Distributed Estimation of channel Gains in Sensor Networks

Sivagnanasundaram Ramanan, John MacLaren Walsh
Dept. of Electrical and Computer Engineering
Drexel University
Philadelphia, PA 19104, USA

***Abstract*—Channel estimation algorithms are proposed for the initialization phase of a wireless sensor network, during which the nodes are still unaware of their neighbors or channel qualities. Both expectation propagation (EP) and diffusion least mean squares (LMS) algorithms are applied to this problem of channel estimation while the network employs a random duty cycling strategy to conserve power. Simulations results which show EP giving better performance are given for both algorithms for a small network.**

I. Introduction

Energy consumption is a key issue in wireless sensor networks because they are often deployed in inaccessible terrains that forbid replacement or replenishment of the sensor node power sources. While part of the energy in the sensors is spent on processing data, a sizable portion of their energy is expended on communication between the nodes 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 if the network nodes are are aware of the link gains on the network's wireless channels.

However, in many cases sensors are deployed randomly, for instance by dropping them out of the back of a plane, and, thus, they do not initially know their neighbors or the 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 [1], [2], 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, Expectation Propagation (EP) and diffusion Least-Mean Squares (LMS), in order to estimate the channel gains, and compare their performance in terms of estimation error.

II. System Model

Consider a network of N sensor nodes $s_1, \ldots, s_N$ which are randomly placed on a flat terrain according to some position probability distribution. Here we choose the sensor positions $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ 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. We apply to the network a regular cyclic random sleep strategy [3], 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. Define the set of nodes awake at time instant k to be $\{\mathcal{S}(k) \mid k \in \{1, \ldots, K\}\}$, where K is the total number of time instants in a sleep cycle. Each sleep cycle consists of K time instants, after which the cycle repeats, so that $\mathcal{S}(K+k) = \mathcal{S}(k)$. Let us 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 equal power consumption throughout the network and thus equal node lifetime. Then, the total number of time instants in a sleep cycle is $K = \frac{c}{d}N$.

Consider next the channel gains on links in the network. Statistical channel modeling studies have consistently shown that the channel gain on a link between any two sensor nodes heavily depends on the distance between them. Frequently this dependence is captured via a path loss model ([4] Ch. 1, [5] Ch. 7), in which the channel gain between two nodes separated by a distance of R is deemed proportional to R^{-n}, where n is known as the pathloss exponent. It has been practically shown that depending on the nature of the ground on which the network lies, the path loss exponent varies between 2 and 6. We have chosen a path loss exponent of 4 for our work, although our analysis is amenable to other exponents as well. We then model the channel gain $h_{i,j}$ between two nodes i and j as distributed a priori according to

$$h_{i,j} \propto \|\mathbf{x}_i - \mathbf{x}_j\|_2^{-4}$$

where $\mathbf{x}_i$ and $\mathbf{x}_j$ are the random positions of the nodes i and j, respectively.

Thus, 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.

S. Ramanan and J. M. Walsh were supported by the National Science Foundation under grant CCF-0728496

This knowledge may be expressed in terms of a prior distribution for the network channel gains, which due to the inverse nonlinear dependence on the node positions is both analytically complex and intractable. Thus, it is interesting to consider channel estimation statistical inference algorithms such as EP [6], [3] which can make effective use of this prior information while not falling prey to unbearable computational complexity. This is achieved in the present context by approximating the complex nonlinear joint prior distribution for the channel gains with a Gaussian with the same mean and covariance. Under this Gaussian approximation, exact statistical inference with belief propagation can be performed, provided the associated approximated factor graph is without loops. Presently we provide the specific information about this Gaussian approximation. More details about the EP iterative method will be given in Section IV-A.

Collect all the channel gains in the network into a vector

$$\mathbf{h} := [h_{i,j} \mid i,j \in \{1,2,\dots,N\}, i<j]$$

and denote the channels 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 $\mathcal{N}(\mathbf{o}, \mathbf{\Sigma})$.

