Distributed Estimation of Channel Gains in Wireless Sensor Networks

Sivagnanasundaram Ramanan, *Student Member, IEEE,* and John M. Walsh, *Member, IEEE*

***Abstract*—We consider the problem of distributed channel estimation in a sensor network which employs a random sleep strategy to conserve energy. If the N network nodes are randomly placed at unknown positions, some prior information about the channel gains can be obtained due to the path loss effect. When considered from a single node perspective this prior information is uninformative because there are on the order of N links to estimate, while there are on the order of N parameters to specify the unknown node positions. However, from a network wide channel estimation perspective, there are on the order of N^2 channel gains, but these are heavily influenced by only $3N$ position parameters. We show that expectation propagation (EP) can provide a distributed channel gain estimation algorithm which makes effective use of this prior information together with standard channel training methods. Exploiting prior information significantly improves estimate performance, as is evidenced by comparison with the prior-information-blind diffusion LMS algorithm. Provided simulation results affirm this conclusion even when shadowing is included and path loss exponents are mismatched or unknown. As communication and computation are both expensive at sensor nodes, we detail the message passing, computation, and memory requirements of both algorithms.**

***Index Terms*—Distributed estimation, channel estimation, expectation propagation, diffusion LMS.**

I. Introduction

ENERGY consumption is a key issue in wireless sensor networks [1] because they are often deployed in inaccessible terrains that forbid replacement or replenishment of the sensor node power sources [2]. While part of the energy in the sensors is spent on processing data, a sizable portion of their energy is expended on communication because of the necessary power amplification of the communications signals. This energy consumption for communications purposes can be minimized, maximizing the communications energy efficiency of the network, through distributed power control [3] if the network nodes are aware of the link gains on the network's wireless channels.

However, in many cases the sensors are deployed randomly, for instance by dropping them out of the back of a plane, and, thus, they do not initially know their positions, neighbors, or channel gains. Thus, they must first estimate the channel gains in order to determine their neighbors and to minimize transmission powers. During this initial channel gain estimation phase, power consumption may be further reduced by duty cycling [4], [5], i.e. keeping only a small subset of the sensors in a high power "awake" state at each time instant.

Following these practical constraints, this paper considers a wireless sensor network in which each sensor estimates the channel gains by collaborating with a few other network nodes. While performing this channel estimation we maintain a low average network energy consumption by employing a random sleep strategy. We apply two estimation algorithms, one derived from the Expectation Propagation (EP)[6] principle and the other the diffusion Least-Mean Squares (LMS)[7] algorithm, in order to estimate the channel gains and compare their performance in terms of estimation error.

The rest of the paper is organized as follows. In Section II, we formally state our channel gain estimation problem and model the observations made during "channel sounding". In Section III, we model the channel gains for the purpose of obtaining prior information and discuss how this prior information influences the estimates. Section IV is perhaps the most important section of this paper which discusses expectation propagation based distributed estimation of channel gains. In this section, we provide a brief introduction to the EP principle and then elaborate on how EP can be applied to distributed channel estimation in a sensor network when a random sleep strategy is employed. To compare the performance of EP in distributed channel estimation, we apply the diffusion LMS algorithm to the same estimation problem in Section V. We simulate both algorithms and provide the simulation results in Section VI in which we show that EP gives promising results compared to diffusion LMS. Issues related to the common constraints in sensor networks such as power consumption, computational ability and memory requirements are discussed in Section VII. In particular, we provide message passing overheads for both algorithms which determine the energy expended on communication between the nodes which is a significant portion of total energy expended. Also, we provide computational complexities and memory requirements of the algorithms as the performance of the algorithms are constrained by limited processing power and limited memory available at the sensor nodes. Section VIII concludes the paper.

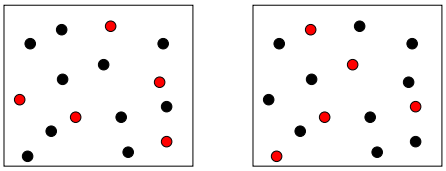
II. Problem Statement

Consider a network of N sensor nodes $1, \ldots, N$ which are randomly placed on a flat terrain to monitor a common

Manuscript received March 06, 2009; revised September 01, 2009; accepted January 30, 2010. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Mark Coates. This work was supported by the National Science Foundation under grant CCF-0728496.

Copyright ©2010 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

S. Ramanan and J. M. Walsh are with the Department of Electrical and Computer Engineering, Drexel University, Philadelphia, PA 19104, USA (e-mail:sur23@drexel.edu, jwalsh@coe.drexel.edu).



(a) Sleep cycle instant k_1 (b) Sleep cycle instant k_2

Fig. 1: Random set of sensor nodes are awake during two different sleep cycle instants.

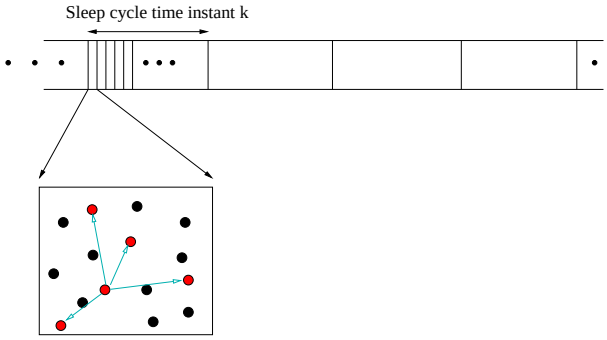


Fig. 2: One of the awake nodes during sleep cycle instant k transmits its training sequence in the first time slot.

phenomenon. Assuming symmetry of the link between two nodes, let $\mathbf{h} := [h_{i,j} \mid i,j \in \{1,2,\ldots,N\}, i < j]$ be the set of channel gains in the network, where $h_{i,j}$ is the gain of the link between nodes i and j. The goal of this paper is to estimate at each node the length $N(N-1)/2$ channel gain vector $\mathbf{h}$ by collaborating with a few other nodes and applying distributed Bayesian estimation techniques.

To reduce the power consumption during the channel estimation phase, we apply to the network a regular cyclic random sleep strategy [8], in which at each discrete time instant a randomly selected collection of d nodes are awake and each sensor maintains the same average power consumption. Each sleep cycle consists of K such discrete time instants after which the cycle repeats. Thus, if we denote the set of nodes awake at time instant k with $\mathcal{S}(k), k \in \{1,\ldots,K\}$, then $\mathcal{S}(K+k) = \mathcal{S}(k)$. Fig. 1 shows two different random set of nodes which are awake during two different sleep cycle instants $k_1, k_2 \in \{1,\ldots,K\}$, $k_1 \neq k_2$. Next denote the number of times one node is awake during a sleep cycle with c, which we require to be the same for all nodes in order to maintain equal power consumption throughout the network and thus equal node lifetime. Then, the total number of time instants in a sleep cycle is $K = \frac{c}{d}N$.

To the network model which we described above, we employ a typical wireless communication channel estimation technique, *channel sounding* or *channel training*, to estimate the channel gains of the links in the network. For communications between the nodes during the training phase and the channel estimation phase which follows the training phase, we use TDMA based medium access control [9] which increases the energy savings by avoiding collisions and retransmissions. These energy savings result at the cost of synchronization which could be achieved using the schemes proposed in the literature [10] [11] [12] [13].

To implement the TDMA based medium access control, we further divide each sleep cycle time instant k into $2c$ time slots. During each of the first c of these slots, each awake node takes turns transmitting its training sequence while all other awake nodes record their observations. For example, Fig. 2 depicts a node transmitting its training sequence at the first time slot while the other nodes which are awake during that sleep cycle instant are listening to it. The remaining slots of a sleep cycle time instant are used for the nodes to exchange estimate information in a manner to be described momentarily.

Now consider the first sleep cycle. Suppose that node i is awake at sleep cycle instant k and it transmits its training sequence $\mathbf{u}_i = [u_{i,1},\ldots,u_{i,M}]$ during its turn, where M is the length of the training sequence. Then, each node $j \in \mathcal{S}(k) \setminus \{i\}$ records its observation. We model the observation $r_{k,j,i,m}$ made for the symbol $u_{i,m}$ at the node j as a function of the channel gain $h_{i,j}$ of the link between nodes i and j as

$$r_{k,j,i,m} = h_{i,j}u_{i,m} + v_{k,j,i,m} \tag{1}$$

where $m \in \{1,\ldots,M\}$ and $v_{k,j,i,m}$ is noise which is assumed to be spatially and temporally independent and to be Gaussian distributed with mean zero and variance σ_N^2. Collect all observations made by node j for symbols $\mathbf{u}_i$ during sleep cycle instant k into a vector $\mathbf{r}_{k,j,i} := [r_{k,j,i,m} \mid m \in \{1,\ldots,M\}]$ and define the following.

$$\begin{aligned} \mathbf{r}_{k,j} &:= [\mathbf{r}_{k,j,i} \mid i \in \mathcal{S}(k) \setminus j] \\ \mathbf{r}_k &:= [\mathbf{r}_{k,i} \mid i \in \mathcal{S}(k)] \end{aligned}$$

Note that because of the random sleep strategy we use, a node which is awake during a sleep cycle instant k can gather information about only the links with the other nodes that are awake at that particular time instant.

