ON THE PERFORMANCE OF AVERAGE CONSENSUS IN MOBILE WIRELESS SENSOR NETWORKS

Valentin Schwarz and Gerald Matz

Institute of Telecommunications, Vienna University of Technology, Vienna, Austria
Email: {vschwarz,gmatz}@nt.tuwien.ac.at

ABSTRACT

Distributed averaging in wireless sensor networks is fundamental for advanced distributed algorithms. In this paper, we study the impact of node mobility on the well-known average consensus algorithm. More specifically, we study the MSE in mobile wireless sensor networks modeled via evolving random geometric graphs. We provide a closed-form lower bound for the MSE that shows that node mobility is beneficial. Numerical results involving various mobility models demonstrate that mobility can be traded against convergence time, transmit power, or node density.

Index Terms— Average consensus, wireless sensor networks, distributed inference

1. INTRODUCTION

Wireless sensor networks (WSN) have numerous potential applications, e.g., environmental engineering or target tracking. In many WSN, the energy supply and computing power of the sensor nodes is very limited. Thus, the design of energy efficient and fault tolerant distributed algorithms in WSN has become an important research topic over the last decade. For many of these algorithms, distributed averaging is a basic building block. Well-known distributed averaging schemes are *average consensus (AC)* [1, 2], gossip (pairwise averaging) schemes [3], and *consensus propagation* [4]. A comparative discussion of AC and gossip algorithms is provided in [5]. Consensus propagation has a structure different from AC and gossiping since it is based on Gaussian belief propagation [6]. AC is particularly popular since its linear structure renders its analysis simple. However, the implementation of AC is hampered by the requirement of synchronous operation.

In this paper, we investigate the performance of AC in mobile WSN, i.e., WSN in which (some of) the nodes are moving. To this end, we extend the mean-square error (MSE) analysis in [7] to random geometric graphs [8] with motion models. Analytical bounds on the MSE performance of AC in mobile WSN and numerical simulations reveal that node mobility is beneficial in the sense that AC convergence is improved (equivalently, the nodes' transmit power or the number of nodes can be reduced).

In related work, the effect of node mobility on gossip algorithms was studied analytically in [9] and motion dynamics in WSN is considered in the context of diffusion algorithms in [10]. With regard to AC, static WSN with dynamically switched links have been investigated in [11]. Furthermore, the use of an external velocity (advection) field to design improved AC weights was considered in [12, 13] (the sensor nodes are fixed, though).

Funded by WWTF Grant ICT-044 and FWF Grant S10606.

2. PRELIMINARIES

WSN can be efficiently modeled in terms of graphs that capture the properties (e.g., connectivity) of the networks under consideration. The behavior of WSN algorithms can then be analyzed using tools from graph theory. A graph is an ordered pair $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, consisting of a node set $\mathcal{V}$ of cardinality $I = |\mathcal{V}|$ and an edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. The nodes represent sensors and the edges represent communication links between the sensor nodes. Due to the reciprocity of wireless channels we consider undirected graphs. We denote by $\mathcal{N}(i)$ the set of neighbors of node i, i.e., $\mathcal{N}(i) = \{j : (i,j) \in \mathcal{E}\}$.

Since the sensor nodes have a limited communication range, the graph topology is determined by the node positions. This property can be captured with random geometric graphs [8], which will be considered in the following. We assume that the sensor nodes are randomly distributed in a square region and connected whenever their Euclidian distance equals at most the given communication range, denoted r. To simplify the analysis we circumvent boundary effects by extending the square region periodically such that the WSN geometry $\mathcal{A}$ effectively becomes a torus.

The task of the sensor network is to determine in a distributed fashion the average of the sensor measurements s_i, $i = 1, \ldots, I$. The average $\bar{s} = \frac{1}{I}\sum_i s_i$ shall be computed using AC. Hence, the measurements are used as the first initial states, i.e., $x_i[0] = s_i$, and then the following state update is applied at each node:

$$x_i[k+1] = w_{ii}\, x_i[k] \,+ \sum_{j \in \mathcal{N}(i)} w_{ij}\, x_j[k]\,,$$

where k is the iteration number and w_{ij} denotes edge weights which are considered in more detail later. The AC updates can be rewritten as

$$\boldsymbol{x}[k+1] = \boldsymbol{W}\boldsymbol{x}[k]\,, \tag{1}$$

with the state vector $\boldsymbol{x}[k] = \big(x_1[k]\, x_2[k]\, \ldots\, x_I[k]\big)^T$ and the weight matrix

$$(\boldsymbol{W})_{ij} = \begin{cases} w_{ij}, & (i,j) \in \mathcal{E} \text{ or } i = j, \\ 0, & \text{else.} \end{cases}$$

It is seen that the weight matrix has the same zero pattern as the adjacency matrix of the graph $\mathcal{G}$. Convergence of (1) in the sense that $\lim_{k\to\infty} \boldsymbol{x}[k] = \bar{s}\mathbf{1}$ is ensured if the weight matrix satisfies $\boldsymbol{W}\mathbf{1} = \mathbf{1}$ and $\rho(\boldsymbol{W} - \boldsymbol{J}) < 1$; here, $\mathbf{1}$ is the all-ones vector, $\rho(\cdot)$ denotes the spectral radius, and $\boldsymbol{J} \triangleq \frac{1}{I}\mathbf{1}\mathbf{1}^T$ is the orthogonal projection matrix on the span of $\mathbf{1}$. Note that the condition $\boldsymbol{W}\mathbf{1} = \mathbf{1}$ implies $w_{ii} = 1 - \sum_{j\in\mathcal{N}(i)} w_{ij}$.

