration, while the ATC strategy uses an average of 5.68 broadcasts per sensor per iteration.

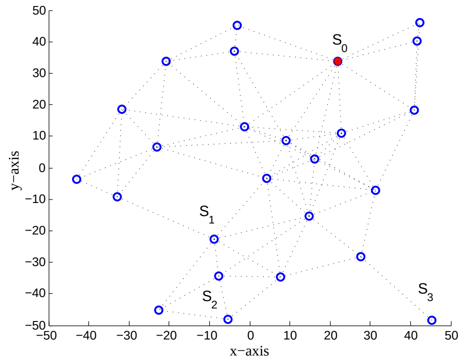
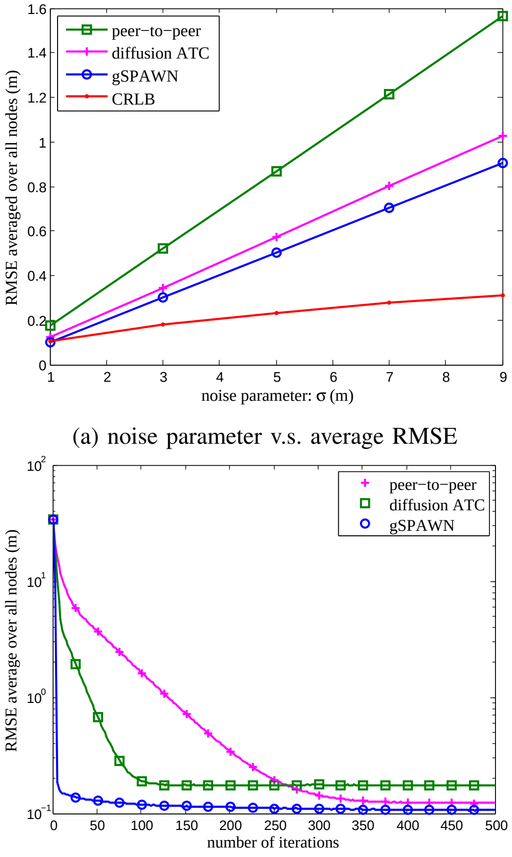


Fig. 5: A signal graph corresponding to a network of 25 nodes. The reference node is s_0.



(a) noise parameter v.s. average RMSE

(b) convergence performance with $\sigma = 1$

Fig. 6: Performance comparison betweenn gSPAWN, diffusion ATC, and peer-to-peer algorithm.

VI. Conclusions

We have studied a distributed local linear parameter estimation algorithm called gSPAWN, where at every iteration, each node in the network broadcasts a fixed size message to all neighbors, with the message size remaining constant regardless of the network size and density. It is therefore well suited for implementation in WSNs. We show that gSPAWN converges in mean, and has mean-square stability under some technical sufficient conditions. We also show how to apply gSPAWN to a network localization problem in NLOS environments. Our simulation results suggest that gSPAWN has better average RMSE performance than the diffusion ATC and peer-to-peer localizaton methods.

Appendix A
Numerical Example of $r(\mathrm{Q}^{(\infty)}) > 1$.

In this appendix, we provide a numerical example that shows that unlike the case where measurements are scalar with $m = d = 1$, the matrix $\mathrm{Q}^{(\infty)}$ given by (20) may have spectral radius greater than one if additional conditions are not imposed. We consider a linear graph with $n = 2$ nodes and a reference node 0. There is an edge between node 0 and node 1, and another edge between node 1 and node 2. Let $\mathrm{G}_{ij} = \mathrm{H}_{ij} = \mathrm{I}_2$ for all i, j in (20), $\alpha_0 = 0$, $\mathrm{C}_{01} = 100 \cdot \mathrm{I}_2$, and for $i, j = 1, 2$, $\mathrm{C}_{ij} = \mathrm{A}_{ij}^{-1} - \mathrm{P}_j^{(\infty)}$, where