We can now write the channel gains $\mathbf{h}$ as

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

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

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

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

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

In the next section, we discuss how information about the channels can be obtained in order to update the statistics of the channel gains and to estimate them.

III. Model for Channel Training

Channel estimation of wireless links is typically performed by transmitting a known sequence of data over the channels that are to be estimated and make observations in order to obtain information about the channels. This technique is known as *channel training* or *channel sounding* and the known data sequence is called *training sequence*.

In order to estimate the channel gain of each point to point wireless link in the network, we transmit from each node a training sequence $u_1,\dots,u_M$ and collect the observations $r_1,\dots,r_M$ at the receiver nodes. Suppose that the channel gain of the point to point link is h, then the observations can be modeled as

$$r_m = h u_m + v_m$$

where $m \in \{1,\dots,M\}$ and v_m is noise which is assumed to be independent over the time and to be Gaussian distributed.

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. Each sleep cycle time instant k is further divided into more 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. 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.

Suppose that nodes i and j are awake at the sleep cycle instant k and node i transmits its training sequence. Then the observations at the node j can be written as

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

for all $i,j \in \mathcal{S}(k)$ and $m \in \{1,\dots,M\}$, where M is the length of the training sequence. Note that the observations collected at the nodes are independent given the channel gains.

If we collect all the observations at the sleep cycle instant k into a vector $\mathbf{r}_k$

$$\mathbf{r}_k := [r_{k,i,j,m} \mid i,j \in \mathcal{S}(k), m \in \{1,\dots,M\}, i \neq j]$$

then $\mathbf{r}_k$ is independent over sleep cycle instants k given the channel gain vector $\mathbf{h}$. Therefore we can write the joint probability distribution of $\mathbf{r}$ and $\mathbf{h}$ as

$$p_{\mathbf{r},\mathbf{h}} = p_{\mathbf{h}} \prod_{k=1}^{K} p_{\mathbf{r}_k|\mathbf{h}}$$

where $\mathbf{r} := [\mathbf{r}_k \mid k \in \{1,\dots,K\}]$.

Each node $i \in \{1,2,\dots,N\}$ in the network has an estimate $\mathbf{h}_i$of the channel gain vector $\mathbf{h}$. Since we want the same a posteriori distribution $p_{\mathbf{h}|\mathbf{r}}$ at each node at the end of the channel estimation, we write the joint distribution as

$$p_{\mathbf{r},\mathbf{h},\mathbf{h}_1,\dots,\mathbf{h}_N} = \prod_{k=1}^{K} p_{\mathbf{r}_k|\mathbf{h}} \prod_{i=1}^{N} \delta(\mathbf{h} - \mathbf{h}_i)(p_{\mathbf{h}}(\mathbf{h}_i))^{\frac{1}{N}} \qquad (1)$$

where δ is the point mass distribution at zero.

We can associate a bipartite graph called a factor graph [7] with the model which we discussed above, as in Fig. 1. Let us represent the nodes $s_1,\dots,s_N$ with the left side nodes (variable nodes) of the factor graph and the time instants $1,2,\dots,K$ 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 $s_i \in \mathcal{S}(k)$, which is awake during the sleep cycle instant k, with kth left node in the factor graph which corresponds to the kth sleep cycle instant.