Next each node estimates the channel gains using the information gathered and disseminates estimate information in the following sleep cycles in hopes of helping other nodes to refine their estimates. When the estimate information is disseminated, due to the limited computational abilities we assume that the nodes will take time on the order of the amount of an entire sleep cycle to decode the information (messages are encoded because they are to be sent over noisy channels) and to use it to encode any outgoing information. Therefore, after a few complete sleep cycles a sensor node will only have an opportunity to obtain information from only those nodes that can be communicated with directly or indirectly(through other nodes) within that number of sleep cycles. Denote the information available at node i after ℓ sleep cycles with $\mathbf{r}(\mathcal{T}(i,\ell))$, i.e.

$$\mathbf{r}(\mathcal{T}(i,\ell)) := \{\mathbf{r}_k | k \in \mathcal{T}(i,\ell)\} \tag{2}$$

where we define $\mathcal{P}(i,\ell)$ as the set of indices j of nodes with which node i can communicate directly or indirectly after ℓ complete sleep cycles and $\mathcal{T}(i,\ell)$ as

$$\mathcal{T}(i,\ell) := \{k |\ j \in \mathcal{S}(k) \text{ and } j \in \mathcal{P}(i,\ell)\} \tag{3}$$

Then after each complete sleep cycle ℓ, each node i refines its channel gain estimates by applying Bayesian estimation techniques, which effectively use the prior information of the channel gains together with the information $\mathbf{r}(\mathcal{T}(i,\ell))$ received from the other nodes. In particular, each node i computes its MMSE estimates of the channel gains as

$$\hat{\mathbf{h}}_i = \int \mathbf{h}\ p_{\mathbf{h}|\mathbf{r}(\mathcal{T}(i,\ell))}\ d\mathbf{h} \tag{4}$$

The prior information used by this MMSE estimator is due to the path loss effect, and this is both analytically complex and intractable because of an inverse nonlinear dependence on the node positions as we show in Section III. Since the posterior distribution $p_{\mathbf{h}|\mathbf{r}(\mathcal{T}(i,\ell))}$ is a function of the prior distribution, the computation of the MMSE estimate becomes intractable and the direct application of the Bayesian techniques is forbidden. Motivated by this problem, we apply an approximate inference algorithm, expectation propagation (EP) [6], [8], to approximate the posterior distribution, and, thus to the channel estimation problem in Section IV. We first show in the next section how one can obtain the prior information on the channel gains which is due to the path loss effect.

III. Prior Information on the Channel Gains

Statistical channel modeling studies have consistently shown that the channel gain on a link depends on 3 components: path loss, large-scale shadowing and small-scale fading [14] [15] [16] [17] [18]. The small scale fading phenomenon refers to fast variations of the received power around a nominal average power which are caused primarily by the constructive and deconstructive interference of different multipath components arriving at mobile receiver [19] [20] [18]. This fast fading, which is less important in immobile scenarios such as the one considered here, can be compensated for using channel coding if the average link gain dictated by the path loss and large scale shadowing effects can be determined. The average link gain, henceforth referred to as the channel gain in this manuscript, can in turn be estimated using periodic channel training on a link by link basis as described in the previous section. Since this average link gain is primarily determined by the path loss and large scale shadowing effects, distributions on these quantities obtained by numerous channel measurement campaigns provide significant prior knowledge about the channel gains to be estimated, as we point out presently for the path loss component.

Path loss models capture the dependence of the channel gains on the distance between transmitter and receiver. In particular, in path loss models the channel gain between two nodes separated by a distance of R is deemed proportional to R^{-n}, where n is known as the path loss exponent. Many measurement campaigns have shown that depending on the nature of the ground on which the network lies, the path loss exponent varies between 2 and 6 [14] [15] [16] [17] [18]. We have chosen a path loss exponent of 4 for our work, although our analysis is amenable to other exponents and unknown exponents as well, as simulation results will later show. For the purposes of prior information for our estimation algorithm, we then model the channel gain $h_{i,j}$ between two nodes i and j as

$$h_{i,j} \propto \|\boldsymbol{\chi}_i - \boldsymbol{\chi}_j\|_2^{-2} \tag{5}$$

where $\boldsymbol{\chi}_i$ and $\boldsymbol{\chi}_j$ are the positions of the nodes i and j, respectively.

The influence of this path loss effect has significant implications for channel estimation when viewed from a network standpoint which are far less important when channel estimation is viewed from a single link perspective (as is traditionally the case). To see this, observe that if the problem of channel estimation is viewed as a single node problem, in which each node in the network estimates only those channel gains incident on it, and then disseminates this knowledge throughout the network, each network node would be estimating $\leq N-1$ channel gains. The path loss phenomenon dictates that these gains are heavily influenced (together with large scale shadowing effects) by the positions of the sensor nodes involved, which, if the nodes are assumed to lie on a flat plain, can be specified using $2N$ real numbers (e.g. Cartesian coordinates). The number of parameters dictating these positions is larger than the number of channel gains that any one node will estimate in an uncoordinated single node approach. Thus, it is unlikely that a path loss model will provide any useful prior information for channel gain estimation carried out at a single network node, since this means the number of unknown parameters in the prior (the positions) is far larger than the number of gains to estimate.

However, when the channel estimation problem is viewed from a global network coordinated perspective, the situation changes significantly. There are a total of $N(N-1)$ (or $\frac{N(N-1)}{2}$ depending on whether symmetry is assumed) channel gains throughout the network, while all of the node positions are specified with only $2N$ real numbers (Cartesian coordinates). In this instance, the prior information offered by the path loss phenomenon is significant. Namely, the prevalence of path loss models dictates that the $N(N-1)$ channel gains are heavily biased (albeit not entirely determined by) by a model dependent on just $2N$ parameters (the node positions). Even for moderate N, that a $\frac{N(N-1)}{2}$ variate model is largely determined by $2N$ parameters is significant. In particular, the path loss phenomenon dictates that the $N(N-1)$ dimensional vector of all channel gains in the network will live in a set that is highly concentrated around a $2N$ dimensional manifold in $\mathbb{R}^{N(N-1)}$.

In fact, even if the positions of the nodes are not known, the path loss phenomenon provides significant prior information for the network channel estimation problem. To see this, suppose that the nodes are placed randomly and independently of one another, and these positions are unknown. Then consider two links which are incident on a common node i. The gains of these two links, $h_{i,j}$ and $h_{i,m}$, are functions of node positions $(\boldsymbol{\chi}_i, \boldsymbol{\chi}_j)$ and $(\boldsymbol{\chi}_i, \boldsymbol{\chi}_m)$, respectively. Clearly the random variables $h_{i,j}$ and $h_{i,m}$ are dependent because they are functions of a common random variable $\boldsymbol{\chi}_i$. Now, consider two links that are not incident on a common node. The gains of these links, $h_{i,j}$ and $h_{m,n}$, are functions of node positions $(\boldsymbol{\chi}_i, \boldsymbol{\chi}_j)$ and $(\boldsymbol{\chi}_m, \boldsymbol{\chi}_n)$, respectively. Since $h_{i,j}$ and $h_{m,n}$ are functions

of independent variables and these functions do not have a random variable in common, they are independent of one another. Thus, in a path loss dominated regime if the nodes are placed randomly and independently of one another, and these positions are unknown, any two channel gains incident on a common network node are statistically dependent. Conversely, any two channel gains which do not share any common network nodes are statistically independent. This knowledge may be expressed in terms of a prior distribution for the network channel gains.

The prior distribution of the channel gains depends on the distribution of the node positions which governs the random placement of the nodes. For the purpose of selecting a distribution for the node positions, we specify the random node positions by Cartesian coordinates in $\mathbb{R}^2$ space, the origin of which is taken to be the position of the common phenomenon. For example, the position of node i is specified by $\boldsymbol{\chi}_i \triangleq (x_i, y_i)$. We need to keep in mind few things when selecting a suitable distribution for the coordinates. The random coordinates can take continuous values; however, practically there must be a minimum separation between any two nodes. Also ideally, we would want more sensors to be placed near the phenomenon to be monitored and fewer sensors to be placed far from the phenomenon to be monitored. Considering these facts, we choose the sensor positions $\{\boldsymbol{\chi}_1, \ldots, \boldsymbol{\chi}_N\}$ to be i.i.d. according to a Gaussian distribution satisfying a minimum separation between any two nodes, although other position distributions may be equally viable and will also be amenable to our analysis. This prior distribution is both analytically complex and intractable due to the inverse nonlinear dependence on the random node positions. For that reason, we employ an approximate inference algorithm, EP, to approximate this distribution with a tractable distribution.