$$\mathrm{A}_{11} = \begin{bmatrix} 0.3586 & -0.1580 \\ -0.1580 & 0.1135 \end{bmatrix}, \mathrm{A}_{12} = \begin{bmatrix} 0.0709 & -0.0890 \\ -0.0890 & 0.1118 \end{bmatrix},$$
$$\mathrm{A}_{21} = \begin{bmatrix} 0.3670 & -0.2359 \\ -0.2359 & 0.1522 \end{bmatrix}, \mathrm{A}_{22} = \begin{bmatrix} 0.3067 & -0.0163 \\ -0.0163 & 0.0357 \end{bmatrix},$$

and

$$\left(\mathrm{P}_1^{(\infty)}\right)^{-1} = \begin{bmatrix} 0.4395 & -0.2470 \\ -0.2470 & 0.2353 \end{bmatrix},$$
$$\left(\mathrm{P}_2^{(\infty)}\right)^{-1} = \begin{bmatrix} 0.6737 & -0.2522 \\ -0.2522 & 0.1879 \end{bmatrix}.$$

It can be checked that $\mathrm{P}_1^{(\infty)}$ and $\mathrm{P}_2^{(\infty)}$ satisfy (19), and

$$\mathrm{Q}^{(\infty)} = \begin{bmatrix} 1.0695 & -0.2156 & -0.1250 & 0.1574 \\ 0.4512 & 0.2560 & -0.5094 & 0.6403 \\ 0.1503 & -0.0943 & 0.8497 & 0.0943 \\ -1.0537 & 0.6834 & 1.0537 & 0.3166 \end{bmatrix},$$

which has a spectral radius of 1.017.

References

[1] N. Bulusu and S. Jha, *Wireless Sensor Networks: A Systems Perspective*. Artech House, 2005.

[2] W. P. Tay, J. N. Tsitsiklis, and M. Z. Win, "Bayesian detection in bounded height tree networks," *IEEE Trans. Signal Process.*, vol. 57, no. 10, pp. 4042–4051, Oct. 2009.

[3] ——, "On the impact of node failures and unreliable communications in dense sensor networks," *IEEE Trans. Signal Process.*, vol. 56, no. 6, pp. 2535–2546, Jun. 2008.

[4] M. R. Gholami, S. Gezici, and E. G. Strom, "Improved position estimation using hybrid tw-toa and tdoa in cooperative networks," *IEEE Trans. Signal Process.*, vol. 60, no. 7, pp. 3770–3785, 2012.

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

[6] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *Information Theory, IEEE Transactions on*, vol. 52, no. 6, pp. 2508–2530, 2006.

[7] T. C. Aysal, M. E. Yildiz, A. D. Sarwate, and A. Scaglione, "Broadcast gossip algorithms for consensus," *IEEE Trans. Signal Process.*, vol. 57, no. 7, pp. 2748–2761, 2009.

[8] C. G. Lopes and A. H. Sayed, "Incremental adaptive strategies over distributed networks," *IEEE Trans. Signal Processing*, vol. 55, no. 8, pp. 4064–4077, August 2007.

[9] F. Cattivelli and A. H. Sayed, "Analysis of spatial and incremental LMS processing for distributed estimation," *IEEE Trans. on Signal Process.*, vol. 59, no. 4, pp. 1465–1480, April 2011.

[10] I. Schizas, G. Mateos, and G. Giannakis, "Distributed LMS for consensus-based in-network adaptive processing," *Signal Processing, IEEE Transactions on*, vol. 57, no. 6, pp. 2365–2382, 2009.

[11] A. H. Sayed, "Diffusion adaptation over networks," in *E-Reference Signal Processing*, R. Chellapa and S. Theodoridis, Eds. Elsevier, 2013. [Online]. Available: http://arxiv.org/abs/1205.4220

[12] S.-Y. Tu and A. Sayed, "Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks," *Signal Processing, IEEE Transactions on*, vol. 60, no. 12, pp. 6217–6234, 2012.

[13] A. H. Sayed, S.-Y. Tu, J. Chen, X. Zhao, and Z. J. Towfic, "Diffusion strategies for adaptation and learning over networks: an examination of distributed strategies and network behavior," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 155–171, 2013.

[14] M. Leng and Y.-C. Wu, "Distributed clock synchronization for wireless sensor networks using belief propagation," *IEEE Trans. Signal Process.*, vol. 59, no. 11, pp. 5404–5414, 2011.