The design of the weight matrix $\boldsymbol{W}$ was the subject of considerable research. A weight design that uses tools from convex optimization to maximize the asymptotic convergence rate of AC was

proposed in [14]. A distributed computation of these weights is not straightforward, however, and their transient performance is sometimes poor. By contrast, Metropolis-Hastings (MH) weights, which have their origin in Markov chain theory [15], can easily be implemented in a distributed manner according to

$$w_{ij}^{\text{MH}} = \begin{cases} \frac{1}{\max\{d_i,d_j\}+1}\,, & \text{for } (i,j)\in\mathcal{E}, \\ 1-\sum_{j'\in\mathcal{N}(i)} w_{ij'}^{\text{MH}}\,, & \text{for } i=j, \\ 0\,, & \text{else.} \end{cases}$$

Here, d_i denotes the degree (the number of neighbors) of node i. Finally, we consider the constant weight (CW) design where all weights are equal, $w_{ij}=w$, $(i,j)\in\mathcal{E}$. The constant weight w can be designed to optimize the convergence speed (cf. [14]), but we here use the simple conservative choice $w<\frac{1}{\max_i\{d_i\}}$.

The designs described above involve a symmetric time-invariant weight matrix. Time-varying weights have the potential to achieve improved performance, e.g., [7, 16]. In [12, 13], asymmetric weight matrices are designed that mimick advection-diffusion processes in order to increase the convergence speed. The additional advection component acts like a blender that improves the mixing of the measurements by enforcing the states to "move" through the WSN. It is therefore reasonable to expect that moving sensors result in a similar performance improvement.

3. AC IN MOBILE WSN

We next consider mobile WSN in which (some of) the sensor nodes are moving. In general, node mobility will result in a time-varying graph topology, i.e., the edge set and hence the node neighbors and degrees changes over time. We model the mobile WSN in terms of a stationary Markovian evolving graph [17] for which the node set V remains unchanged but the edge set changes over a time, thus forming a sequence of graph topologies $\mathcal{T}=\{\mathcal{E}_1,\ldots,\mathcal{E}_k\}$ that constitutes a stationary Markov chain. We denote the set of neighbors of node i at time k by $\mathcal{N}_k(i)=\{j:(i,j)\in\mathcal{E}_k\}$.

The node mobility entails time-varying AC weights $\boldsymbol{W}_k$. Note that in contrast to time-varying AC weights in static WSN, the weight matrices $\boldsymbol{W}_k$ here can even have different zero patterns since the adjacency matrix of the underlying graph has changed. The AC update equation thus reads

$$\boldsymbol{x}[k+1]=\boldsymbol{W}_{k+1}\,\boldsymbol{x}[k]\,. \tag{2}$$

Combining all state updates since the initial measurements yields (here $\boldsymbol{s}=(s_1\ s_2\ \ldots\ s_I)^T$)

$$\boldsymbol{x}[k]=\boldsymbol{W}_{k\to 1}\boldsymbol{s}, \qquad \text{with } \boldsymbol{W}_{k\to l}=\boldsymbol{W}_k\boldsymbol{W}_{k-1}\cdots\boldsymbol{W}_l.$$

In order for (2) to converge to $\bar{s}\mathbf{1}=\boldsymbol{J}\boldsymbol{s}$, it is sufficient that each individual weight matrix $\boldsymbol{W}_k$ satisfies the convergence conditions of AC, i.e., $\boldsymbol{W}_k\mathbf{1}=\mathbf{1}$ and $\rho(\boldsymbol{W}_k-\boldsymbol{J})<1$. We will stick to this sufficient condition, since it allows us to use the MH or CW approach in each time instant to design the weights $\big(\boldsymbol{W}_k\big)_{ij}$ based on the current node degrees and thereby guarantee convergence. Note that such a design only requires to locally track the node degrees and hence can easily be implemented in a distributed manner. In the following, we aim at quantifying the performance gains achieved by node mobility.