The ultimate aim of this paper is to demonstrate that the network channel gains can be estimated by exploiting this prior information along with the information received from standard channel training techniques discussed in the previous section. While the phenomenon of large scale shadowing also provides significant prior information which could be exploited in estimating the average link gains throughout the network, we start for the sake of simplicity with only the prior information afforded by path loss. As will be evidenced in the simulations, which include both shadowing and path loss exponent mismatch, significant estimation performance improvement can be obtained by incorporating prior information into the estimator due to path loss effects alone.

In the next section, we derive an algorithm from EP which effectively uses this prior information together with the information obtained from the channel training to estimate the channel gains (average link gains) in the network.

IV. Distributed Estimation with Expectation Propagation

This section briefly discusses expectation propagation and then describes expectation propagation based distributed estimation [21] of channel gains.

A. Expectation Propagation

For many probabilistic models of interest, working with the true posterior distribution is intractable. In such situations, the true posterior distribution must be approximated with a tractable probability distribution such that the approximate distribution is as close as possible to the true distribution. Expectation propagation [6] [22] is an approximate inference algorithm which approximates an intractable true posterior distribution having the form of product of factors with an exponential family distribution by minimizing the Kullback-Leibler divergence between the two distributions.

To mathematically describe expectation propagation, let $\mathcal{D}$ be data and $\boldsymbol{\theta}$ be latent variables. Suppose that the posterior distribution of $\boldsymbol{\theta}$ given $\mathcal{D}$ factorizes as follows.

$$p(\boldsymbol{\theta}|\mathcal{D}) = \frac{1}{p(\mathcal{D})} \prod_{i=0}^{n} f_i(\boldsymbol{\theta}) \tag{6}$$

where $f_0(\boldsymbol{\theta})$ is the prior distribution of $\boldsymbol{\theta}$. Suppose that this posterior distribution is intractable and let $q(\theta)$ be another distribution such that

$$q(\boldsymbol{\theta}) = \frac{1}{T} \prod_{i=0}^{n} \hat{f}_i(\boldsymbol{\theta}) \tag{7}$$

where T is the normalization constant. EP approximates the posterior distribution $p(\boldsymbol{\theta}|\mathcal{D})$ with distribution $q(\boldsymbol{\theta})$ by restricting the factors $\hat{f}_i(\boldsymbol{\theta})$ to be exponential family distributions and minimizing the KL divergence between the distributions in the reverse form, i.e. $KL(p\|q)$. Ideally one would want to minimize $KL(p\|q)$ in one step, but this is intractable because this involves averaging with respect to $p(\boldsymbol{\theta}|\mathcal{D})$. Thus, EP approximates each factor $\hat{f}_j(\boldsymbol{\theta})$ in turn by minimizing

$$KL\left(\frac{1}{T_j} f_j(\boldsymbol{\theta}) q^{\setminus j}(\boldsymbol{\theta}) \;\middle\|\; q^{new}(\boldsymbol{\theta})\right) \tag{8}$$

where

$$q^{\setminus j}(\boldsymbol{\theta}) = \frac{q(\boldsymbol{\theta})}{\hat{f}_j(\boldsymbol{\theta})}, \quad T_j = \int f_j(\boldsymbol{\theta}) q^{\setminus j}(\boldsymbol{\theta}) d\boldsymbol{\theta}$$

In this fashion, each factor is revised in turn and the approximation is continued for several iterations. The approximate distribution is given by $q^{new}(\boldsymbol{\theta})$. With this brief description of EP, we begin describing how EP can be applied for channel estimation.

B. Gaussian Approximation of the Prior Distribution

EP is applied in this inference problem after approximating the complex nonlinear joint prior distribution for the channel gains with a Gaussian distribution. Under this Gaussian approximation, exact statistical inference with belief propagation [23] or expectation propagation can be performed, provided the associated approximated factor graph, which is to be defined momentarily, is without loops. Presently we provide the specific information about this Gaussian approximation. We first approximate the distribution of the channel gains in dB with a Gaussian distribution with the same mean and covariance for tractability. Then we show that the distribution of

the channel gains in the linear scale can also be approximated to a Gaussian distribution.

To see this, denote the channel gains in dB with $\mathbf{h}_{dB}$. Suppose that the mean and covariance of the channel gains in dB are $\mathbf{m}$ and Σ, respectively. Then, we can write the approximate distribution of $\mathbf{h}_{dB}$ as

$$\mathbf{h}_{dB} \sim \mathcal{N}(\mathbf{m}, \mathbf{\Sigma})$$

Furthermore, we can write $\mathbf{h}_{dB}$ as

$$\mathbf{h}_{dB} = \mathbf{m} + \mathbf{w}$$

where $\mathbf{w}$ is a random vector with distribution $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$. We can now write the channel gains $\mathbf{h}$ as

$$\mathbf{h} = 10^{(\mathbf{m}/10 + \mathbf{w}/10)} = 10^{\mathbf{m}/10}\, e^{(\ln(10)/10)\mathbf{w}}$$

where element-wise operation exp and element-wise multiplication is implied. Approximating the term $e^{(\ln(10)/10)\mathbf{w}}$, the channel gains $\mathbf{h}$ can be written as

$$\mathbf{h} \approx 10^{\mathbf{m}/10}\left(1 + \frac{\ln(10)}{10}\mathbf{w}\right) \tag{9}$$

Thus the prior distribution of the channel gains can be approximated as

$$\mathbf{h} \sim \mathcal{N}\Big(10^{\mathbf{m}/10}, \Big(\frac{\ln(10)}{10}\Big)^2 \mathrm{diag}(10^{\mathbf{m}/10})\, \mathbf{\Sigma}\, \mathrm{diag}(10^{\mathbf{m}/10})\Big)$$

Even though both the initial log-normal approximation and then the normal approximation are very coarse, they make the ultimate inference algorithm tractable. We will observe in Section VI via simulation that these coarse approximations still provide sufficient prior information to greatly enhance channel estimate performance over an algorithm not employing any use of prior information.

C. Factor Graph and Expectation Propagation

We presently illustrate how EP can be applied to this inference problem under the Gaussian approximation by associating the inference problem to a probabilistic graphical model. Suppose that each node $i \in \{1, 2, \ldots, N\}$ in the network has an estimate $\mathbf{h}_i$ of $\mathbf{h}$ and all nodes have the same prior information, i.e. $p_{\mathbf{h}}(\mathbf{h}_i) = p_{\mathbf{h}}(\mathbf{h}_j)$ for all i, j.

We write a joint distribution indicating the information available to node i after ℓ complete sleep cycles as

$$p_{\mathbf{r}(\mathcal{T}(i,\ell)),\mathbf{h},\mathbf{h}(\mathcal{P}(i,\ell))} = \prod_{k\in\mathcal{T}(i,\ell)} p_{\mathbf{r}_k|\mathbf{h}} \prod_{j\in\mathcal{P}(i,\ell)} \delta(\mathbf{h}-\mathbf{h}_j)(p_{\mathbf{h}}(\mathbf{h}_j))^{\frac{1}{g(\ell)}} \tag{10}$$

where δ is the point mass distribution at zero and $g(\ell) = c(c-1)^\ell(d-1)^\ell$ is the number of sensor nodes with which node i can communicate directly or indirectly after ℓ complete sleep cycles. Also, we define $\mathbf{h}(\mathcal{P}(i,\ell))$ as

$$\mathbf{h}(\mathcal{P}(i,\ell)) := \{\mathbf{h}_j | j \in \mathcal{P}(i,\ell)\} \tag{11}$$

and $\mathbf{r}(\mathcal{T}(i,\ell))$ as in (2). Note that in (10), we have used the fact that the observations $\mathbf{r}_k$ collected at different time instants are independent given the channel gains, because all of the training sequences in the network are known ahead of time at each node.