[15] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, 2001.

[16] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[17] F. S. Cattivelli and A. H. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1035–1048, 2010.

[18] N. Takahashi, I. Yamada, and A. H. Sayed, "Diffusion least-mean squares with adaptive combiners: Formulation and performance analysis," *IEEE Trans. Signal Process.*, vol. 58, no. 9, pp. 4795–4810, 2010.

[19] Y. Weiss and W. T. Freeman, "Correctness of belief propagation in Gaussian graphical models of arbitrary topology," *Neural Comput.*, vol. 13, no. 10, pp. 2173–2200, Oct. 2001.

[20] J. K. Johnson, D. M. Malioutov, and A. S. Willsky, "Walk-sum interpretation and analysis of Gaussian belief propagation," *Advances in Neural Information Processing System*, vol. 18, 2006.

[21] T. Heskes, "On the uniqueness of loopy belief propagation fixed points," *Neural Comput.*, vol. 16, no. 11, pp. 2379–2413, Nov. 2004.

[22] C. Moallemi and B. Van Roy, "Convergence of min-sum message passing for quadratic optimization," *IEEE Trans. Inf. Theory*, vol. 55, no. 5, pp. 2413–2423, May 2009.

[23] A. T. Ihler, I. Fisher, J. W., R. L. Moses, and A. S. Willsky, "Nonparametric belief propagation for self-localization of sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 4, pp. 809–819, 2005.

[24] S. Zhu and Z. Ding, "Distributed cooperative localization of wireless sensor networks with convex hull constraint," *IEEE Trans. Wireless Commun.*, vol. 10, no. 7, pp. 2150–2161, 2011.

[25] C. K. Seow and S. Y. Tan, "Non-line-of-sight localization in multipath environments," *IEEE Trans. Mobile Comput.*, vol. 7, no. 5, pp. 647–660, 2008.

[26] A. M. Sarhan, N. M. EI-Shazly, and E. M. Shehata, "On the existence of extremal positive definite solutions of the nonlinear matrix equation $\mathbf{X}^r + \sum_{i=1}^m \mathbf{A}_i^* \mathbf{X}^{\delta_i} \mathbf{A}_i = \mathbf{I}$," *Mathematical and Computer Modelling*, vol. 51, no. 9-10, 2010.

[27] G. A. Milliken, "A theorem on the difference of the generalized inverse of two nonnegative matrices," *Commun. Statist.-Theor. Meth.*, vol. A6(1), 1977.

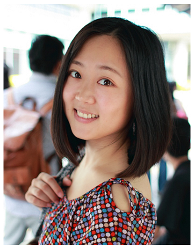
[28] E. Seneta, *Nonnegative matrices and markov chains*, 2nd ed., ser. Springer Series in Statistics. Springer, 1981.

[29] M. Meringer, "Fast generation of regular graphs and construction of cages," *Journal of Graph Theory*, vol. 30, pp. 137–146, 1999.

[30] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Englewood Cliffs, NJ, 1993.

[31] M. Leng, W.-P. Tay, and T. Quek, "Cooperative and distributed localization for wireless sensor networks in multipath environments," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, 2012, pp. 3125–3128.

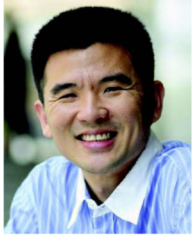
[32] J. B. Andersen and K. I. Pedersen, “Angle-of-arrival statistics for low resolution antennas,” *IEEE Trans. Antennas Propag.*, vol. 50, no. 3, pp. 391–395, 2002.



Mei Leng received the B.Eng. degree from University of Electronic Science and Technology of China (UESTC) in 2005 and Ph.D. degree from The University of Hong Kong in 2011. She is currently a research scientist at Temasek Laboratories@NTU. Her research interests include statistical signal processing, optimization, machine learning as well as Bayesian analysis, with applications to wireless sensor networks and wireless communication systems. She has been working on navigation and tracking projects, and she is particularly interested in the algorithm design and the implementation using software-defined radio.