3.1. Main Result

As a performance metric, we use the per-node MSE in iteration k, defined as

$$\bar{\epsilon}^2[k]\triangleq\frac{1}{I}\,\mathrm{E}_{\mathcal{T}}\{\epsilon^2[k]\} \quad \text{with } \epsilon^2[k]\triangleq\mathrm{E}_{\boldsymbol{s}|\mathcal{T}}\big\{\|\boldsymbol{x}[k]-\bar{s}\mathbf{1}\|^2\big\}. \tag{3}$$

Here, $\mathrm{E}_{\mathcal{T}}$ and $\mathrm{E}_{\boldsymbol{s}|\mathcal{T}}$ denote expectation with respect to the sequence $\mathcal{T}$ of graph topologies (which determines the time-varying AC weights) and conditional expectation with respect to the measurements $\boldsymbol{s}$ given $\mathcal{T}$. We define $\omega_k=\|\boldsymbol{W}_{2k}\boldsymbol{W}_{2k-1}\|_F^2$ and $P_{\bar{s}}=\mathrm{E}_{\boldsymbol{s}|\mathcal{T}}\{\bar{s}^2\}=\frac{1}{I^2}\mathbf{1}^T\boldsymbol{R}_s\mathbf{1}$ where $\boldsymbol{R}_s=\mathrm{E}_{\boldsymbol{s}|\mathcal{T}}\{\boldsymbol{s}\boldsymbol{s}^T\}$ denotes the measurement correlation matrix.

Theorem 1. *Consider AC on a stationary Markovian evolving graph and assume that $P_{\bar{s}}$ does not depend on $\mathcal{T}$, that $\boldsymbol{R}_s$ has full rank, and that $\mathrm{E}_{\mathcal{T}}\{\omega_k\omega_{k-1}\}\geq\mathrm{E}_{\mathcal{T}}\{\omega_k\}\,\mathrm{E}_{\mathcal{T}}\{\omega_{k-1}\}$. Then, the MSE is lower bounded as*

$$\bar{\epsilon}^2[k]\geq(I-1)P_{\bar{s}}\left[\frac{\mathrm{E}_{\mathcal{T}}\{\omega_1\}-1}{I-1}\right]^{\lceil\frac{k}{2}\rceil}. \tag{4}$$

Discussion. The assumption that $P_{\bar{s}}$ is independent of $\mathcal{T}$ effectively implies an i.i.d. sensor node placement. Furthermore, the condition $\mathrm{E}_{\mathcal{T}}\{\omega_k\omega_{k-1}\}\geq\mathrm{E}_{\mathcal{T}}\{\omega_k\}\,\mathrm{E}_{\mathcal{T}}\{\omega_{k-1}\}$ essentially excludes evolving graphs that oscillate between being strongly and weakly connected. It is seen that the MSE lower bound decays exponentially, with an MSE improvement by a factor of $\frac{\mathrm{E}_{\mathcal{T}}\{\omega_1\}-1}{I-1}$ every second iteration. Note that $\mathrm{E}_{\mathcal{T}}\{\omega_1\}$ describes how on average the connectivity of the evolving graph changes from one time instant to the next (see Section 3.2).

Proof. We develop the MSE by noting that $\bar{s}\mathbf{1}=\boldsymbol{J}\boldsymbol{s}$ and hence

$$\epsilon^2[k]=\mathrm{E}_{\boldsymbol{s}|\mathcal{T}}\big\{\|(\boldsymbol{W}_{k\to 1}-\boldsymbol{J})\boldsymbol{s}\|^2\big\}=\mathrm{tr}\big\{\overline{\boldsymbol{W}}_{k\to 1}\boldsymbol{R}_s\overline{\boldsymbol{W}}_{k\to 1}^T\big\}, \tag{5}$$

where $\overline{\boldsymbol{W}}_{k\to l}=\boldsymbol{W}_{k\to l}-\boldsymbol{J}$. Using the fact that $\boldsymbol{W}_{k\to l}\boldsymbol{J}=\boldsymbol{J}$ and $\boldsymbol{R}_{\boldsymbol{x}[l-1]}=\boldsymbol{W}_{(l-1)\to 1}\boldsymbol{R}_s\boldsymbol{W}_{(l-1)\to 1}^T$, it can be shown that $\overline{\boldsymbol{W}}_{k\to 1}=\overline{\boldsymbol{W}}_{k\to l}\overline{\boldsymbol{W}}_{(l-1)\to 1}$ and $\overline{\boldsymbol{W}}_{(l-1)\to 1}\boldsymbol{R}_s\overline{\boldsymbol{W}}_{(l-1)\to 1}^T=\overline{\boldsymbol{R}}_{\boldsymbol{x}[l-1]}\triangleq(\boldsymbol{I}-\boldsymbol{J})\boldsymbol{R}_{\boldsymbol{x}[l-1]}(\boldsymbol{I}-\boldsymbol{J})$. We therefore obtain

$$\epsilon^2[k]=\mathrm{tr}\left\{\overline{\boldsymbol{W}}_{k\to l}\overline{\boldsymbol{R}}_{x[l-1]}\overline{\boldsymbol{W}}_{k\to l}^T\right\}$$

Exploiting the fact that the null space of both $\overline{\boldsymbol{W}}_{k\to l}$ and $\overline{\boldsymbol{R}}_{x[l-1]}$ is spanned by $\mathbf{1}$ and denoting the smallest non-zero eigenvalue of $\overline{\boldsymbol{R}}_{x[l-1]}$ by $\lambda_{\min}$, it can be shown that

$$\mathrm{tr}\left\{\overline{\boldsymbol{W}}_{k\to l}\overline{\boldsymbol{R}}_{x[l-1]}\overline{\boldsymbol{W}}_{k\to l}^T\right\}\geq\lambda_{\min}\left\|\overline{\boldsymbol{W}}_{k\to l}\right\|_F^2. \tag{6}$$