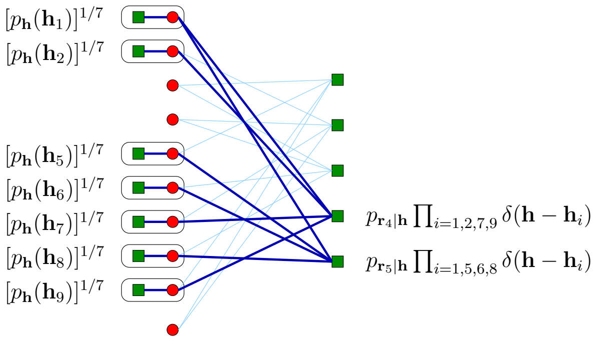

$[p_{\mathbf{h}}(\mathbf{h}_1)]^{1/7}$
$[p_{\mathbf{h}}(\mathbf{h}_2)]^{1/7}$
$[p_{\mathbf{h}}(\mathbf{h}_5)]^{1/7}$
$[p_{\mathbf{h}}(\mathbf{h}_6)]^{1/7}$
$[p_{\mathbf{h}}(\mathbf{h}_7)]^{1/7}$
$[p_{\mathbf{h}}(\mathbf{h}_8)]^{1/7}$
$[p_{\mathbf{h}}(\mathbf{h}_9)]^{1/7}$
$p_{\mathbf{r}_4|\mathbf{h}} \prod_{i=1,2,7,9} \delta(\mathbf{h}-\mathbf{h}_i)$
$p_{\mathbf{r}_5|\mathbf{h}} \prod_{i=1,5,6,8} \delta(\mathbf{h}-\mathbf{h}_i)$

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

Because of 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 messages (messages are encoded because they are to be sent over the noisy channels) and to use them in calculating the outgoing messages. Therefore, after l complete sleep cycles a sensor node s_i will have opportunity to obtain information from only those nodes that are no more than $2l$ edges away from s_i on the factor graph. Thus, after l complete sleep cycles the estimate for $\mathbf{h}_i$ can only directly exploit observations at the nodes that are no more than $2l$ edges away from s_i. This enables us to write a joint distribution indicating the information available to node s_i as

$$p^{l}_{\mathbf{r}(\mathcal{T}(i,j)),\mathbf{h},\mathbf{h}(\mathcal{P}(i,l))} = \prod_{k\in\mathcal{T}(i,j)} p_{\mathbf{r}_k|\mathbf{h}} \prod_{j\in\mathcal{P}(i,l)} \delta(\mathbf{h}-\mathbf{h}_j)(p_{\mathbf{h}}(\mathbf{h}_j))^{\frac{1}{g(l)}} \quad (2)$$

where $g(l) = c(c-1)^l(d-1)^l$ is the number of sensor nodes no more than $2l$ edges away from sensor node s_i on the factor graph after l iterations, and $\mathcal{T}(i,j)$ and $\mathcal{P}(i,l)$ are the sleep cycle instants k' and node indices j no more than $2l$ edges away from sensor node s_i in the factor graph respectively. Also

$$\begin{aligned}\mathbf{h}(\mathcal{P}(i,l)) &:= \{\mathbf{h}_j | j \in \mathcal{P}(i,l)\}\\ \mathbf{r}(\mathcal{T}(i,j)) &:= \{\mathbf{r}_k | k \in \mathcal{T}(i,j)\}\end{aligned}$$

We presently show how EP and LMS can be applied for the channel gain estimation.

IV. Algorithms for Distributed Estimation

There are a number of algorithms, such as belief propagation (BP), expectation propagation (EP) and least-mean square (LMS), which may be used for distributed estimation in sensor networks. In this paper we compare the performance of EP and LMS in estimating the channel gains of the sensor nodes. We first show how EP can be used for distributed channel estimation when the random sleep strategy is used during the initialization phase of the sensor network.

A. Expectation Propagation

Suppose that the mean and covariance of the channel gains are $\mathbf{m}_{\mathbf{h}}$ and $\mathbf{\Sigma}_{\mathbf{h}}$, respectively. Then the approximated prior joint distribution on $\mathbf{h}$ can be written as

$$p_{\mathbf{h}}(\mathbf{h}) \propto \exp\{-\frac{1}{2}[(\mathbf{h}-\mathbf{m}_{\mathbf{h}})^T\mathbf{\Sigma}_{\mathbf{h}}^{-1}(\mathbf{h}-\mathbf{m}_{\mathbf{h}})]\} \quad (3)$$

Also the conditional joint distribution on the observations $\mathbf{r}_{k,i}$ collected during sleep cycle instant k at node $i \in \mathcal{S}(k)$ can be written as

$$p_{\mathbf{r}_{k,i}|\mathbf{h}}(\mathbf{r}_{k,i}|\mathbf{h}) \propto \exp\{-\frac{1}{2}[(\mathbf{r}_{k,i}-\mathbf{m}_{\mathbf{r}k,i})^T\mathbf{\Sigma}_{\mathbf{r}_{k,i}}^{-1}(\mathbf{r}_{k,i}-\mathbf{m}_{\mathbf{r}k,i})]\} \quad (4)$$

where

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

where σ_N^2 is noise variance.