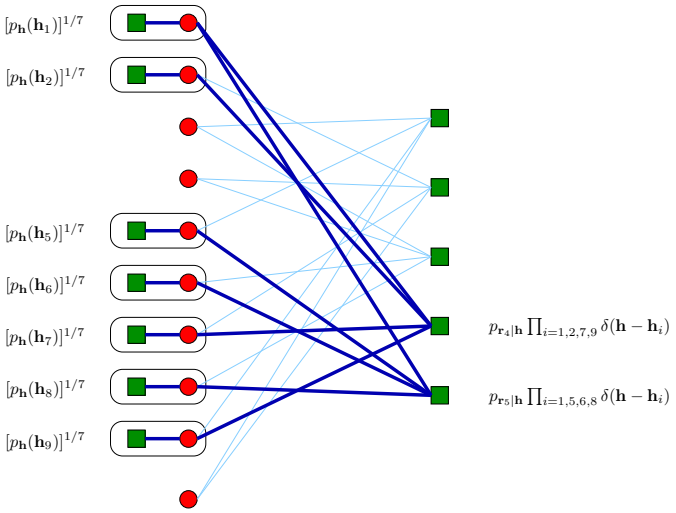


Fig. 3: An example factor graph used for EP based channel estimation with only one sleep cycle ($l = 1$)

If the posterior distribution $p(\mathbf{h}(\mathcal{P}(i,\ell))|\ \mathbf{r}(\mathcal{T}(i,\ell)))$ is approximated by applying EP using the set of approximating distributions $q(\mathbf{h}(\mathcal{P}(i,\ell))$ taking the form

$$q(\mathbf{h}(\mathcal{P}(i,\ell)) \propto \prod_{j\in\mathcal{P}(i,\ell)} q(\mathbf{h}_j) \tag{12}$$

with a Gaussian initialization to all $q(\mathbf{h}_j)$, then the optimal solution for each factor $q(\mathbf{h}_j)$ is given by the corresponding marginal of $p(\mathbf{h}(\mathcal{P}(i,\ell))|\ \mathbf{r}(\mathcal{T}(i,\ell)))$ [22]. In such a case, the expectation propagation reduces to the loopy belief propagation. These optimal solutions, the marginals of $p(\mathbf{h}(\mathcal{P}(i,\ell))|\ \mathbf{r}(\mathcal{T}(i,\ell)))$, can be computed by associating the corresponding joint distribution with a bipartite graph called a factor graph [24].

For that reason, we now associate the model in (10) with an approximate factor graph as shown in Fig. 3. We will call it an approximate factor graph since it represents an approximate joint distribution. Let us represent the sensor nodes $1, \ldots, N$ with the left side nodes (variable nodes) of the factor graph and the time instants $1, 2, \ldots, K$ of the random sleep cycle with the right side nodes (factor nodes) of the factor graph. We use an edge to connect the right node $i \in \mathcal{S}(k)$, which is awake during the sleep cycle instant k, with k^{th} left node in the factor graph which corresponds to the k^{th} sleep cycle instant.

Each node in the network, knowing the prior statistics on the channel gains, has an initial estimate of channel gains $\mathbf{h}_i = \mathbf{m}_{\mathbf{h}}$, where $\mathbf{m}_{\mathbf{h}} = 10^{\mathbf{m}/10}$. They can update their estimates by updating the statistics (mean and covariance) of the gains using the observations they made during the training phase. Once we have associated the joint distribution on the channel gains $\mathbf{h}$ and the observations $\mathbf{r}$ with a factor graph as shown in Fig. 3, we can apply EP [6] [8] to compute the posterior distribution of $\mathbf{h}$. When the conditions for belief propagation (BP) [23] are satisfied, exact statistical inference with expectation propagation (EP) can be performed, provided the associated factor graph is without loops.

The computation of the posterior distribution can be described as message passing on the factor graph as in Section IV-D. Equivalently, this computation can be explained in terms of real information exchange between the sensor nodes as in Section IV-E.

D. Message Passing on Factor Graphs

The computation of these marginals could be described using the sum-product algorithm [24] which passes messages along the edges of the factor graph to calculate beliefs. In the sum-product algorithm, message from a variable node to a factor node is computed by taking the product of messages received at the variable from the factor nodes except the factor node to which the message is to be sent. A message from a factor node to a variable node is computed by taking the product of the function corresponding to the factor node and the messages received from all variable nodes but the variable node to which the message is to be sent, and, then marginalizing. By a "message" we mean an appropriate description of the corresponding function and by a "product of messages" we mean an appropriate description of the product of corresponding functions.

We next determine an appropriate description for the functions connected to our factor graph. The factor graph represents the approximated prior joint distribution $p_{\mathbf{h}}$ of the channel gains and the conditional joint distributions $p_{\mathbf{r}_k|\mathbf{h}}$ of the observations. Consider first the approximated prior joint distribution $p_{\mathbf{h}}$ which is given by

$$p_{\mathbf{h}}(\mathbf{h}) \propto \exp\{-\frac{1}{2}[(\mathbf{h}-\mathbf{m}_{\mathbf{h}})^T\boldsymbol{\Sigma}_{\mathbf{h}}^{-1}(\mathbf{h}-\mathbf{m}_{\mathbf{h}})]\} \tag{13}$$

where $\mathbf{m}_{\mathbf{h}}$ is the mean and $\boldsymbol{\Sigma}_{\mathbf{h}}$ is the covariance. Consider next the conditional joint distribution on the observations $\mathbf{r}_{k,i}$ collected during sleep cycle instant k at node $i \in \mathcal{S}(k)$ which is given by

$$\begin{aligned} p_{\mathbf{r}_{k,i}|\mathbf{h}}(\mathbf{r}_{k,i}|\mathbf{h}) \quad \propto \quad & \exp\{-\frac{1}{2}[(\mathbf{r}_{k,i}-\mathbf{m}_{\mathbf{r}_{k,i}|\mathbf{h}})^T\boldsymbol{\Sigma}_{\mathbf{r}_{k,i}|\mathbf{h}}^{-1} \\ & (\mathbf{r}_{k,i}-\mathbf{m}_{\mathbf{r}_{k,i}|\mathbf{h}})]\} \end{aligned} \tag{14}$$

where

$$\begin{aligned} \mathbf{m}_{\mathbf{r}_{k,i}|\mathbf{h}} &:= [h_{i,j}\mathbf{u}_j | j \in \mathcal{S}(k)\setminus\{i\}] \\ \boldsymbol{\Sigma}_{\mathbf{r}_{k,i}|\mathbf{h}} &:= \sigma_N^2\mathbf{I}_{(d-1)M\times(d-1)M} \end{aligned}$$

where σ_N^2 is noise variance. Since both distributions are exponential family distributions (Gaussian), they can be easily parameterized. This enables us to select the sufficient statistics of the exponential family distributions to be

$$\mathbf{v}(\mathbf{h}) = \begin{pmatrix} \mathbf{h}_y & \mathbf{h}_z & \mathbf{h} \end{pmatrix}^T \tag{15}$$

where

$$\begin{aligned} \mathbf{h}_y &:= [h_{i,j}^2 | i,j \in \{1,\dots,N\}, i<j] \\ \mathbf{h}_z &:= [h_{i,j}h_{m,n} | i,j,m,n \in \{1,\dots,N\}, \\ &\qquad i<j, m<n, m>i] \end{aligned}$$

We can rewrite the prior distribution in terms of parameterization of the message exponential family as

$$p_{\mathbf{h}}(\mathbf{h}) \propto \exp\{-\frac{1}{2}(\mathbf{v}(\mathbf{h}).\boldsymbol{\tau} + \mathbf{m}_{\mathbf{h}}{}^T\boldsymbol{\Sigma}_{\mathbf{h}}^{-1}\mathbf{m}_{\mathbf{h}})\} \tag{16}$$

where the parameter vector $\boldsymbol{\tau}$ is

$$\boldsymbol{\tau} = \begin{pmatrix} \mathbf{a}_y & 2\mathbf{a}_z & -2\boldsymbol{\Sigma}_{\mathbf{h}}^{-1}\mathbf{m}_{\mathbf{h}} \end{pmatrix}^T \tag{17}$$

where $\boldsymbol{\Sigma}_{\mathbf{h}}^{-1} = [a_{i,j}]_{\frac{1}{2}N(N-1)\times\frac{1}{2}N(N-1)}$ and

$$\begin{aligned} \mathbf{a}_y &:= [a_{i,i} | i \in \{1,\dots,\frac{1}{2}N(N-1)\}] \\ \mathbf{a}_z &:= [a_{m,n} | m,n \in \{1,\dots,\frac{1}{2}N(N-1)\}, n>m] \end{aligned}$$

We can also rewrite the conditional joint distribution on the observations as

$$p_{\mathbf{r}_{k,i}|\mathbf{h}}(\mathbf{r}_{k,i}|\mathbf{h}) \propto \exp\{-\frac{1}{2\sigma_N^2}(\mathbf{v}(\mathbf{h}).\mathbf{t}_{k,i} + \mathbf{r}_{k,i}^T\mathbf{r}_{k,i})\} \tag{18}$$

where

$$\mathbf{t}_{k,i} = \begin{pmatrix} \boldsymbol{\nu}_{k,i} & \mathbf{0} & \boldsymbol{\mu}_{k,i} \end{pmatrix}^T \tag{19}$$

where

$$\begin{aligned} \boldsymbol{\nu}_{k,i} &:= [\mathbf{u}_n^T\mathbf{u}_n\delta(i-m)\delta(j-n) | m,n \in \{1,\dots,N\}, \\ &\qquad m<n \text{ if } i<j, m>n \text{ if } i>j,\ j \in \mathcal{S}(k)\setminus\{i\}] \\ \boldsymbol{\mu}_{k,i} &:= [-2\mathbf{u}_n^T\mathbf{r}_{k,m,n}\delta(i-m)\delta(j-n) | m,n \in \{1,\dots,N\}, \\ &\qquad m<n \text{ if } i<j, m>n \text{ if } i>j,\ j \in \mathcal{S}(k)\setminus\{i\}] \end{aligned}$$

Here note that each vector in $\mathbf{t}_{k,i}$ is of the same length as the corresponding vectors in $\mathbf{v}(\mathbf{h})$.