This bound is tight for the case where $\overline{\boldsymbol{R}}_{x[l-1]}=\lambda_{\min}(\boldsymbol{I}-\boldsymbol{J})$. With the constraint that the trace of $\boldsymbol{R}_{x[l-1]}$ is fixed, it follows that

$$\begin{aligned}\mathrm{tr}\{\boldsymbol{R}_{x[l-1]}\}&=\mathrm{tr}\{\overline{\boldsymbol{R}}_{x[l-1]}\}+\mathrm{tr}\{\boldsymbol{J}\boldsymbol{R}_{x[l-1]}\boldsymbol{J}\}\\&=\lambda_{\min}(I-1)+\mathrm{tr}\{\boldsymbol{J}\boldsymbol{R}_s\boldsymbol{J}\},\end{aligned}$$

which further implies

$$\lambda_{\min}=\frac{\mathrm{tr}\{\boldsymbol{R}_{x[l-1]}-\boldsymbol{J}\boldsymbol{R}_s\boldsymbol{J}\}}{I-1}=\frac{\mathrm{tr}\{\overline{\boldsymbol{W}}_{(l-1)\to 1}\boldsymbol{R}_s\overline{\boldsymbol{W}}_{(l-1)\to 1}\}}{I-1}.$$

Inserting this result into (6) and comparing with (5), we obtain

$$\epsilon^2[k]\geq\epsilon^2[l-1]\,\frac{\left\|\boldsymbol{W}_{k\to l}\right\|_F^2-1}{I-1}, \tag{7}$$

where we further used $\left\|\overline{\boldsymbol{W}}_{k\to l}\right\|_F^2=\left\|\boldsymbol{W}_{k\to l}\right\|_F^2-1$. Assuming k even, applying this bound recursively with $l=k-1$, $l=k-3$, etc, and using $\epsilon^2[0]\geq P_{\bar{s}}I(I-1)$ with $P_{\bar{s}}=\mathrm{E}\{\bar{s}^2\}=\frac{1}{I^2}\mathbf{1}^T\boldsymbol{R}_s\mathbf{1}$, we have

$$\epsilon^2[k]\geq I(I-1)P_{\bar{s}}\prod_{l=1}^{k/2}\frac{\omega_l-1}{I-1}. \tag{8}$$

Note that the iterated application of (7) implies that the bound (8) becomes less and less tight as k increases. For odd iteration indices $k = 2l-1$, we simply use the fact that $\epsilon^2[2l-1] \geq \epsilon^2[2l]$. Note that the combination of two iterations is important to capture mobility effects. The final result (4) is obtained by inserting (8) into (3), using the independence of $P_{\bar{s}}$ and $\mathcal{T}$, and noting that

$$\mathrm{E}_{\mathcal{T}}\left\{\prod_{l=1}^{k/2}\frac{\omega_l-1}{I-1}\right\} \geq \prod_{l=1}^{k/2}\mathrm{E}_{\mathcal{T}}\left\{\frac{\omega_l-1}{I-1}\right\} = \left[\frac{\mathrm{E}_{\mathcal{T}}\{\omega_1\}-1}{I-1}\right]^{\frac{k}{2}}.$$

Here, we used the assumption $\mathrm{E}_{\mathcal{T}}\{\omega_k\omega_{k-1}\} \geq \mathrm{E}_{\mathcal{T}}\{\omega_k\}\,\mathrm{E}_{\mathcal{T}}\{\omega_{k-1}\}$ and exploited the fact that for stationary Markovian evolving graphs the sequence ω_l is stationary Markovian, too. □

3.2. Mean Frobenius Norm

The key quantity in (4) is the mean Frobenius norm of $\boldsymbol{G} \triangleq \boldsymbol{W}_2\boldsymbol{W}_1$,

$$\mathrm{E}_{\mathcal{T}}\{\omega_1\} = \sum_{i,j}\mathrm{E}_{\mathcal{T}}\{g_{ij}^2\}, \tag{9}$$

where

$$g_{ij} = (\boldsymbol{G})_{ij} = \sum_{l\in\tilde{\mathcal{N}}_2(i)\cap\tilde{\mathcal{N}}_1(j)}(\boldsymbol{W}_2)_{il}(\boldsymbol{W}_1)_{lj}.$$

Here, $\tilde{\mathcal{N}}_k(i) = \{i\} \cup \mathcal{N}_k(i)$. Hence, g_{ij} involves node j at time $k=1$, node i at time $k=2$, and the nodes that are neighbors of node j at time $k=1$ and of node i at time $k=2$. Note that it is possible that $g_{ij} \neq 0$ even if there is no path between nodes i and j at any given time. This is in contrast to the static case and explains in part why node mobility can be beneficial for distributed averaging. We assume that not necessarily all nodes are moving; the set of moving nodes is defined as $\mathcal{V}_{\mathrm{m}} \subseteq \mathcal{V}$ and the number of moving nodes is denoted by $I_{\mathrm{m}} = |\mathcal{V}_{\mathrm{m}}|$. We can then distinguish four disjoint types of expectations: (i) $\mathrm{E}_{\mathcal{T}}\{|g_{ii}|^2\}$ if node i is moving, (ii) $\mathrm{E}_{\mathcal{T}}\{|g_{ii}|^2\}$ if node i is not moving, (iii) $\mathrm{E}_{\mathcal{T}}\{|g_{ij}|^2\}$ if node i or j is moving and (iv) $\mathrm{E}_{\mathcal{T}}\{|g_{ij}|^2\}$ if neither node i nor node j is moving. Under the assumption of a spatially homogenous node placement, these expectations are independent of the node indices and hence we can collect equivalent terms to express the expectated Frobenius norm of $\boldsymbol{G}$ as,