Each node in the network, knowing the prior statistics on the channel gains, has an initial estimate of channel gains $\hat{\mathbf{h}} = \mathbf{m}_{\mathbf{h}}$. They may want to 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 $\mathbf{h}$ with a factor graph as shown in Fig. 1, we can apply EP [6] [3] to calculate posterior distribution of $\mathbf{h}$. EP iteratively passes messages on the edges of the factor graph in order to calculate the posterior distribution. We have to first select a message family [3] in order to apply EP on the factor graph. Considering the prior distribution $p_{\mathbf{h}}(\mathbf{h})$ and the conditional joint distributions $p_{\mathbf{r}_{k,i}|\mathbf{h}}(\mathbf{r}_{k,i}|\mathbf{h})$, we select the the message exponential family to be used in EP to be multivariate Gaussian distributed with sufficient statistics

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

where

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

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

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

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

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

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

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

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

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

where

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

where

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

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 $\{p_{\boldsymbol{\theta}_i|\mathbf{h}}|i \in \{1,\ldots,L\}\}$, in which each distribution takes the form

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

Then, the product of the distributions can be written as

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

We presently explain how the nodes apply EP to calculate the posterior distribution. 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 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 messages are from the same exponential family, when the messages are multiplied the parameters of the messages sum up. Thus, it is sufficient for each node to pass only the parameters of the messages. 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 $\rho_{i\to k}^{(p)}$.

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

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 $\boldsymbol{\lambda}_{k\to i}^{(p)}$.

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

At the final iteration, node i calculates the parameters of the posterior distribution by summing $\boldsymbol{\lambda}_{k\to i}^{(p)}$ from k in all c sleep cycle instants it was awake, stacking it with the parameter vectors calculated using the information available at the node, and adding it to $\boldsymbol{\tau}$. Each node calculates its estimate by finding the mean of the posterior distribution.

B. Diffusion LMS Algorithm

The least-mean square algorithm is an adaptive filtering technique which can be used for the distributed estimation of the channel gains. We assume that each node has a copy of the channel gain vector $\mathbf{h}$ and it takes an initial value of $\mathbf{m}_{\mathbf{h}}$.

Suppose that 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 diffusion LMS algorithm [8] to estimate $h_{i,i'}$. Let us denote the estimate of $h_{i,i'}$ at time instant m of sleep cycle instant k by $\hat{h}_{i,i'}^{k,m}$. Then,

$$\hat{h}_{i,i'}^{k,m} = \hat{h}_{i,i'}^{k,m-1} + \mu u_{i,m}(r_{k,i,i',m} - \hat{h}_{i,i'}^{k,m-1}u_{i,m}) \qquad (5)$$

where μ is step size.

We assume that at the end of each sleep cycle instant k, the nodes awake at k diffuse their estimates by

$$\tilde{\mathbf{h}}^k = \sum_{i\in\mathcal{S}(k)} a(k,i)\hat{\mathbf{h}}_i^k \qquad (6)$$

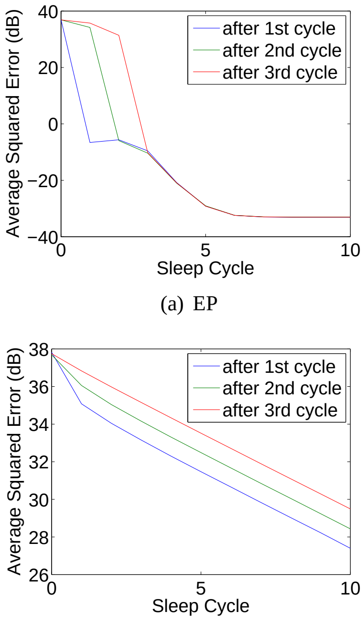
where $\hat{\mathbf{h}}_i^k$ is the estimate of $\mathbf{h}$ at node i at the end of the sleep cycle instant k and $a(k,i)$ satisfy $\sum_{i\in\mathcal{S}(k)} a(k,i) = 1$. The nodes $i \in \mathcal{S}(k)$ use the combined estimate $\tilde{\mathbf{h}}^k$ for estimation during the later sleep cycle instants.