It is useful to note an important property of exponential family distributions before we continue. Consider a set of distributions $\{p_{\boldsymbol{\theta}_i|\mathbf{h}} | i \in \{1,\dots,L\}\}$, in which each distribution takes the form

$$p_{\boldsymbol{\theta}_i|\mathbf{h}} \propto \exp\{-\frac{1}{2}[\mathbf{v}(\mathbf{h}).\mathbf{f}_i(\boldsymbol{\theta}_i) - \mathbf{w}_i(\boldsymbol{\theta}_i)]\} \quad \forall i \in \{1,\dots,L\}$$

Then, the product of the distributions can be written as

$$\prod_{i=1}^{L} p_{\boldsymbol{\theta}_i|\mathbf{h}} \propto \exp\{-\frac{1}{2}[\mathbf{v}(\mathbf{h}).\sum_{i=1}^{L}\mathbf{f}_i(\boldsymbol{\theta}_i)]\}$$

This special property of the exponential family distribution makes the calculation of the messages easy. In particular, the appropriate description of the product of messages (functions) is the summation of the parameters of corresponding function. Thus, variable node i computes the message $\phi_{i\to k}^{(p)}$ to factor node k during sleep cycle p in terms of the messages $\varphi_{k'\to i}^{(p-1)}$ received from the factor nodes during sleep cycle $p-1$ as

$$\phi_{i\to k}^{(p)} = \sum_{k'\in\mathcal{N}(i)\setminus\{k\}} \varphi_{k'\to i}^{(p-1)} \tag{20}$$

where $\mathcal{N}(i) := \{k | i \in \mathcal{S}(k)\}$. Factor node k computes the message $\varphi_{k\to i}^{(p)}$ to variable node i during sleep cycle p in terms of the messages $\phi_{i'\to k}^{(p-1)}$ received from the variables nodes during sleep cycle $p-1$ and the corresponding factor as

$$\varphi_{k\to i}^{(p)} = \mathbf{t}_k + \sum_{i'\in\mathcal{S}(k)\setminus\{i\}} \phi_{i'\to k}^{(p-1)} \tag{21}$$

where $\mathbf{t}_k := \sum_{i\in\mathcal{S}(k)}\mathbf{t}_{k,i}$. The number of iterations ℓ that EP is to be run is decided ahead of time and the message passing is initialized by setting $\phi_{i\to k}^{(0)} = \boldsymbol{\tau}/g(\ell)$. At the final iteration, variable node i sums its incoming messages to get

the parameters corresponding to the approximated posterior distribution. We explained the computation of the posterior distribution on the factor graph treating the factor nodes as if they were processors. Although this is mathematically correct, there are no such processors in reality. Since all the processors are located at the sensor nodes in reality, it is interesting to see the exchange of information between the sensor nodes during the computation of the posterior distribution.

E. Information Exchange between the Sensor Nodes

We presently describe the information exchange that occurs between the sensor nodes during the computation of the posterior distribution. Each node initializes the parameter vector $\boldsymbol{\tau}$ of the prior distribution $p_{\mathbf{h}}(\mathbf{h})$ to the values in (17). Also, each node i initializes the $\frac{N(N-1)}{2}$ vector $\boldsymbol{\mu}_{k,i}$ in (19) to all zeros. The remaining vectors in (19) need not be initialized or passed during message passing, because the parameters corresponding to the vector $\mathbf{h}_y$ in $\mathbf{v}(\mathbf{h})$ involve only the training sequences which are already available at each node. Thus, each node can calculate the parameters corresponding to the vectors $\mathbf{h}_y$ and $\mathbf{h}_z$ using the information available at the node.

During the *first* sleep cycle $p = 1$, during each sleep cycle instant k each node $i \in \mathcal{S}(k)$ calculates $-2\mathbf{u}_j^T \mathbf{r}_{k,i,j}$ for each other awake node $j \in \mathcal{S}(k) \backslash \{i\}$, and adds it to the appropriate element of the vector $\boldsymbol{\mu}_{k,i}$.

$$[\boldsymbol{\mu}_{k,i}]_{i,j} = [\boldsymbol{\mu}_{k,i}]_{i,j} - 2\mathbf{u}_j^T \mathbf{r}_{k,i,j} \tag{22}$$

Here, by $[\boldsymbol{\mu}_{k,i}]_{i,j}$ we mean the element corresponding to node j (corresponding to the channel gain $h_{i,j}$) in the vector $\boldsymbol{\mu}_{k,i}$. At every iteration p and every sleep cycle instant k, the awake nodes $i \in \mathcal{S}(k)$ multiply $p_{\mathbf{r}_{k,i}|\mathbf{h}}$ with the functions corresponding to the messages obtained in all of the *other* $c-1$ sleep cycle time instants ($\mathcal{N}(i) \setminus \{k\}$) it was awake during the previous $(p-1)$th sleep cycle to obtain the outgoing message. Since all the functions are from the same exponential family with sufficient statistics $\mathbf{v}(\mathbf{h})$, when the messages are multiplied the parameters of the messages sum up as explained above. Each node then passes the parameters corresponding to the product of the functions. Furthermore, nodes need to pass only the vectors $\boldsymbol{\mu}_{k,i}$ because each node can calculate the other vectors based on the information available at the node. Thus, the nodes $i \in \mathcal{S}(k)$ sum $\boldsymbol{\mu}_{k,i}$ with the vectors $\boldsymbol{\lambda}_{k' \to i}^{(p-1)}$ to obtain $\rho_{i \to k}^{(p)}$.

$$\rho_{i \to k}^{(p)} = \boldsymbol{\mu}_{k,i} + \sum_{k' \in \mathcal{N}(i) \backslash \{k\}} \boldsymbol{\lambda}_{k' \to i}^{(p-1)} \tag{23}$$

The $\frac{N(N-1)}{2}$ dimensional vector $\rho_{i \to k}^{(p)}$ is then broadcast to all other awake nodes.

Each node $i \in \mathcal{S}(k)$ then sums the $d-1$ messages $\rho_{j \to k}^{(p)}$ it heard from the other awake nodes $j \in \mathcal{S}(k) \setminus \{i\}$ with $\boldsymbol{\mu}_{k,i}$, and stores the result in $\boldsymbol{\lambda}_{k \to i}^{(p)}$.

$$\boldsymbol{\lambda}_{k \to i}^{p} = \boldsymbol{\mu}_{k,i} + \sum_{i' \in \mathcal{S}(k) \backslash \{i\}} \boldsymbol{\rho}_{i' \to k}^{p} \tag{24}$$

At the final iteration, node i sums $\boldsymbol{\lambda}_{k \to i}^{(p)}$ from k in all c sleep cycle instants it was awake, adds it to $\boldsymbol{\tau}$, and multiplies the result by the (offline computed $\frac{N(N-1)}{2} \times \frac{N(N-1)}{2}$ dimensional) new covariance matrix formed from the training data to get its estimate. We summarize this algorithm below.

- Initialize $\boldsymbol{\mu}_{k,i}$ to all zeros and $\boldsymbol{\tau}$ as in (17)
- During the 1st sleep cycle $p = 1$ and each sleep cycle instant k, at each node $i \in \mathcal{S}(k)$ calculate $[\boldsymbol{\mu}_{k,i}]_{i,j}$ as in (22).
- During sleep cycle $1 \leq p \leq \ell - 1$ and each sleep cycle instant k, at each node $i \in \mathcal{S}(k)$ repeat:
  - Calculate the message $\rho_{i \to k}^{(p)}$ as in (23) and broadcast it.
  - Sum the messages $\rho_{j \to k}^{(p)}$ received from nodes $j \in \mathcal{S}(k) \backslash \{i\}$ with $\boldsymbol{\mu}_{k,i}$ as in (24) to get $\boldsymbol{\lambda}_{k \to i}^{p}$ and go to sleep.
- At final sleep cycle $p = \ell$, at node i: Sum $\boldsymbol{\lambda}_{k \to i}^{(p)}$ from $k \in \mathcal{N}(i)$, add to $\boldsymbol{\tau}$ and multiply with the new covariance matrix to get the estimate.