$$\begin{aligned}\mathrm{E}_{\mathcal{T}}\{\omega_1\} = {} & (I-I_{\mathrm{m}})\,\mathrm{E}_{\mathcal{T}}\{g_{ii}^2; i\notin\mathcal{V}_{\mathrm{m}}\} + I_{\mathrm{m}}\,\mathrm{E}_{\mathcal{T}}\{g_{ii}^2; i\in\mathcal{V}_{\mathrm{m}}\} \\ & + ((I-1)I - I_{\mathrm{m}}(2I-I_{\mathrm{m}}-1))\,\mathrm{E}_{\mathcal{T}}\{g_{ij}^2; i,j\notin\mathcal{V}_{\mathrm{m}}\} \\ & + I_{\mathrm{m}}(2I-I_{\mathrm{m}}-1)\,\mathrm{E}_{\mathcal{T}}\{g_{ij}^2; i\in\mathcal{V}_{\mathrm{m}} \text{ or } j\in\mathcal{V}_{\mathrm{m}}\}.\end{aligned}$$

Up to now, our development is independent of a particular network and motion model. We next consider specific models that are simple enough to obtain closed-form expressions for $\mathrm{E}_{\mathcal{T}}\{\omega_1\}$.

3.3. Static WSN

In what follows, we restrict our attention to the case of random geometric graphs on the surface of a torus $\mathcal{A}$ (equivalent to a periodically extended rectangular region). The I nodes initially have an i.i.d. uniform distribution. We first consider the static scenario (i.e., $\boldsymbol{W}_1 = \boldsymbol{W}_2$), which serves as a basis for the case with node mobility. Here,

$$\mathrm{E}_{\mathcal{T}}\{\omega_1\} = I\,\mathrm{E}_{\mathcal{T}}\{g_{ii}^2\} + (I-1)I\,\mathrm{E}_{\mathcal{T}}\{g_{ij}^2; i\neq j\}.$$

For a given weight design, the computation of $\mathrm{E}_{\mathcal{T}}\{g_{ij}^2\}$ requires to determine three basic probabilities that characterize the geometric relations between the nodes. More specifically, there is the pairwise neighborhood probability

$$p = \mathrm{P}\{j\in\mathcal{N}(i)\} = \frac{r^2\pi}{|\mathcal{A}|},$$

which equals the ratio of the area within which a node i can communicate and the overall WSN area. Furthermore, we need the triple-neighborhood probability, i.e., the probability that three nodes are mutual neighbors. The underlying geometry is illustrated in Fig. 1. Here, nodes i and j have to be neighbors, i.e. $r_{ij} \leq r$ (where r_{ij} is the distance between nodes i and j); furthermore, node l has to lie in the intersection of the communication range of nodes i and j, denoted $\mathcal{A}_{ij}$, which happens with probability $\frac{|\mathcal{A}_{ij}|}{|\mathcal{A}|}$. The area of $\mathcal{A}_{ij}$ can be computed using standard techniques, leading to

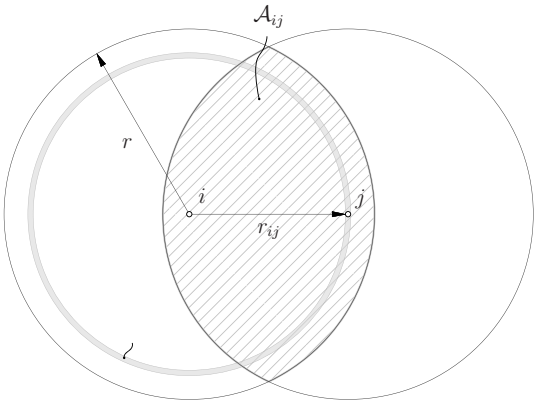

Fig. 1. *Geometry of triple neighborhoods.*

$$\mathrm{P}\{j\in\mathcal{N}(i),\, l\in\mathcal{N}(i)\cap\mathcal{N}(j)\} = p^2\left(1-\frac{3\sqrt{3}}{4\pi}\right). \tag{10}$$

Finally, we require the probability for the existence of a four-hop path between two nodes, which can be derived using similar considerations, leading to

$$\mathrm{P}\{l\in\mathcal{N}(i)\cap\mathcal{N}(j),\, m\in\mathcal{N}(i),\, n\in\mathcal{N}(j)\} = p^3\left(1-\frac{16}{3\pi^2}\right).$$