Wee Peng Tay (S’06 M’08 SM’14) received the B.S. degree in Electrical Engineering and Mathematics, and the M.S. degree in Electrical Engineering from Stanford University, Stanford, CA, USA, in 2002. He received the Ph.D. degree in Electrical Engineering and Computer Science from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 2008. He is currently an Assistant Professor in the School of Electrical and Electronic Engineering at Nanyang Technological University, Singapore. His research interests include distributed detection and estimation, distributed signal processing, sensor networks, social networks, information theory, and applied probability.

Dr. Tay received the Singapore Technologies Scholarship in 1998, the Stanford University President’s Award in 1999, and the Frederick Emmons Terman Engineering Scholastic Award in 2002. He is the coauthor of the best student paper award at the 46th Asilomar conference on Signals, Systems, and Computers. He is currently serving as the chair of DSNIG in IEEE MMTC, a member of MSLP TC, and has served as a technical program committee member for various international conferences.



Tony Q.S. Quek (S’98-M’08-SM’12) received the B.E. and M.E. degrees in Electrical and Electronics Engineering from Tokyo Institute of Technology, Tokyo, Japan, respectively. At Massachusetts Institute of Technology, he earned the Ph.D. in Electrical Engineering and Computer Science. Currently, he is an Assistant Professor with the Information Systems Technology and Design Pillar at Singapore University of Technology and Design (SUTD). He is also a Scientist with the Institute for Infocomm Research. His main research interests are the application of mathematical, optimization, and statistical theories to communication, networking, signal processing, and resource allocation problems. Specific current research topics include sensor networks, heterogeneous networks, green communications, smart grid, wireless security, compressed sensing, big data processing, and cognitive radio.

Dr. Quek has been actively involved in organizing and chairing sessions, and has served as a member of the Technical Program Committee as well as symposium chairs in a number of international conferences. He is serving as the PHY & Fundamentals Track for IEEE WCNC in 2015, the Communication Theory Symposium for IEEE ICC in 2015, and the PHY & Fundamentals Track for IEEE EuCNC in 2015. He is currently an Editor for the IEEE Transactions on Communications, the IEEE Wireless Communications Letters, and an Executive Editorial Committee Member for the IEEE Transactions on Wireless Communications. He was Guest Editor for the IEEE Signal Processing Magazine (Special Issue on Signal Processing for the 5G Revolution) in 2014, and the IEEE Wireless Communications Magazine (Special Issue on Heterogeneous Cloud Radio Access Networks) in 2015.

Dr. Quek was honored with the 2008 Philip Yeo Prize for Outstanding Achievement in Research, the IEEE Globecom 2010 Best Paper Award, the CAS Fellowship for Young International Scientists in 2011, the 2012 IEEE William R. Bennett Prize, the IEEE SPAWC 2013 Best Student Paper Award, and the IEEE WCSP 2014 Best Paper Award.

Hyundong Shin (S?01-M?04-SM?11) received the B.S. degree in electronics engineering from Kyung Hee University, Korea, in 1999, and the M.S. and Ph.D. degrees in electrical engineering from Seoul National University, Korea, in 2001 and 2004, respectively. During his postdoctoral research at the Massachusetts Institute of Technology (MIT) from 2004 to 2006, he was with the Wireless Communication and Network Sciences Laboratory within the Laboratory for Information Decision Systems (LIDS).

In 2006, Dr. Shin joined Kyung Hee University, Korea, where he is now an Associate Professor at the Department of Electronics and Radio Engineering. His research interests include wireless communications and information theory with current emphasis on MIMO systems, cooperative and cognitive communications, network interference, vehicular communication networks, location-aware radios and networks, physical-layer security, molecular communications.

Dr. Shin was honored with the Knowledge Creation Award in the field of Computer Science from Korean Ministry of Education, Science and Technology (2010). He received the IEEE Communications Society’s Guglielmo Marconi Prize Paper Award (2008) and William R. Bennett Prize Paper Award (2012). He served as a Technical Program Co-chair for the IEEE WCNC (2009 PHY Track) and the IEEE Globecom (Communication Theory Symposium, 2012). He was an Editor for IEEE Transactions on Wireless Communications (2007-2012). He is currently an Editor for IEEE Communications Letters.