F. Convergence and Sensitivity of the Algorithm

Having described the EP based algorithm for distributed estimation of channel gains, we next discuss the convergence properties and its sensitivity to node failures. As it was discussed in Section II, after ℓ complete sleep cycles each (variable) node i will have communicated with nodes up to 2ℓ edges away from it in the factor graph. For any finite number of iterations ℓ, as the number of nodes $N \to \infty$ the subgraph which has root at i and contains the nodes no more than 2ℓ edges away from i becomes a tree with probability $\to 1$ [25] [8]. When applied for an appropriate finite number of iterations in such a case, our algorithm converges to the approximate posterior distribution of the channel gains given the observations at nodes no more than 2ℓ edges away from node i, because after approximating the prior distribution this equivalent to applying BP and it is well known that BP converges on trees [23] [24].

Under this convergence assumption (which is the case for larger networks), we now examine the robustness of our algorithm to node failures. Consider the subgraph (tree) which has root at node i and contains the nodes no more than 2ℓ edges away from i. Suppose that one of the internal nodes in the tree has failed. This node failure results in a situation in which node i cannot exploit the observations of those nodes which should be communicated with through the failed node. However, as it can be seen from the discussion in Section IV-E, the information exchange between the sensor nodes will continue until the algorithm converges to a solution which is less accurate than it would have been otherwise.

V. Distributed Estimation with Diffusion LMS

Since the ultimate aim of this paper is to demonstrate that the prior information can be effectively used to estimate network channel gains, we compare the performance of our EP based estimation algorithm with an another algorithm, the diffusion LMS [7], which does not make use of the prior distribution. The diffusion LMS uses only the observations made during the training phase to estimate the channel gains.

Suppose that each node has a copy of the channel gain vector $\mathbf{h}$ and it takes an initial value of $\mathbf{m_h}$. If node i transmits its training sequence during a sleep cycle instant k, then all other nodes which are awake during the sleep cycle instant k have access to $\{u_{i,m}, r_{k,i',i,m}\}$ where $i' \in \mathcal{S}(k) \setminus \{i\}$ and $u_{i,m}$ is the input regression signal and $r_{k,i',i,m}$ is the desired signal. Note that $u_{i,m}$ and $r_{k,i',i,m}$ obey the equation

$$r_{k,i',i,m} = h_{i,i'} u_{i,m} + v_{k,i',i,m}$$

The network nodes $i' \in \mathcal{S}(k) \setminus \{i\}$ can use the diffusion LMS algorithm [7] to estimate $h_{i,i'}$. Denote the estimate of $h_{i,i'}$ at time instant m of sleep cycle instant k by $\hat{h}_{i,i'}^{k,m}$. Then,

$$\hat{h}_{i,i'}^{k,m} = \hat{h}_{i,i'}^{k,m-1} + \mu u_{i,m}(r_{k,i',i,m} - \hat{h}_{i,i'}^{k,m-1} u_{i,m}) \qquad (25)$$

where μ is the step size.

At the end of each sleep cycle instant k, the nodes that are awake diffuse their estimates to get the combined estimate $\tilde{\mathbf{h}}^k$ as

$$\tilde{\mathbf{h}}^k = \sum_{i \in \mathcal{S}(k)} a(k,i) \hat{\mathbf{h}}_i^k \qquad (26)$$

where $\hat{\mathbf{h}}_i^k$ is the estimate of $\mathbf{h}$ at node i at the end of the sleep cycle instant k and $a(k,i)$ satisfy $\sum_{i \in \mathcal{S}(k)} a(k,i) = 1$. The nodes $i \in \mathcal{S}(k)$ use the combined estimate $\tilde{\mathbf{h}}^k$ for estimation during the later sleep cycle instants.

We summarize this algorithm below.

- At each node i, initialize $\hat{\mathbf{h}}_i$ to $\mathbf{m_h}$
- During each sleep cycle and sleep cycle instant k, at each node $i \in \mathcal{S}(k)$ repeat:
  - For all $i' \in \mathcal{S}(k) \setminus \{i\}$ calculate estimate $\hat{h}_{i,i'}^{k,m}$ as in (25).
  - At the end of sleep cycle instant k, diffuse the estimate to get $\tilde{\mathbf{h}}^k$ as in (26).

VI. Simulation Results

We have simulated the algorithms described in the previous sections to estimate the channel gains in a network and have plotted the estimation errors for both algorithms. We describe the experiment and present the results in this section. In our experiment, we estimate the channel gain vector $\mathbf{h}$ for a network with 20 sensors applying EP and LMS. We selected a moderate size (20 nodes) network for our simulations, because testbeds on which initial studies can be done consist of nodes on the order of 10 [26]. The network is formed as described below. Candidate sensor positions are generated on the plane $\mathbb{R}^2$ such that they are i.i.d. and Gaussian distributed with mean zero and variance 1. These candidate sensor positions are then refined to actual sensor positions by keeping only those positions that are 0.08 apart from one another, because when the separation is less than 0.08 the channel gains become unrealistically large. A random sleep strategy with $K = 10$ and $d = 4$ is applied to this network, where the value of d is selected by simulating the algorithm for different values of d for fixed N, K and by choosing the one which gives better estimation error performance.

Although the diffusion LMS does not require the statistics of these channel gains for operation, EP requires the statistics for the estimation of these channel gains. As discussed in Section III, the joint prior distribution of the channel gains is analytically complex and intractable. Thus, we empirically calculate the statistics (mean and covariance matrix) of the channel gains using many channel gains generated by plugging in sensor positions in the equation obtained by applying proportionality constant 1 to (5). Then, we generate a new set of channel gains as described in Section VI-A to test the algorithms with. Next, we generate the BPSK training sequences of length 1000 randomly and uniformly. We run 1000 Monte Carlo simulations for each algorithm using a noise variance of $\sigma_N^2 = 1$ for this experiment.

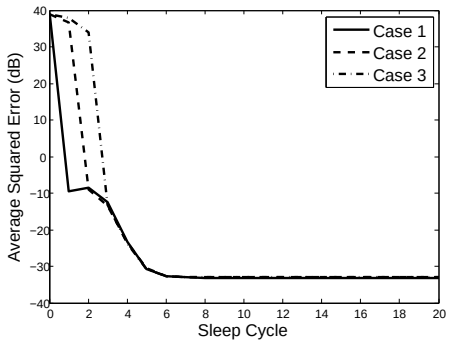
Due to the random sleep strategy and to the local nature of both algorithms, after ℓ iterations the observations made at a particular network node can propagate to the nodes only up to 2ℓ edges away from that node in the factor graph. Thus, after ℓ iterations each node will have observed information (either directly or indirectly) about only a subset of the network links and this subset differs for each node. To plot the average estimation error we consider only these subsets of links, because including the estimation errors of those channels unobserved gives large average estimation errors since the nodes cannot make good estimates of the unobserved channels unless the correlations between the channels incident on the same node is very (physically unrealistically) large. Say node i has information about subset $\mathbf{h}_{i,\ell}$ of links after ℓ iterations. Then, we calculate the average squared estimation error first by averaging the squared estimation errors of $\mathbf{h}_{i,\ell}$ at each node i and then averaging over all nodes. Note that, here ℓ can be chosen independently from the number of iterations that we run the algorithm and $\mathbf{h}_{i,\ell_1} \subseteq \mathbf{h}_{i,\ell_2}$ for $\ell_1 \leq \ell_2$. Thus, one may consider different such $\mathbf{h}_{i,\ell}$ (each corresponds to different ℓ) at each node i and plot the average estimation error. For simplicity, we consider the subsets $\mathbf{h}_{i,\ell}$ only for $\ell = 1, 2, 3$ and plot the average estimation errors in Section VI-A. We call each of these cases 'Case 1', 'Case 2' and 'Case 3', respectively.

A. Comparison of EP and Diffusion LMS

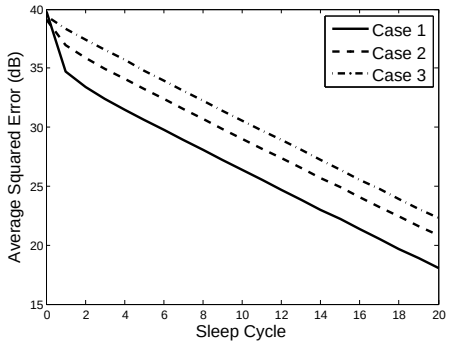
We now simulate EP and the diffusion LMS and compare the performance of the two algorithms. Using these simulations, we show that although EP utilizes a path loss model it is also robust to the shadowing effects in the channel gains.