V. Simulation Results

We have done some simulations and have plotted the estimation errors for both algorithms. In this section we discuss our simulation results.

First we generate candidate sensor positions on the plane $\mathbb{R}^2$ such that they are i.i.d. and Gaussian distributed with zero mean and variance 1. These candidate sensor positions are the 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. Then, we generate the channel gains. We calculate the statistics (mean and covariance matrix) of the channel gains using many sensor positions generated in this manner, then generate a new set of sensor positions to test the algorithms with. Next, we generate the training sequences of length 1000 randomly and uniformly with values of either 1 or -1. We run 400 Monte Carlo simulations for each algorithm for the experiment presented below.

In this experiment, we estimate the channel gain vector $\mathbf{h}$ for a network with 20 sensors applying EP and LMS. We apply a random sleep strategy with $K = 10$ and $d = 4$ to the network.



(a) EP

(b) LMS

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

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. It is obvious that 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, and thus, including the estimation errors of those channels unobserved gives large average estimation errors. For these reasons, we plot the average estimation errors of only those channel gains which can be observed directly or indirectly by the nodes after first, second and third sleep cycles.

Fig. 2(a) shows the average squared estimation error of the observed channel gains in dB when EP is applied. Note that there is a drastic change in the estimation error after first sleep cycle for the channel gains which were observed directly by the nodes. Also note that the drastic change shifts to the second sleep cycle and third sleep cycle for the channel gains observed directly or indirectly after second sleep cycle and third sleep cycle, respectively, as is to be expected. Another interesting thing that can be observed in the plot is that the estimation errors decrease even after the the drastic change, which is 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. 2(b) shows average squared error of the estimated channel gains when diffusion LMS is used with step size $\mu = 1.995$. 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. The message passing overhead of the two algorithms is exactly the same at each sleep cycle time instant: they both exchange messages of equal dimension to the number of channel gains between each pair of nodes that are awake at a particular sleep cycle time instant. The computational complexity of the two algorithms is also comparable. EP requires no more than $(d(c+1)+d(d+1))\frac{N(N-1)}{2}+d^2M$ additions and d^2M multiplications per sleep cycle time instant and the solution of a $\frac{N(N-1)}{2}$ dimensional linear system at each sensor node in the network during the final iteration. Diffusion LMS requires $d^2\frac{N(N-1)}{2}+2d(d-1)M$ multiplications and $d(d-1)\left(\frac{N(N-1)}{2}+2M\right)$ additions per sleep cycle time instant iteration with no additional computation necessary at the final iteration.

VI. Conclusion

EP converges to the exact a posteriori distribution provided that there has no loop occurred during the message passing. Since the probability of encountering loops on the factor graph for finite number of iteration → 0 as the number of nodes in the network → ∞ [3], for finite number of iterations ℓ EP calculates the exact a posteriori distribution given those observations no more that ℓ edges away in the sleep cycle (factor) graph. The mean of the a posteriori distribution gives us the estimate of the channel coefficients at the end of the random sleep cycles, which is the best estimate in the mean squared error sense. We have extended these theoretical results established in [3] by simulations showing that EP achieves better performance with faster convergence than the diffusion LMS algorithm in the distributed channel estimation. The computational complexity of the two methods for this problem were also compared.

References

[1] 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, 2006.

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

[3] 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, To appear.

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

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

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

[7] 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, 2001.

[8] C. G. Lopes and A. H. Sayed, "Diffusion least-mean-squares over adaptive networks," in *Proc. IEEE International Conference on Accoustics, Speech and Signal Processing*, 2007.