3.4. Mobile WSN

We next consider mobile WSN in which the movements of different nodes are statistically independent and stationary, thus leading to a stationary geometric Markovian evolving graph [17] in which the node positions remain i.i.d. over time. With regard to the computation of $\mathrm{E}_{\mathcal{T}}\{\omega_1\}$, the node mobility is captured by the following seven probabilities:

$$\begin{aligned}&\mathrm{P}\{j\in\mathcal{N}_1(i)\cap\mathcal{N}_2(i)\},\\ &\mathrm{P}\{l\in\mathcal{N}_1(i)\cap\mathcal{N}_2(j)\},\\ &\mathrm{P}\{j\in\mathcal{N}_1(i),\, l\in\mathcal{N}_1(i)\cap\mathcal{N}_2(j)\},\\ &\mathrm{P}\{j\in\mathcal{N}_1(i)\cap\mathcal{N}_1(l),\, l\in\mathcal{N}_2(i)\},\\ &\mathrm{P}\{j\in\mathcal{N}_1(i)\cap\mathcal{N}_2(i),\, l\in\mathcal{N}_1(i)\cap\mathcal{N}_2(i)\},\\ &\mathrm{P}\{j\in\mathcal{N}_1(i)\cap\mathcal{N}_2(i),\, l\in\mathcal{N}_1(i)\cap\mathcal{N}_2(j)\},\\ &\mathrm{P}\{i\in\mathcal{N}_1(l)\cap\mathcal{N}_1(n),\, j\in\mathcal{N}_2(l)\cap\mathcal{N}_2(n)\}.\end{aligned}$$

These probabilities depend on the number I_{m} of moving sensors and on the probabilities introduced for static networks in Section 3.3. Let

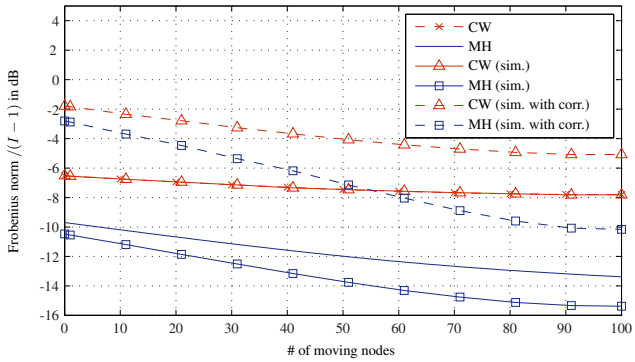

Fig. 2. *Frobenius norm versus the number of moving nodes for different weight designs and initial correlations.*

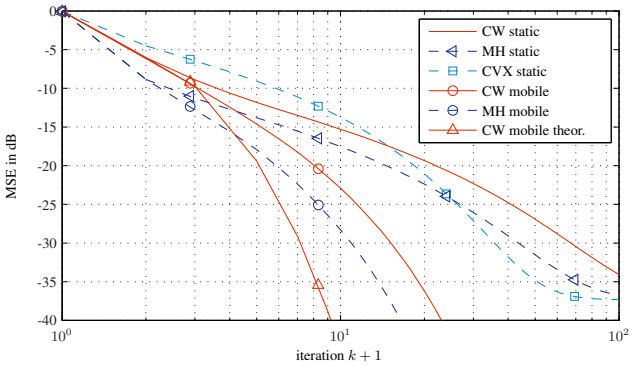

Fig. 3. *MSE behavior over the iteration number k for 20 nodes moving.*

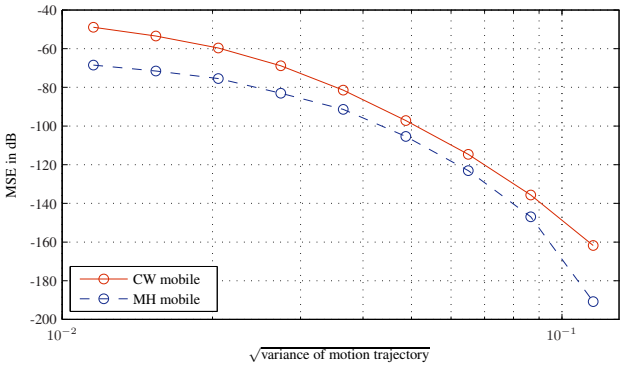

Fig. 4. *MSE versus the standard deviation of the distance traveled by the nodes.*

us compare the probabilities in line 3 and 4. If no node is moving, these expressions both simplify to (10); however, if node i is moving, the probability in line 3 remains the same but the probability in line 4 equals p^3.

As an example, we consider a motion model where the new positions of each of the I_m moving nodes are i.i.d. uniform. The weights for AC are constant (CW) (see Section 2); we denote by w the mean of the constant weight. For this model, the computation of the above probabilities is straightforward but cumbersome, i.e., the resulting expressions involve many terms. We therefore omit the full details due to lack of space and just illustrate the type of results obtained by specifying $\mathrm{E}_\mathcal{T}\{g_{ii}^2; i \notin \mathcal{V}_\mathrm{m}\}$. Specifically, it can be shown that

$$\mathrm{E}_\mathcal{T}\{g_{ii}^2; i \notin \mathcal{V}_\mathrm{m}\} = 1 + \pi \sum_{m,n} q_{mn} w^{\frac{m}{2}} (\sqrt{2}r)^m I^n.$$

The coefficients q_{mn} are shown in Table 1 (with $p_\mathrm{m} = \frac{I_\mathrm{m}-1}{I-1}$).