We do not consider the fading effects here, because fading is less important in immobile scenarios such as the one considered here and any multipath effects caused by the fixed reflectors can be included in the log-normal shadowing. Thus, we generate the channel gains to be estimated with path loss and log-normal shadowing effects. Shadowing effect is included in the channel gains by first generating gains as described above, and, then adding a Gaussian variable distributed $\mathcal{N}(0, 18.5)$ to the associated power in dB.

Fig. 4a shows the average squared estimation error of the observed channel gains in dB when EP is applied. Interestingly, even with the Gaussian approximation of the joint prior



(a) EP



(b) LMS

Fig. 4: Average squared estimation error of only those channel gains observed directly or indirectly by the nodes after 1st, 2nd and 3rd sleep cycles

distribution and inclusion of shadowing effects, EP yields impressive estimation error performance for estimation of the true channel gains. Note that there is a drastic change in the average estimation error after first sleep cycle for 'Case 1' since these links were observed by the nodes directly. Also note that the drastic change shifts to the second sleep cycle and third sleep cycle for 'Case 2' and 'Case 3', respectively, as is to be expected. The estimation errors continue to decrease even after this initial drastic change because the nodes make use of the correlation with the other channel gains observed in subsequent iterations to refine estimates of these particular channel gains.

Fig. 4b shows the average squared error of the estimated channel gains when diffusion LMS is used with step size $\mu = 1$. This step size was chosen to be the maximum step size for which the diffusion LMS does not diverge in order to give the algorithm the chance to converge as quickly as possible, since EP converges faster. Note that EP gives better performance than the diffusion LMS when the network is required to estimate the channel coefficients within a small number of sleep cycles.

B. Mismatch of Path loss Exponent

The presented EP channel estimation algorithm can still be used to estimate the channel gains with small errors even if the actual path loss exponent differs by a small range of values from the path loss exponent used in the prior distribution. To show this, we generate the prior statistics with path loss exponent 4 and the channel gains to be estimated as described in Section VI-A but with path loss exponent 6. Fig. 5 shows the average estimation error when EP is applied for this experiment and proves that our algorithm is robust to path loss exponent mismatch between 4 and 6 when the nodes are placed i.i.d. according to a Gaussian distribution.

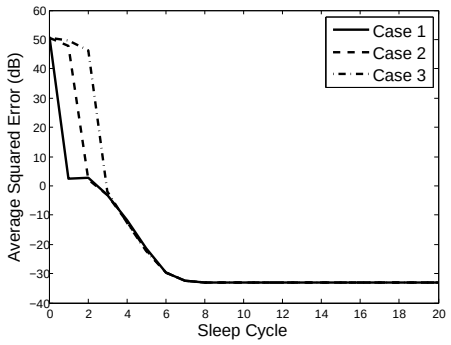


Fig. 5: Average squared estimation error when EP (uses path loss exponent 4) is applied to estimate channel gains having path loss exponent 6

VII. Message Passing Overhead, Computational Complexity and Memory Requirement

Although the estimation error performance is the most important aspect of channel estimation in sensor networks, this is not the only aspect that must be considered. Any operation in sensor networks is constrained by factors such as power consumption, computational ability and memory requirements. In this section, we discuss the issues related to these constraints. In particular, we calculate the number of messages that are to be passed between the nodes, the number of computations required and the number of memory required for both algorithms during distributed estimation of channel gains.

A. Message Passing Overhead for EP and diffusion LMS

A significant portion of the power in the nodes is expended on internode communication and this heavily depends on the message passing overhead. Thus we compute the message passing overhead of algorithms in this subsection.

The message passing overhead for EP can be calculated by analyzing the message exponential family. Although the sufficient statistics vector $\mathbf{v}(\mathbf{h})$ of the message exponential family has a dimension of $2\,\frac{N(N-1)}{2} + \frac{[N(N-1)-1][N(N-1)]}{8}$, each node i needs to pass only $\frac{N(N-1)}{2}$ parameters $\boldsymbol{\mu}_{k,i}$ which are corresponding to the vector $\mathbf{h}$ in $\mathbf{v}(\mathbf{h})$ in (15). The parameters corresponding to the vector $\mathbf{h}_y$ in $\mathbf{v}(\mathbf{h})$ in (15) involve only the training sequences that are already available at each node. Similarly, the parameters corresponding to $\mathbf{h}_z$ in $\mathbf{v}(\mathbf{h})$ in (15) involve only the elements of the covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{h}}$ of the prior distribution. Thus, each node can calculate the parameters corresponding to the vectors $\mathbf{h}_y$ and $\mathbf{h}_z$ using the internal information. Thus, the number of messages required to be passed for EP during a complete sleep cycle is $Kd\frac{N(N-1)}{2}$.

TABLE I: Message passing overhead, computational complexity and memory requirements

	EP	Diffusion LMS
Message passing overhead (per sleep cycle)	$Kd\frac{N(N-1)}{2}$	$Kd\frac{N(N-1)}{2}$
Computational complexity Additions (per P cycles)	$Kd(d-1)(M+1)+PK(dc+d^2)\frac{N(N-1)}{2}$ $+\frac{N^2(N-1)}{2}\left(c+\frac{N(N-1)}{2}\right)$	$PKd\left(2(d-1)M+d\frac{N(N-1)}{2}\right)$
Multiplications (per P cycles)	$Kd(d-1)M+N\left(\frac{N(N-1)}{2}\right)^2$	$PKd(d-1)\left(2M+\frac{N(N-1)}{2}\right)$
Memory requirements	$\frac{N(N-1)(N^2-N+2c+5)}{4}+MN$	$\frac{N(N-1)}{2}+MN$

When the diffusion LMS is applied to the channel gain estimation, each node calculates its own estimates and, then, diffuses its estimates at the end of the sleep cycle instant. Thus, the message passing overhead for the diffusion LMS is simply the number of channel gains in the network. The total message passing overhead for the diffusion LMS is $Kd\frac{N(N-1)}{2}$ per sleep cycle. Thus, the total message passing overhead is exactly the same for both algorithms.

One might argue that not all $\frac{N(N-1)}{2}$ parameters in the vectors need to be passed, because when EP is applied many elements in $\boldsymbol{\mu}_{k,i}$ are zero and when diffusion LMS is applied many elements in $\hat{\mathbf{h}}_i^k$ remain unchanged. However, if one wants to take the zero and unchanged elements into account to reduce the message passing overhead, one must keep track of the elements that change and perform some indexing when he passes messages. Thus, clearly there is a trade off between reduction in the number of parameters that are passed and increment in the number of computations. It is unclear whether there is any saving on the energy consumption at the nodes from such a practice. For that reason, we pass all $\frac{N(N-1)}{2}$ parameters in the vectors.

B. Computational Complexity of EP and diffusion LMS

We calculate the number of computations required for each algorithm based on the expressions given in Section IV-E and Section V. We first consider the computational complexity of EP.

The computations related to the parameters of the prior distribution $p_{\mathbf{h}}(\mathbf{h})$ are done offline. Each instance of (22) requires M multiplications and $M+1$ additions, and is run in each of d nodes for $d-1$ other nodes in each sleep cycle instant in the first iteration, giving a total of $d(d-1)M$ multiplications and $d(d-1)(M+1)$ additions spent in (22) per sleep cycle instant in the first iteration (over the whole network). Each instance of (23) requires $c\frac{N(N-1)}{2}$ additions, and is run in each of d nodes in each sleep cycle instant, giving a total of $dc\frac{N(N-1)}{2}$ additions spent in (23) in each sleep cycle instant (over the whole network). Each instance of (24) requires $d\frac{N(N-1)}{2}$ additions, and is run in each of d nodes in each sleep cycle instant, giving a total of $d^2\frac{N(N-1)}{2}$ additions spent in (23) in each sleep cycle instant (over the whole network).

Finally, the final iteration requires each of N nodes to perform $(c+1)\frac{N(N-1)}{2}$ additions, then multiplication of a matrix times a vector requiring $\left[\frac{N(N-1)}{2}\right]^2$ multiplications and $\frac{N(N-1)}{2}\left[\frac{N(N-1)}{2}-1\right]$ additions. This gives a total of $N\left[(c+1)\frac{N(N-1)}{2}+\frac{N(N-1)}{2}\left(\frac{N(N-1)}{2}-1\right)\right]$ additions and $N\left[\frac{N(N-1)}{2}\right]^2$ multiplications over the entire network during the last iteration.

For the diffusion LMS, (25) is run during each sleep cycle instant k at each awake node $i\in\mathcal{S}(k)$ for every other awake node $i'\in\mathcal{S}(k)\backslash\{i\}$ for each training instant $m\in\{1,\ldots,M\}$. One calculation of (25) consists of two multiplications and two additions. Thus, $2d(d-1)M$ multiplications and $2d(d-1)M$ additions are spent in each sleep cycle time instant on calculations of the form (25). Then the nodes diffuse the estimates and calculate (26). This involves $d\frac{N(N-1)}{2}$ multiplications and $(d-1)\frac{N(N-1)}{2}$ additions in each of d sensor nodes, giving a total of $d^2\frac{N(N-1)}{2}$ multiplications and $d(d-1)\frac{N(N-1)}{2}$ additions spent on calculations for (26) per sleep cycle time instant.

If a total of P sleep cycles are performed, required number of additions and multiplications over the entire network for EP and diffusion LMS are given in Table I.

C. Memory Requirement

It is also important to analyze memory requirement of the algorithms, because in some applications memory is limited. We first calculate the memory requirement for EP. When EP is employed for channel gain estimation, some parameters are calculated offline and stored in the sensor nodes while the rest of the parameters are calculated and stored online. First, consider the offline computed parameters. Each node requires storing information on the prior distribution and the training sequences of all nodes. The storage of the parameter vector $\boldsymbol{\tau}$ and the training sequences $\mathbf{u}=\{\mathbf{u}_i|i\in\{1,\ldots,N\}\}$ requires $\frac{N(N-1)(N^2-N+6)}{8}$ and MN memory locations, respectively. Also, each node initializes the vector $\boldsymbol{\mu}_{k,i}$ to all zeros which requires $\frac{N(N-1)}{2}$ memory. Furthermore, at the final iteration each node utilizes a offline computed $\frac{N(N-1)}{2}\times\frac{N(N-1)}{2}$ matrix to calculate its estimate. Since this is a symmetric matrix, this requires $\frac{N^2(N-1)^2}{8}$ memory locations at each node. When EP is run, each node will allocate memory to store messages that are to be received during different sleep cycle instants. This will require $c\frac{N(N-1)}{2}$ memory locations at each node. Thus, when EP is employed each node requires a total of $\frac{N(N-1)(N^2-N+2c+5)}{4}+MN$ memory locations.

When diffusion LMS is employed, each node initializes the estimates and stores the training sequence offline. The

initialization of the estimates require $\frac{N(N-1)}{2}$ memory while the storage of the training sequence requires MN memory. The diffusion LMS does not require any additional memory when the algorithm is in progress. Thus, the diffusion LMS requires a total of $\frac{N(N-1)}{2} + MN$ memory at each node.

Table I summarizes the results derived in Section VII.

VIII. Conclusion

We considered a distributed channel estimation problem in a sensor network which employs a random sleep strategy to conserve energy. We modeled the channel gains in the network using a path loss model and gathered information about these channels using channel sounding. One might want to estimate the channel gains by using the prior information and information gathered from channel sounding, and by applying traditional Bayesian estimation techniques. We showed that a direct application of Bayesian estimation is forbidden in this case due to the intractability of the prior distribution, and showed that EP can be applied to this channel estimation problem through approximation of the posterior distribution.

We compared the performance of EP with the diffusion LMS and showed EP gives a better estimation error performance using the simulation results. We also proved that although our algorithm utilizes a path loss model with an a priori fixed path loss exponent, it is robust to shadowing effects and variation of path loss exponents. Finally, we compared the message passing overhead, computational complexity and memory requirements of both algorithms.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Computer Networks (Elsevier)*, vol. 38, no. 4, pp. 393–422, Mar. 2002.

[2] A. Ephremides, "Energy concerns in wireless networks," *IEEE Wireless Communications*, vol. 9, no. 4, pp. 48–59, Aug. 2002.

[3] V. Kawadia and P. R. Kumar, "Principles and protocols for power control in wireless ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 1, pp. 76–88, Jan. 2005.

[4] C. fan Hsin and M. Liu, "Randomly duty-cycled wireless sensor networks: Dynamics of coverage," *IEEE Transactions on Wireless Communications*, vol. 5, no. 11, pp. 3182–3192, Nov. 2006.

[5] O. Dousse, P. Mannersalo, and P. Thiran, "Latency of wireless sensor networks with uncoordinated power saving mechanisms," in *Proceedings of Mobihoc*, 2004.

[6] T. P. Minka, "A family of algorithms for approximate bayesian inference," PhD Thesis, Massachusetts Institute of Technology, 2001.

[7] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks: Formulation and performance analysis," *IEEE Transactions on Signal Processing*, vol. 56, no. 7, pp. 3122–3136, July 2008.

[8] J. M. Walsh, S. Ramanan, and P. A. Regalia, "Optimality of expectation propagation based distributed estimation for wireless sensor network initialization," in *IEEE International Workshop on Signal Processing Advances for Wireless Communications*, 2008.

[9] K. Kredo II and P. Mohapatra, "Medium access control in wireless sensor networks," *Computer Networks (Elsevier)*, vol. 51, no. 4, pp. 961–994, Mar. 2007.

[10] G. Scutari, S. Barbarossa, and L. Pescosolido, "Distributed decision through self-synchronizing sensor networks in the presence of propagation delays and asymmetric channels," *IEEE Transactions on Signal Processing*, vol. 56, no. 4, pp. 1667–1684, Apr. 2008.

[11] S. Barbarossa and G. Scutari, "Bio-inspired sensor network design," *IEEE Signal Processing Magazine*, vol. 24, no. 3, pp. 26–35, May 2007.

[12] B. Sundararaman, U. Buy, and A. D. Kshemkalyani, "Clock synchronization for wireless sensor networks: a survey," *Ad Hoc Networks (Elsevier)*, vol. 3, no. 3, pp. 281–323, May 2005.

[13] F. Sivrikaya and B. Yener, "Time synchronization in sensor networks: A survey," *IEEE Network*, vol. 18, no. 4, pp. 45–50, July-Aug. 2004.

[14] J. B. Andersen, T. S. Rappaport, and S. Yoshida, "Propagation measurements and models for wireless communications channels," *IEEE Communications Magazine*, vol. 33, no. 1, pp. 42–49, Jan. 1995.

[15] D. Cassioli, M. Z. Win, and A. F. Molisch, "The ultra-wide bandwidth indoor channel: From statistical model to simulations," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 6, pp. 1247–1257, Aug. 2002.

[16] E. Green and M. Hata, "Microcellular propagation measurements in an urban environment," in *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, 1991.

[17] A. Saleh and R. Valenzuela, "A statistical model for indoor multipath propagation," *IEEE Journal on Selected Areas in Communications*, vol. 5, no. 2, pp. 128–137, Feb. 1987.

[18] A. Goldsmith, *Wireless Communications*. Cambridge University Press, 2005.

[19] A. F. Molisch, *Wireless Communications*. John Wiley & Sons, Ltd., 2005.

[20] G. L. Stuber, *Principles of Mobile Communication*, 2nd ed. Kluwer Academic Publishers, 2001.

[21] A. Ribeiro and G. B. Giannakis, "Bandwidth-constrained distributed estimation for wireless sensor networks - part i: Gaussian case," *IEEE Transactions on Signal Processing*, vol. 54, no. 3, pp. 1131–1143, Mar. 2006.

[22] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer Science + Business Media, LLC, 2006.

[23] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Constructing free-energy approximations and generalized belief propagation algorithms," *IEEE Transactions on Information Theory*, vol. 51, no. 7, pp. 2282–2312, July 2005.

[24] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[25] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 599–618, Feb. 2001.

[26] W. R. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive protocols for information dissemination in wireless sensor networks," in *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, 1999.

Sivagnanasundaram Ramanan (S'09) received the B.Sc. degree in electrical and electronic engineering from the University of Peradeniya, Sri Lanka in 2006. He is currently working towards the Ph.D. degree in electrical engineering at Drexel University, Philadelphia, PA.

His research interests include distributed estimation, distributed source coding and wireless communication.

John M. Walsh (S'01-M'07) received the B.S. (*magna cum laude*), M.S. and Ph.D. degrees in electrical and computer engineering from Cornell University, Ithaca, NY in 2002, 2004, and 2006, respectively.

In September 2006, he joined the Department of Electrical and Computer Engineering at Drexel University, Philadelphia, PA, where he is currently an Assistant Professor. His current research interests include: (a)the performance and convergence of distributed collaborative estimation in wireless sensor networks via expectation propagation, (b)delay mitigating codes and rate-delay tradeoffs in multipath routed and network coded networks, and (c) joint source separation and identification.

Dr. Walsh is a member of HKN and TBP.