4. NUMERICAL RESULTS

Throughout this section we are using periodic (toroidal) random geometric graphs with $I = 100$ and $\mathcal{A} = [0,1] \times [0,1]$. $I_m \leq 100$ of these nodes move around randomly, with their positions in each time instant being i.i.d. uniformly distributed. Unless stated otherwise, the communication radius was $r = 0.16$ and the initial measurements were i.i.d. Gaussian with zero mean.

Fig. 2 shows analytical and numerical results for the mean Frobenius norm $\mathrm{E}_\mathcal{T}\{\omega_1\}$ obtained with the CW and MH weight design. For CW, we choose w equal to the reciprocal of the average maximum degree, estimated from 500 Monte Carlo runs. It is seen that for CW, analytical and numerical results match exactly, thus confirming the analysis from the previous Section. While this analysis does not apply directly to MH, using the CW result and replacing w with the mean of the arithmetic average of the weights yields a reasonable (but slightly pessimistic) approximation. For both weight designs, the Frobenius norm decreases with increasing number of mobile nodes. While the gains seem to be moderate in this figure, it should be kept in mind that $\mathrm{E}_\mathcal{T}\{\omega_1\}$ quantifies the MSE improvement for i.i.d. measurements after only two iterations. For comparison, we also show $\mathrm{E}_\mathcal{T}\,\mathrm{tr}\{\boldsymbol{R}_s \boldsymbol{W}_{2\to1}^T \boldsymbol{W}_{2\to1}\}$ for the case of correlated measurements, where the mobility gain is even larger.

We next study the convergence of AC in static and mobile WSN (with 20 nodes moving). Fig. 3 shows the MSE versus k, averaged over 100 scenarios, for the static case with three different weight designs (CVX denotes the optimum weights of [14]) and for the mobile case with MH and CW (including our analytical lower bound for CW). Clearly, both MH and CW in mobile WSN outperform the static case by far, with mobile MH performing best. The lower bound for mobile CW is exact for $k = 2$ but become less tight as k increases. While both mobile CW and mobile MH are realistic for practical mobile WSN applications, the MH design requires to determine the local node degrees in each iteration step. By contrast, the CW design has to be done only once. Finally, since the graph realizations are not necessarily connected the MSE saturates when with all static settings.

The next result illustrates the dependence of the MSE after 100 iterations on the amount of mobility. Here, 50 of the 100 nodes are moving a random distance in a given direction, perturbed by a small Gaussian jitter; the direction is initially chosen randomly and then remains constant. Fig. 4 shows the MSE versus the standard deviation of the distance traveled by the mobile nodes, which quantifies the amount of mobility. It is seen that the MSE decreases rapidly with increased node mobility. Compared to the static case, where CM and MH achieve an MSE of about -30 dB and -42 dB, respectively, the MSE improvement with high mobility is more than 130 dB.

Our last simulation results, shown in Fig. 5, shows the MSE achieved with CW and MH after 50 and 150 iterations for different graph connectivities (again averaged over 100 scenarios). The graph connectivity is quantified in terms of the normalized communication range $r\sqrt{I}$. For the mobile case, 20 out of 100 nodes are moving according the model described for the previous simulation (the standard deviation of the traveled distance here was 0.114). For very

$n \diagdown m$	2	4	6	8
0	$w+2$	$\frac{\pi}{2}(4w^2-5p_{\mathrm{m}}w^2+3)$	$\frac{w}{12}(15\pi^2 p_{\mathrm{m}}-80p_{\mathrm{m}}+6\pi^2)$	$-\frac{3\pi^3}{4}$
1	$w-2$	$\pi(w+\frac{15}{4}p_{\mathrm{m}}w^2-\frac{7}{2}w^2-3)$	$-\frac{1}{8}(2\pi^2 w-80p_{\mathrm{m}}w+8\pi^2+15\pi^2 p_{\mathrm{m}}w)$	π^3
2	0	$-\frac{\pi}{4}(4w+5p_{\mathrm{m}}w^2-6w^2-6)$	$-\frac{1}{24}(80p_{\mathrm{m}}w+12\pi^2 w-36\pi^2-15\pi^2 p_{\mathrm{m}}w)$	$-\frac{\pi^3}{16}$
3	0	0	$\frac{\pi^2}{4}(w-2)$	$-\frac{\pi^3}{4}$
4	0	0	0	$\frac{\pi^3}{16}$

Table 1. Coefficients q_{mn} of $\mathrm{E}_{\mathcal{T}}\{g_{ii}^2; i \notin \mathcal{V}_{\mathrm{m}}\} = 1+\pi\sum_{m,n} q_{mn} w^{\frac{m}{2}} (\sqrt{2}r)^m I^n$.

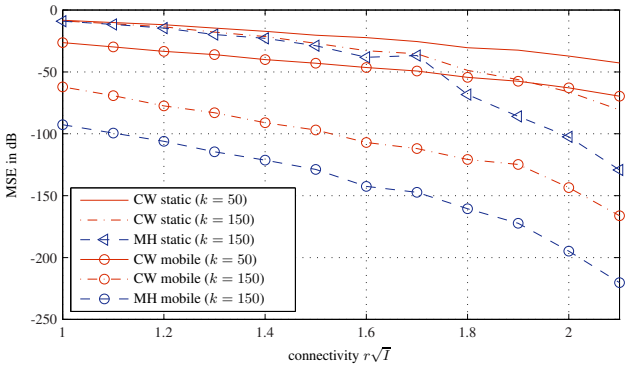

Fig. 5. *Impact of graph connectivity on the MSE.*

low connectivity, it is very likely that the graphs are not connected. Hence, for the static case there is little MSE improvement when going from 50 to 150 iterations below the connectivity threshold of $r\sqrt{I} \leq 1.7$. The behavior for the mobile WSN is quite different. Apart from performing uniformly better than the static WSN, AC works very well even way below the connectivity threshold. This can be attributed to the fact that node motion allows information to diffuse even between graph components that are temporarily not connected. Since the communication range relates directly to the node transmit power, it follows that mobile WSN can achieve the same performance as static WSN with much less transmit power. For example, to achieve an MSE of -100 dB after 150 iterations with MH, the transmit power in the mobile WSN can be about 6 dB smaller than in the static WSN.

5. CONCLUSION

In this work we have studied the MSE performance of average consensus (AC) in mobile WSN. Modeling the mobile WSN via stationary Markovian evolving graphs, we have derived a closed-form lower bound for the MSE that corroborates the beneficial impact of node mobility on the performance of distributed averaging. This lower bound and corresponding numerical results reveal that the MSE gain resulting from node mobility increases with increasing number of mobile nodes and higher node mobility. Interestingly, AC in mobile WSN can also overcome the limits of graph connectivity. Furthermore, the improved MSE translates into shorter averaging times or reduced transmit power.

REFERENCES

[1] J.N. Tsitsiklis, *Problems in Decentralized Decision making and Computation*, Ph.D. thesis, Massachusetts Institute of Technology, Dec. 1984.

[2] R. Olfati-Saber, J. Fax, and R. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007.

[3] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Trans. Inf. Theory*, vol. 52, pp. 2508–2530, June 2006.

[4] C. C. Moallemi, *A Message-Passing Paradigm For Optimization*, Ph.D. thesis, Stanford University, Sept. 2007.

[5] Patrick Denantes, Florence Benezit, Patrick Thiran, and Martin Vetterli, "Which distributed averaging algorithm should I choose for my sensor network?," in *Proc. IEEE INFOCOM-2008*, Phoenix, AZ, April 2008, pp. 986–994.

[6] D. Bickson, *Gaussian Belief Propagation: Theory and Application*, Ph.D. thesis, The Hebrew University of Jerusalem, 2008.

[7] V. Schwarz and G. Matz, "Mean-square optimal weight design for average consensus," in *Proc. IEEE SPAWC-2012*, Cesme, TR, June 2012, pp. 374–378.

[8] Mathew Penrose, *Random Geometric Graphs*, Oxford University Press, 2003.

[9] A.D. Sarwate and A.G. Dimakis, "The impact of mobility on gossip algorithms," *IEEE Trans. Inf. Theory*, vol. 58, no. 3, pp. 1731 –1742, March 2012.

[10] Sheng-Yuan Tu and A.H. Sayed, "Mobile adaptive networks," *IEEE J. Sel. Topics Signal Processing*, vol. 5, no. 4, pp. 649–664, Aug. 2011.

[11] S. Kar and J.M.F. Moura, "Sensor networks with random links: Topology design for distributed consensus," *IEEE Trans. Signal Processing*, vol. 56, no. 7, pp. 3315–3326, July 2008.

[12] S. Sardellitti, M. Giona, and S. Barbarossa, "Fast distributed average consensus algorithms based on advection-diffusion processes," *IEEE Trans. Signal Processing*, vol. 58, no. 2, pp. 826–842, Feb. 2010.

[13] V. Schwarz and G. Matz, "Average consensus in wireless sensor networks: Will it blend?," in *submitted to Proc. IEEE ICASSP-2013*, 2013.

[14] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems & Control Letters*, vol. 53, no. 1, pp. 65–78, 2004.

[15] S. Boyd, P. Diaconis, and L. Xiao, "Fastest mixing Markov chain on a graph," *SIAM Review*, vol. 46, no. 4, pp. 667–689, Dec. 2004.

[16] V. Schwarz and G. Matz, "Nonlinear average consensus based on weight morphing," in *Proc. IEEE ICASSP-2012*, Kyoto, JP, March 2012, pp. 3129–3132.

[17] E. F. C. Andrea, A. Monti, F. Pasquale, and R. Silvestri, "Information spreading in stationary Markovian evolving graphs," in *IEEE Int. Symp. on Parallel Distributed Processing*, May 2009, pp. 1–12.